UNIVERSITY OF SOUTHAMPTON

THE IMPLEMENTATION
OF
COST EFFECTIVE DEBRIS PROTECTION
IN
UNMANNED SPACECRAFT

by HEDLEY STOKES (B.Sc, M.Sc)

THESIS SUBMITTED FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

SCHOOL OF ENGINEERING SCIENCES
AERONAUTICS AND ASTRONAUTICS

February 2002

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING AND APPLIED SCIENCE

DEPARTMENT OF AERONAUTICS AND ASTRONAUTICS

Doctor of Philosophy

THE IMPLEMENTATION OF COST EFFECTIVE DEBRIS PROTECTION IN UNMANNED SPACECRAFT

By Hedley Stokes (B.Sc, M.Sc)

The spatial density of debris in popular regions of low Earth orbit is sufficiently high that the impact risk to unmanned spacecraft can no longer be ignored. To enhance protection, designers generally consider increasing the shielding of existing materials and structures. This may be through the addition of layers of high-strength materials such as Nextel and Kevlar to MLI-covered surfaces, or by modifying the design of honeycomb panels. The benefit of this type of protection comes at the price of increased mass and cost. Another approach is to consider the vulnerability and layout of equipment on a spacecraft, and configure the spacecraft so that mission-critical items are protected from the most vulnerable regions, possibly by less critical neighbouring items. This approach provides protection enhancement without incurring significant mass penalty. However, up until now, it has generally been poorly considered. Part of the reason for this has been the lack of a methodology for evaluating and comparing the survivability of different spacecraft designs. The development of a solution to this problem forms the core of this research.

The basis of the solution is the construction of a software tool called SHIELD. This has some similarities to existing tools such as ESABASE / DEBRIS, in that it determines the impact and penetration distributions on a representative 3-D model of a given spacecraft, and the probabilities of penetration. However, in SHIELD, the computation goes a step further by calculating the probability of failure of the satellite. SHIELD does this by determining the internal damage caused by each of the penetrating particles distributed on the satellite. Any equipment that lies within the line of sight of a given particle's trajectory is considered vulnerable. Multi-wall ballistic limit equations are called up to ascertain the extent of penetration inside the satellite, and therefore identify exactly which units are damaged. Knowing the criticality of these units, the consequences for the mission are then easily derived. When all penetrators are considered in this fashion, the result is the desired satellite failure probability. To derive a true measure of a satellite's survivability, SHIELD combines this failure probability with information on the various costs associated with the mission, i.e. a life cycle cost model. The result is a 'survivability metric' that enables the cost-effectiveness of radically different protection strategies to be determined and compared in a completely objective manner. This metric is also incorporated into a genetic algorithm, which is capable of searching many competing protection solutions and converging on cost-effective options.

Preliminary simulations, by applying SHIELD to idealised spacecraft, have demonstrated its validity in several respects. The model has also been used to investigate the relationship between satellite failure probability and debris particle size. This is a crucial result for satellite constellation replenishment strategies and predictions of debris environment evolution. Finally, to illustrate the model's potential for application to an actual spacecraft design, the survivability of the METOP Service Module is evaluated. One of the general conclusions to emerge from these results is that due consideration should be given to the vulnerability and layout of equipment on an unmanned spacecraft if its survivability is to be enhanced cost-effectively. In summary, the implementation of cost effective debris protection in unmanned spacecraft is a challenging problem that requires original thinking and innovative modelling techniques. This research programme represents a first step towards that objective.

List of Contents

List of Tables

List of Figures

Acknowledgements

The author wishes to recognise his employer, QinetiQ - formerly the Defence Evaluation & Research Agency (DERA), in sponsoring the Ph.D. course. Recognition is also extended to the Ministry of Defence Applied Research Programme for funding the development of the prototype model, and to the British National Space Centre, who contributed funds towards part of the SHIELD development.

Family, friends and colleagues have all provided encouragement to the author during his research, and their support is greatly appreciated. In particular, Richard Crowther is thanked for directing the author towards the subject of spacecraft debris protection, and for opening many doors into the space debris community. Roger Walker is thanked for providing numerous interesting discussions during the course of the research, especially with regard to debris environment flux modelling and collision risk analysis. Jenny Wilkinson is thanked for her extensive technical support during the ESA 'Cost Effective Debris Shielding' study, several results of which also feature in this thesis.

Finally, the author would especially like to thank and acknowledge his Ph.D. supervisor, Dr. Graham Swinerd, for providing invaluable guidance and meticulous technical reviews throughout the research programme.

1 Introduction

1.1 Background

Since 1957, when the first satellite, Sputnik 1, was launched there have been nearly 4,000 launches into space. Typically, each of these launches has placed at least one satellite into orbit. The result is a large population of satellites and related items orbiting the Earth. Currently, a total of approximately 9,000 objects are being tracked by the US Space Command's network of ground-based detectors [Jenkin (1998)]. Only ~6% of these objects are operational satellites, the remainder are uncontrolled and fall into a category known as orbital debris. Limitations in detector capability prevent objects smaller than approximately 10 cm being tracked in low Earth orbit (LEO) and 1 m in geosynchronous Earth orbit (GEO)[1]. The number of debris objects in the 1 to 10 cm range are estimated to exceed 100,000 [Office of Science and Technology Policy (1995)]. It is generally accepted that a satellite probably would not survive an impact with a piece of debris larger than ~1 cm in diameter. This means there is a technology gap between debris that can be shielded against (< 1 cm) and debris that can be tracked and avoided (> 10 cm).

The increasing likelihood of hazardous collisions occurring between orbital debris and operational spacecraft is currently receiving much attention internationally. There are several reasons for this:

- The first ever recorded collision between trackable objects occurred on 24th July 1996 when the French satellite 'Cerise' was hit by a fragment from an Ariane launcher upper stage that had exploded ten years earlier [Alby *et al* (1997)]. The collision severed a gravity gradient stabilisation boom, causing the satellite to tumble rapidly.
- The ongoing launch and construction of the International Space Station (ISS) represents a very large, expensive, high profile, manned asset that is considered particularly susceptible to impacts from debris up to ~1 cm in size. As a result, much effort is being expended to shield the most vulnerable elements.

[1] Here low Earth orbit is defined as the region of space lying between 120 km and 2,000 km above the Earth. Geosynchronous Earth orbit refers to the orbital region at approximately 36,000 km altitude.

- The imminent launch of constellation systems comprising large numbers of communications satellites (over 1,000 planned to date [Jenkin (1998)]) could strongly influence the growth of the LEO debris environment over the next fifty years. With satellites repeatedly passing through a constellation, such as spacecraft from another constellation that are orbiting/de-orbiting or background debris objects that are decaying, the risk of a collision could be significant [Walker (2000)]. Should a collision occur then the resulting cloud of fragments would spread around the orbit of the fragmented satellite, thereby posing a collision risk to other members of the constellation. It is not difficult to see that a run-away effect is possible here, known as a collisional cascade. The end result is an orbital region so populated by debris that it would become unusable for centuries.

To help overcome such problems, and mitigate the growth of debris in general, international co-operation, planning and regulation are required. For such measures to be effective they must be realistic and achievable for all users of space. Therefore, a balance must be struck that is reinforced by supporting analysis. The Office of Science and Technology Policy's 'Interagency Report on Orbital Debris' recognises this and identifies important technical issues which must be addressed, including continued development and refinement of debris environment models, satellite protection models, and shielding technologies. To these ends, various government organisations and agencies, such as the British National Space Centre (BNSC), the European Space Agency (ESA), and the National Aeronautical and Space Administration (NASA), have invested in debris research programmes.

The author's employer, QinetiQ - formerly the Defence Evaluation and Research Agency (DERA), set up a debris research group several years ago to support BNSC and other agencies in the assessment of risks posed by the orbital debris environment, and identification of measures to reduce those risks in a cost-effective manner. Initially, the research focused on the development of a high-resolution debris environment model called the Integrated Debris Evolution Suite (IDES). Once completed, preliminary predictions of future debris population growth and the associated collision risk to space systems, particularly constellations, were carried out and published. With an improved understanding of the extent of the problem, the research programme then concentrated on the question of whether it was necessary to apply debris protection to unmanned spacecraft, and if so how. This particular problem also forms the basis of the author's Ph.D.

The threat of damage to a spacecraft from orbital debris is still not fully appreciated by some satellite operators and designers. However, industry awareness is growing, and many manufacturers now recognise that the risks posed by debris impacts can no longer be ignored when designing expensive, long duration satellite missions. A single impact from a centimetre size piece of debris travelling at 10 km/s may be enough to cause the catastrophic failure of a $1 billion satellite. For example, during the Space Shuttle STS-61 mission - a Hubble Space Telescope (HST) maintenance flight - observations were made of the impact damage to the HST. Several large impact craters were identified [Christiansen & Hyde (1994)]. One of these (measuring approximately 1 cm in diameter) occurred on the high gain antenna (HGA). Calculations based on the current known debris environment predicted a 1 in 10 chance of such an impact on the HGA during the first 3.5 years. Although this impact rate might seem quite small, if the impactor had struck a critical component, it could have resulted in failure of the whole mission.

An examination of any historical database of spacecraft failures should reveal that anomalies generally are not attributed to debris impacts. For example, a simple query of the SpaceBase database at QinetiQ shows that, with certain exceptions such as Cerise, spacecraft malfunction as a result of random, wear-out or systematic failures [SpaceBase (2001)]. It is conceivable, however, that a percentage of the random failures, which are often ascribed to a component or subsystem suddenly malfunctioning, may in fact have their root cause in debris impacts. Furthermore, databases such as SpaceBase show that 10 – 15 % of all spacecraft anomalies and failures are of unknown cause. Within this residual, one might again conjecture that impacts could have played a role. A study to investigate this possibility was conducted by McKnight *et al* in 1997. They compared the spatial density of orbital debris with the number of spacecraft anomalies of unknown cause, and found that there was some evidence for a correlation. Their findings, although not conclusive, suggest that many of the anomalies are impact-induced.

Indirect evidence such as this, combined with actual observations of spacecraft impact damage, underlines how important it is to consider implementing appropriate protection measures in spacecraft. Such measures may take the form of dedicated shielding, where mass is added to enhance the survivability of existing structures and equipment. Shielding can utilise a variety of configurations, such as multi-layer bumpers, and novel materials, e.g. Nextel or Kevlar. However, the benefits of adding shielding must be weighed against the increased mass and cost to understand whether it is appropriate for the particular mission of interest.

Another protection approach is to configure the layout of equipment in a spacecraft design so critical units are less vulnerable to impacts. Essentially, this strategy maximises the latent protection offered by less critical neighbouring items. The approach should not introduce any significant costs provided it is considered early in a spacecraft design; therefore it has the potential to be a very cost-effective way of improving survivability.

A cursory look at the literature shows that there has been very little research into spacecraft architecture design as a protection method, either by itself or in conjunction with standard shielding methods. The likelihood is that both strategies together will provide the most cost-effective protection solution for a given spacecraft. Furthermore, at the outset of this Ph.D., no technique had been devised to quantify and compare these contrasting strategies. Therefore, one important aim of the research programme is to formulate a new method to solve this problem.

Because of the potentially large number of permutations for arranging a satellite's components and shielding, the identification of a cost-effective design is likely to be a tedious, time-consuming task involving much trial-and-error. To overcome this, it would seem advantageous to utilise a technique that could search through the options in an automated and efficient manner. The possible implementation of such a technique represents another important goal of the research programme.

1.2 Objectives

The focus of this Ph.D. research programme is to propose and present a new approach for implementing cost-effective debris protection in unmanned spacecraft.

The emphasis on unmanned spacecraft is deliberate, for two reasons. First, the methodology for applying manned spacecraft protection is quite well understood and generally accepted - essentially, the underlying principle is to reduce the probability of penetration to acceptable levels at reasonable cost. Second, a separate and unique methodology for unmanned spacecraft is less clearly defined. Instead, there appears to be a tendency to adopt the 'manned' approach, if any at all. While this may not necessarily give a poor design solution, it is far from certain that the implemented protection will be cost-effective. With the pressures of budgetary constraints ever present in unmanned spacecraft design, the need to define a new approach becomes more apparent.

The aims of the Ph.D. are summarised briefly as follows:

- Review state-of-the-art knowledge of the space debris environment and the impact risks posed to unmanned spacecraft.
- Review the current approaches for designing debris impact protection in spacecraft.
- Propose a new methodology for implementing cost-effective debris protection in unmanned spacecraft.
- Investigate whether modern stochastic optimisation techniques can be utilised to assist in the search for cost-effective protection.
- Develop a computer simulation model that can assess the impact and penetration risks to unmanned spacecraft, evaluate competing protection solutions, and search for a cost-effective design.
- Validate those elements of the model that are common to other existing models by comparing results and, for those elements of the model that are unique, demonstrate that the results are at the very least sensible and credible, while setting out the most likely route to achieve full validation.
- Apply the model to a series of relatively simple spacecraft designs to illustrate the diversity and complexity involved in implementing cost-effective protection in spacecraft, and thus underline the inherent benefits within the model.
- Use the model to establish a preliminary relationship between the size of an impacting debris particle and the probability that an unmanned spacecraft will fail.
- Conclude by applying the model to an actual spacecraft design, thus demonstrating its potential for contributing to the normal space environment risk assessment process.

1.3 Thesis Structure

An important element of a scientific or engineering research project is to perform a review of the literature. This provides a sound basis for the project as it establishes the need and potential benefit of the research, and mitigates the possibility of unnecessary repetition. The first two chapters in the thesis (after this one) are therefore devoted primarily to reviewing the literature and setting the scene for the programme of work undertaken in the subsequent chapters.

Chapter 2 describes the meteoroid and orbital debris environments, both of which pose a hazard to the safe operation of a spacecraft. Meteoroids, sometimes referred to as natural debris, are characterised by two population types: sporadic and streams. Several models have been developed by leading researchers such as Grün, Divine, Cour-Palais, and Jenniskens to describe the flux of particles in these two populations. Chapter 2 summarises some of the flux results from these models, thus providing an indication of the meteoroid impact risk to an orbiting spacecraft. The remainder of the chapter focuses on understanding the orbital debris environment, which is the dominant impact hazard to a spacecraft in low Earth orbit. All the known sources and sinks of orbital debris are categorised before summarising the various detection, tracking and measurement methods used to characterise the debris population. This information is essential for the construction and validation of computer models of the debris environment. A variety of models are described, including two models developed by colleagues of the author, i.e. IDES and DELTA (the Debris Environment Long Term Analysis model). The respective modelling approaches are discussed and differences highlighted. Finally, a selection of results derived from IDES and DELTA is presented. Specifically, projections of the long-term growth of the LEO debris population are discussed, with emphasis on the consequences of launching satellite constellations and the benefits of introducing mitigation measures. An example analysis of the long-term collision risk to a constellation is also performed, which predicts the number of satellites requiring replacement because of collision-induced failures. To do the analysis, a simplifying assumption is made regarding the probability of failure of a satellite when impacted by a debris particle of a given size. This assumption forms the basis of an investigation in Chapter 6.

Chapter 3 starts by reviewing some of the damage effects caused by debris impacting different elements of an unmanned spacecraft, and the possible consequences for the mission as a whole. Debris in the size range 0.1 mm to 10 mm is considered to be the most hazardous due to a combination of its impact energy and population size. The desire to protect against these particles has led to the emergence of a number of state-of-the-art shields, which are described. For manned spacecraft, these include lightweight multiple wall shields such as the Whipple, stuffed Whipple, multi-shock, and mesh double-bumper. Shielding options for unmanned spacecraft are also discussed; in particular methods for enhancing typical structural elements such as honeycomb panels. The performances of these shields are characterised by damage equations that calculate crater size, hole size, and perhaps most importantly ballistic limit[2]. In the chapter some of the most well known and commonly used examples are listed. These equations are a central feature of methodologies and tools for assessing and reducing the impact risk to spacecraft. The current widely accepted risk assessment methodology is described, together with the two most popular damage assessment tools – ESA's ESABASE/DEBRIS and NASA's BUMPER. Application of these to various spacecraft design projects has contributed to the development of a set of general guidelines for implementing protection, which are summarised. Finally, the chapter concludes with an important section discussing whether the current approaches to implementing protection, derived mainly from experience in the manned space programme, are entirely valid for unmanned spacecraft. In particular, the assumption that penetration equals failure is questioned. So too is the notion that the addition of dedicated shielding is the primary means of achieving cost-effective protection. The logical extension of this argument is that a more subtle protection optimisation approach should be possible.

In Chapter 4, a new approach is explored for the design and implementation of protection. The basis of the approach is to find the most cost-effective protection solution for an unmanned spacecraft by simultaneously considering the two main options, namely: adding shielding mass and / or rearranging the location of critical equipment. To do this efficiently, the use of an automated search technique would seem desirable. Simulated annealing, genetic algorithms, and evolutionary programming are all discussed as plausible options. However, it is revealed that some researchers view genetic algorithms as offering perhaps the greatest

[2] The ballistic limit is defined as the critical size of particle that can cause a certain type of penetration damage, such as the onset of perforation, to a structure / shield.

potential for optimisation problems. The incorporation of a genetic algorithm into a comprehensive protection optimisation model would be an innovative and challenging problem. To check the viability of such a proposition, the focus of the remainder of the chapter is the development of a prototype demonstrator. First, the high-level architecture design of the prototype is briefly described. Then, a relatively simple method for evaluating the debris impact survivability of a satellite is derived. This is necessary to distinguish between radically different protection solutions during the genetic algorithm optimisation process. The method follows three basic steps, and results in a measure of effectiveness called a survivability metric. Finally, the design and operation of the genetic algorithm is described. In the last part of the chapter, results from three distinct prototype simulations are discussed. The first result illustrates the efficiency of the genetic algorithm in finding a 'good' protection solution for a simple example satellite exposed to an artificially generated isotropic debris impact distribution. The second result shows how this solution is influenced when satellite design constraints, such as the need to retain mass balance, are introduced into the protection optimisation. The third result is essentially a repeat of the first result, except that the example satellite is subjected to a non-isotropic impact distribution. As a consequence of these results, the chapter concludes by discussing the merits of this optimisation methodology, and recommending the development of a full version of the model.

Chapter 5 provides a description of the full model, called SHIELD. As with any complex piece of computer code, the SHIELD model is designed in accordance with software engineering standards. Thus, the chapter begins by specifying the model's fundamental user and software requirements, and outlining the high-level architecture design. Four main functions are identified in the architecture design before each is described in detail. The functions are:

1. To construct and view a 3-dimensional (3-D) representation of a satellite. The method for constructing the various elements of a satellite, such as the main body, components, shelves and internal walls, is defined and explained in terms of the underlying mathematics of planes, vectors, and transformation matrices.
2. To simulate debris impacts on the satellite during its mission life. Using IDES, the generation of target-centred debris collision fluxes is a necessary first step for this simulation. The IDES approach is summarised before describing how the resulting fluxes

are used in the SHIELD Poisson generator to produce directional particles. A ray-trace approach is used to define possible intersections between these particles and the satellite geometry. Again, the mathematics of vectors and planes is called upon.

3. To evaluate the survivability of a specific satellite design. This is one of the most challenging functions in the model construction, and so a substantial portion of the chapter is devoted to addressing this issue. It begins by describing the creation of a distribution of penetrating particles on the satellite. Then, the possibilities for assessing any subsequent internal damage are discussed. These include various models to characterise the evolution of secondary debris clouds behind penetrated targets. The methodology chosen for SHIELD enables the satellite probability of failure to be determined. This is a key element of the cost-risk metric, which is derived in the chapter as the means of quantifying satellite survivability.
4. To optimise the debris protection strategy of a satellite. This final section in the chapter covers the implemented genetic algorithm search method. Essentially, the algorithm follows a very similar operation to that used in the prototype model, although an important difference does exist, which is highlighted. That is, there are more variables in the SHIELD optimisation.

The results from various SHIELD simulations are reported in Chapter 6. These provide an initial validation of the model, as well as illustrating its potential and diversity. The validation computer runs focus on confirming the accuracy of impact and penetration flux distributions on a cube-shaped test satellite. Results are compared with those already published from similar models such as ESABASE/DEBRIS and BUMPER. Next, the survivability evaluation aspects of the model are demonstrated by investigating two example satellites. In the first example satellite, the survivability of three entirely different protection strategies are quantified and compared. In the second example satellite, the potential benefit of achieving an effective arrangement of internal spacecraft equipment is illustrated. Another important result from SHIELD is its ability to run in an automated search mode for the purposes of optimising a satellite's protection. This is investigated with a satellite design that allows variability of internal equipment positioning and external shielding. The efficiency of the genetic algorithm search method is also illustrated. The chapter then moves on to discuss an important result that has not been previously reported. That is, to identify the probability that a spacecraft will fail due to an impact with a debris particle of a given size. This relationship between failure probability and impactor size is shown to be a crucial element in the determination of satellite constellation replenishment strategies, as discussed in Chapter 3. Finally, the chapter concludes by illustrating how SHIELD can be applied to assessing the debris risk to an actual spacecraft design. The results identify the individual vulnerable spacecraft equipment items, and highlight the contribution they make towards the overall debris-induced spacecraft failure probability.

Chapter 7 is the final chapter of the thesis. It provides a summary of the main elements of the research undertaken, and the principle conclusions that have been reached. It also sets out a series of recommendations for future research, including possible improvements to the SHIELD model and further simulations.

2 The Space Debris Environment

2.1 Introduction

The near-Earth space environment contains populations of natural and artificial debris, which can range in size from sub-micron to several metres. Since the earliest days of spaceflight, designers have been aware of the impact risks posed by natural debris, i.e. meteoroids, and taken measures to protect spacecraft, particularly manned vehicles. During the past forty years, though, the rapid growth in man's utilisation of space has resulted in a population of artificial debris that now represents an additional hazard to spacecraft, particularly in LEO [Kessler (1993)]. Furthermore, unless measures are taken to arrest the growth, this hazard may eventually cause entire regions of near-Earth space to become unusable. Unfortunately, many designers and operators of unmanned spacecraft may not be fully cogniscent of the current and future risks, and therefore do not necessarily incorporate appropriate risk mitigation strategies. This chapter sets out to quantify the space debris environment, and establish the basis for implementing cost-effective protection in unmanned spacecraft. Particular emphasis is placed on the population of artificial orbital debris, as opposed to meteoroids, since this is now considered to be the dominant impact threat for a large number of Earth-orbiting satellites.

2.2 The Meteoroid Environment

In this section the description of the meteoroid environment is distilled from a review by Sdunnus *et al* (1999). Meteoroids are interplanetary dust particles that orbit the sun. They pass through Earth orbital space rather than orbit the Earth as in the case of man-made space debris. This influx of natural debris particles is referred to as the meteoroid environment. At any instant, a total mass of ~200 kg of meteoroid particles is located in the low Earth orbit region.

There are two populations of meteoroids: sporadic and streams. The sporadic population is formed from meteoroid streams that have dispersed throughout the solar system over many years as a result of gravitational perturbation effects. Therefore, the sporadic flux is approximately constant throughout the year. The flux is omnidirectional with respect to the Earth, except near the Earth where gravitational effects have to be considered. Meteoroid streams are the remnants of decaying comets, and so closely follow the orbit of the parent object. This means that the Earth's orbit will intersect with the stream's orbit at a well-defined time of the year. The effect of this is to enhance the meteoroid flux significantly.

2.2.1 Meteoroid Models

Since the early 1960's meteoroid sensors have been placed on satellites to provide valuable data for quantifying the distribution of meteoroids in the solar system. This data, coupled with ground-based observations, has enabled researchers such as Grün and Divine to develop mathematical models of the sporadic meteoroid environment. Similarly, Cour-Palais and Jenniskens have been able to construct models of the meteoroid stream fluxes.

Grün Sporadic Meteoroid Model

The sporadic model developed by Grün *et al* (1985) expresses the average meteoroid flux in terms of the number of particles (larger than a given mass) per m^2 per year impacting a randomly oriented plate in the ecliptic plane at a distance of 1AU from the sun. The model assumes that the flux is isotropic, and that the average absolute velocity of a meteoroid is ~20 km/s. Flux predictions from the model have been compared with meteoroid impact fluxes on the surfaces of retrieved satellites, such as the Long Duration Exposure Facility (LDEF). A significant discrepancy was reported, which was attributed to the use of a mean velocity in the flux calculation. It was concluded that more detailed flux assessments require the entire meteoroid velocity distribution. Such a distribution was derived in the early 1970's from the Harvard Radio Meteor Project (HRMP) and later refined by Taylor (1995), as shown in Figure 2-1.

Divine Sporadic Meteoroid Model

Divine *et al* (1993) developed a mathematical model using dust measurements from interplanetary probes and observations from ground-based radars to determine the orbital distributions of meteoroids at a certain epoch. The meteoroid environment was separated into five 'phases' comprising asteroidal, core, inclined, eccentric, and halo populations. It was assumed that the core population represents the major part of the total population, while the others are subsets with common characteristics. The resulting flux equation derived by Divine *et al* includes a weighting factor to account for detector characteristics, but does not consider Earth shielding or gravitational focusing.

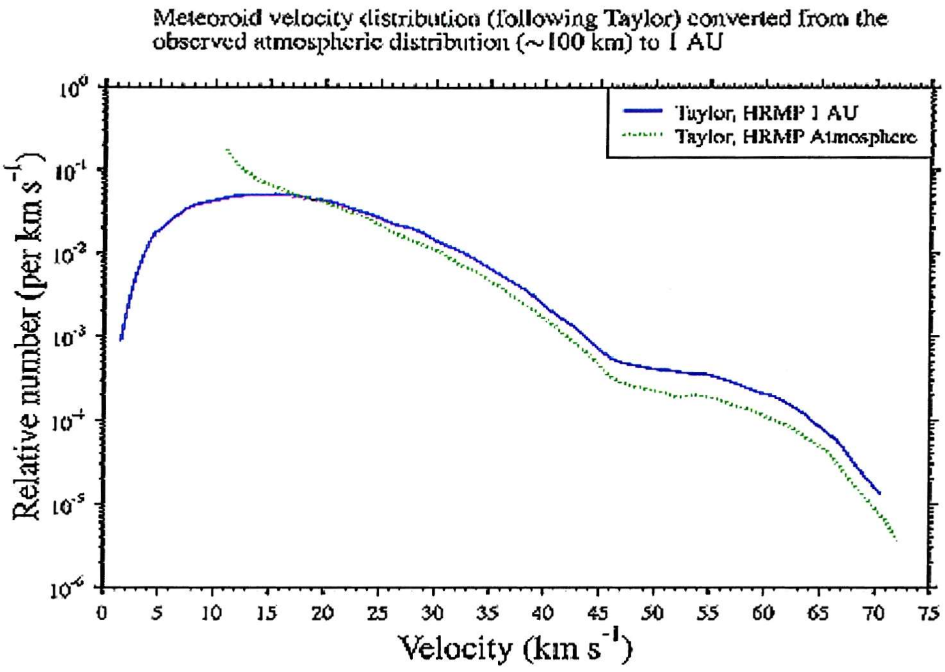


Figure 2-1 Meteoroid velocity distribution [Taylor (1995)]

Cour-Palais Meteoroid Stream Model

The Cour-Palais (1969) model describes stream activity in terms of the ratio of cumulative flux of each stream to the average cumulative sporadic flux. 18 streams are considered in total. The cumulative flux-mass model applies to each individual stream for particles in the mass range 10^{-6} grams to 100 grams. The model does not include gravitational focusing or Earth shielding effects.

Jenniskens Meteoroid Stream Model

Jenniskens (1994) has proposed a stream model based on extensive naked eye and photographic observations taken over the ten year period 1981 to 1991. 50 separate streams were identified in this data set. The activity profile of single streams is described in terms of the Zenith Hourly rate (ZHR), and it is shown that the profiles can be characterised by a set of exponential curves. One shortcoming of the current model is the absence of a mass-flux relationship.

2.2.2 Meteoroid Collision Fluxes

Flux predictions from the Jenniskens model have been compared with the Cour-Palais model, as shown in Figure 2-2 [Sdunnus *et al* (1999)]. The figure emphasises the fact that many streams are missing from the Cour-Palais model, although good agreement is achieved towards the end of the year. As a consequence, the annual mean flux predicted by the Jenniskens model is approximately a factor of three higher than that from the Cour-Palais model.

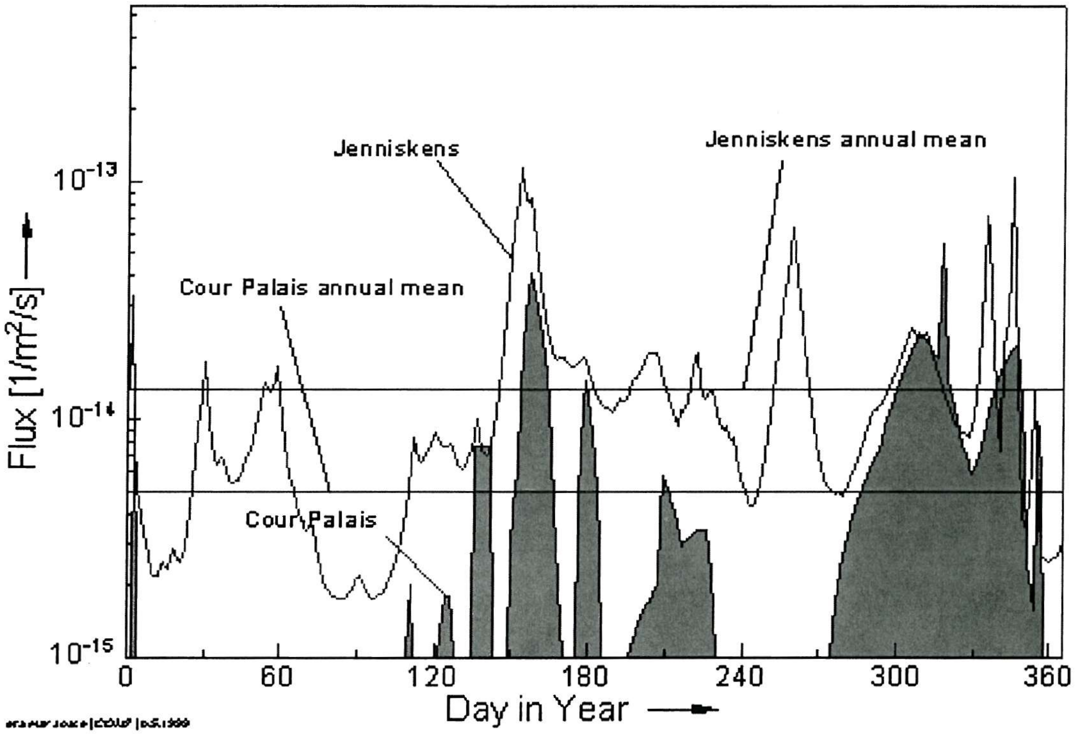


Figure 2-2 Annual meteoroid stream flux for particles > 10^{-4} kg [Sdunnus et al (1999)]

Figure 2-3 [Sdunnus *et al* (1999)] compares these annual mean fluxes as a function of particle size. The log-log plot shows a linear relationship for both models, however the Jenniskens fluxes are higher over the whole size range.

The precession of a satellite's orbit and the tilt of the Earth's equatorial plane relative to the ecliptic plane results in a meteoroid environment that can be considered omnidirectional. Meteoroid trajectories are therefore distributed isotropically in the geocentric reference frame. However, the motion of a spacecraft causes a distortion to this isotropic probability density function, which leads to a change of flux direction and magnitude. Consequently, average impact velocities and fluxes on the leading surfaces of a spacecraft are higher than on trailing surfaces. Another directionality is introduced by the presence of bodies such as the Earth or the Moon, which can provide a degree of shielding.

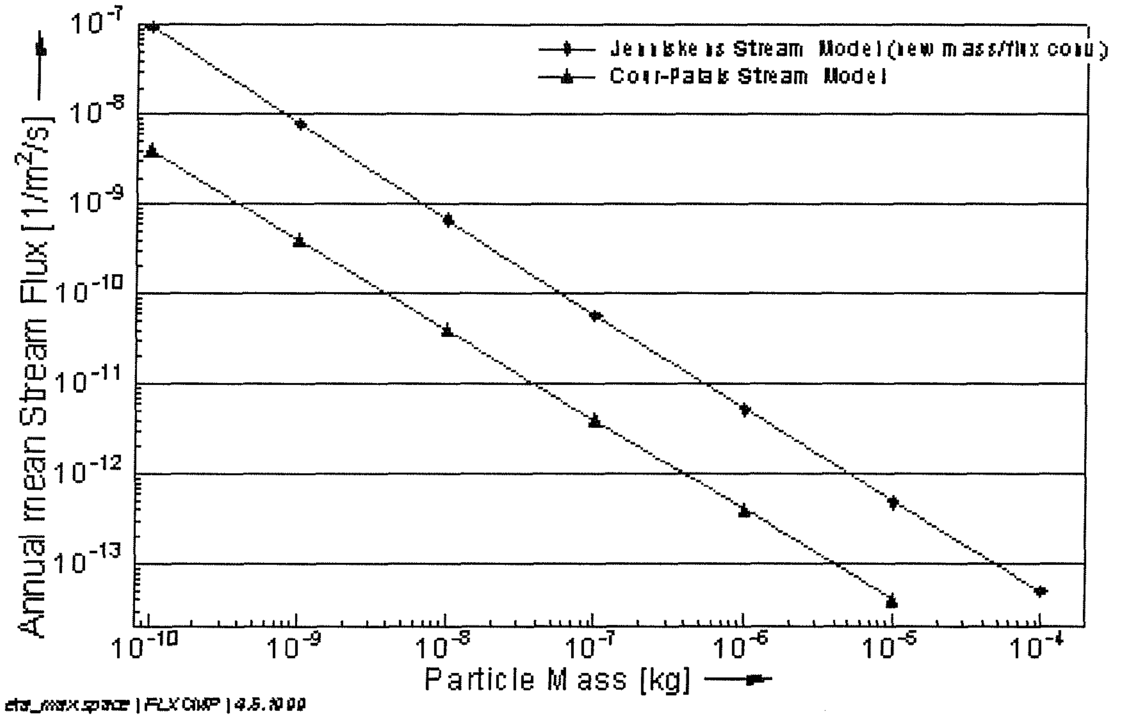


Figure 2-3 Mean annual meteoroid stream flux as a function of particle mass [Sdunnus et al (1999)]

2.3 The Orbital Debris Environment

Since the start of the space age, mankind has launched almost 5,000 satellites into orbit around the Earth [Johnson (1998)]. An unfortunate by-product associated with this activity has been the creation of a population of orbital debris. This comprises a variety of man-made objects, including fragments from on-orbit spacecraft break-ups, non-functional payloads, spent rocket bodies, mission-related objects, sodium-potassium droplets, solid rocket motor particles, and the products of surface deterioration of spacecraft [Johnson (1992)]. Approximately 70% of debris is estimated to be aluminium, with the remaining 30% being plastic, steel, copper wire and other materials [Scott (1999)].

Of the ~2,500 satellites in Earth orbit at the moment, only approximately one-quarter are operational. The remainder have reached the end of their operational life and most will continue to orbit the Earth for many decades or centuries. To place these satellites into orbit has required numerous launches, and consequently an associated orbital population of launch vehicle upper stages has also been created. Both satellites and rocket bodies play a crucial role in the development of the orbital debris population because of their size and potential for explosion. Since 1961, more than 160 of these large objects have broken up, mainly as a result of accidental explosions, and added significant numbers of smaller objects to the debris population. The large surface area of satellites and rocket bodies also means that they are susceptible to the risk of a collisional break-up, with much the same consequences.

During the deployment of a satellite, a number of mission-related objects are sometimes released, thereby adding to the debris population. These include nuts and bolts, lens covers, and launcher shrouds. Throughout their orbital lives, spacecraft can generate debris due to the gradual degradation of surface materials in the harsh space environment. For example, paint can flake off and thermal blanketing can become detached as a result of small debris and meteoroid impacts, thermal cycling, atomic oxygen erosion, and solar radiation embrittlement. One particular system of satellites known as the RORSATs (Russian Radar Ocean Reconnaissance Satellites) has also generated a large debris population of ~1 cm size sodium-potassium droplets. These leaked from the nuclear reactor coolant systems on board the satellites.

2.3.1 Measuring the Orbital Debris Environment

The techniques used for quantifying the orbital debris population are (1) detection with optical and radar systems, (2) measurements of spacecraft surfaces that have been retrieved from space, and (3) in situ measurements from debris impact detectors on board spacecraft. Radar measurements are typically used to observe debris in LEO, whereas optical observations are used for high Earth orbit. Only a small percentage of the total debris population is tracked and entered into a catalogue. This catalogue contains a record of an object's origin and its orbital parameters. The primary space object catalogue is maintained by the United States Space Command's (USSPACECOM) world-wide Space Surveillance Network (SSN), which provides the capability to track objects ≥ 10 cm in size in LEO and ≥ 1 m in size at GEO altitudes. Since 1957, more than 25,000 objects have been catalogued, and approximately 8,800 objects are currently being tracked by the SSN. Figure 2-4 shows the historical evolution of the catalogue up to 1999, broken down by the different object types [Orbital Debris Program Office (1999)]. The overall trend is one of linear growth. 94% of the catalogued objects are debris. Fragmentation debris from the break-up of satellites and rocket bodies accounts for nearly half of the tracked debris population.

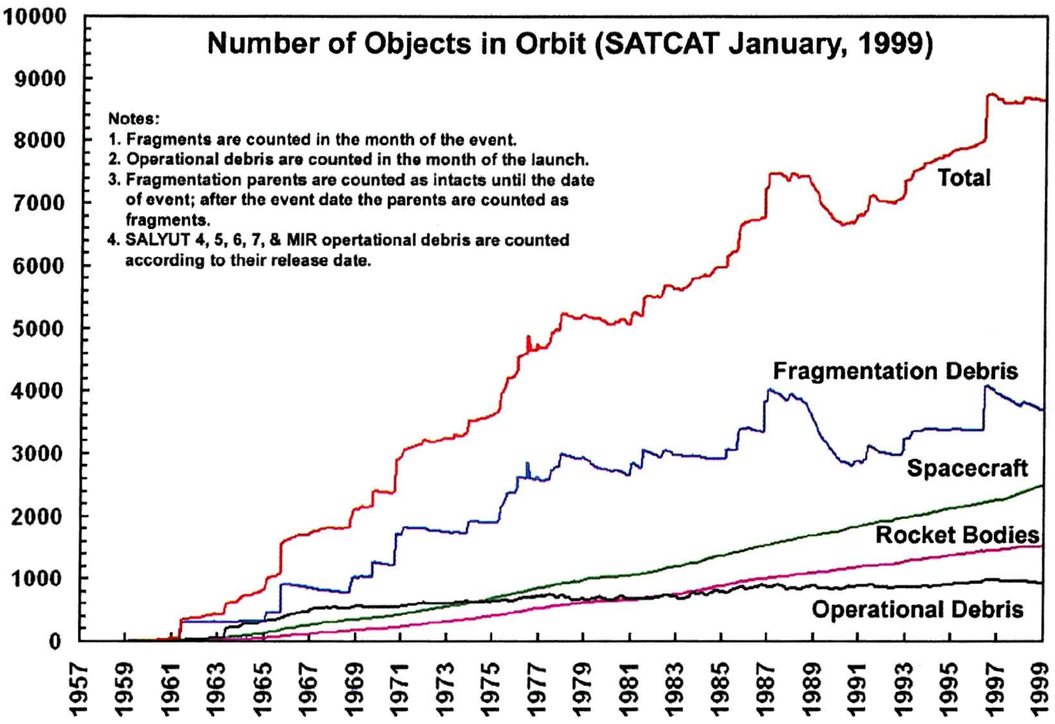
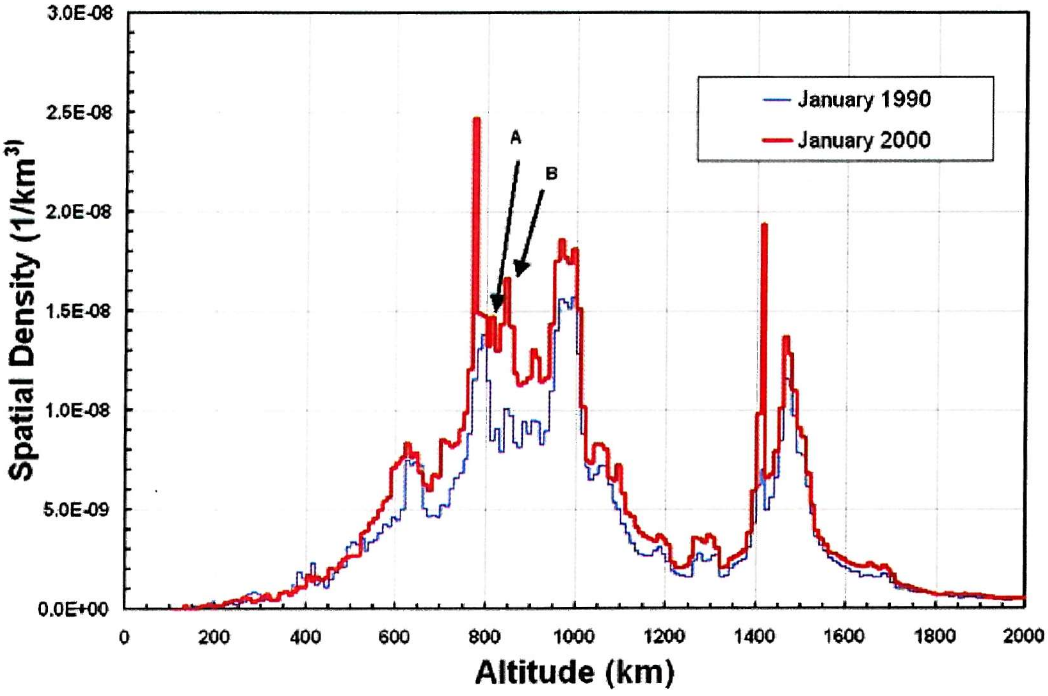


Figure 2-4 Historical evolution of the catalogued debris population [Orbital Debris Program Office (1999)]



A: Orbcomm satellite constellation; B: Tselina-2 satellite constellation.

Figure 2-5 Spatial density distribution of the USSPACECOM catalogue [Anz-Meador (2000)]

The distribution of spatial density from the USSPACECOM catalogue shows a strong dependency on altitude and latitude, especially in LEO where several recently launched multi-satellite constellations systems are operating in very narrow altitude bands. Figure 2-5 illustrates this dependency, and how it has changed during the last 10 years [Anz-Meador (2000)]. There are two peak density regions, one between 800 km and 1000 km, and the other between 1400 km and 1500 km.

The untracked space debris environment is characterised by sampling regions of near-Earth space, and extrapolating these statistically small snapshots to determine the general population. In LEO, it is possible to detect objects down to a few millimetres in size. For example, the US Haystack radar can detect 6 mm size debris at an altitude of 1000 km [Settecerri *et al* (1999)], while the US Goldstone radar is sensitive to 2 mm size debris at an altitude of 500 km [Matney *et al* (1999)]. Data on the small-size population is also acquired by examining the surfaces of spacecraft, and parts thereof, that have been returned to Earth. LDEF, EURECA (EUropean REtrievable CArrier), and a solar array from the HST have all been retrieved and analysed in this fashion [Drohlshagen *et al* (1995)]. The size, chemical composition, and distributions of impact craters have been measured to infer the relative fluxes of small debris and meteoroids in the orbital regions where these satellites operated. Figure 2-6 summarises the results of these measurement campaigns [UNCOPUOS (1997)].

From this figure one can see that a spacecraft operating in LEO will experience approximately equal meteoroid and debris fluxes in the 0.01 mm - 1 mm size range. However, outside this range, the debris flux is dominant. These results also allow estimates to be made of the total number of objects in the untracked orbital debris population, as shown in Table 2-1 [Office of Science and Technology Policy (1995)].

Size	Number of objects	% mass
> 10 cm	~ 8,800	99.93
1 - 10 cm	~ 110,000*	0.03*
0.1 - 1 cm	~ 35,000,000*	0.03*
< 0.1 cm	Trillions*	< 0.01*

* Statistically estimated values (dependent upon model used)

Table 2-1 Number of objects in the orbital debris population [Office of Science and Technology Policy (1995)]

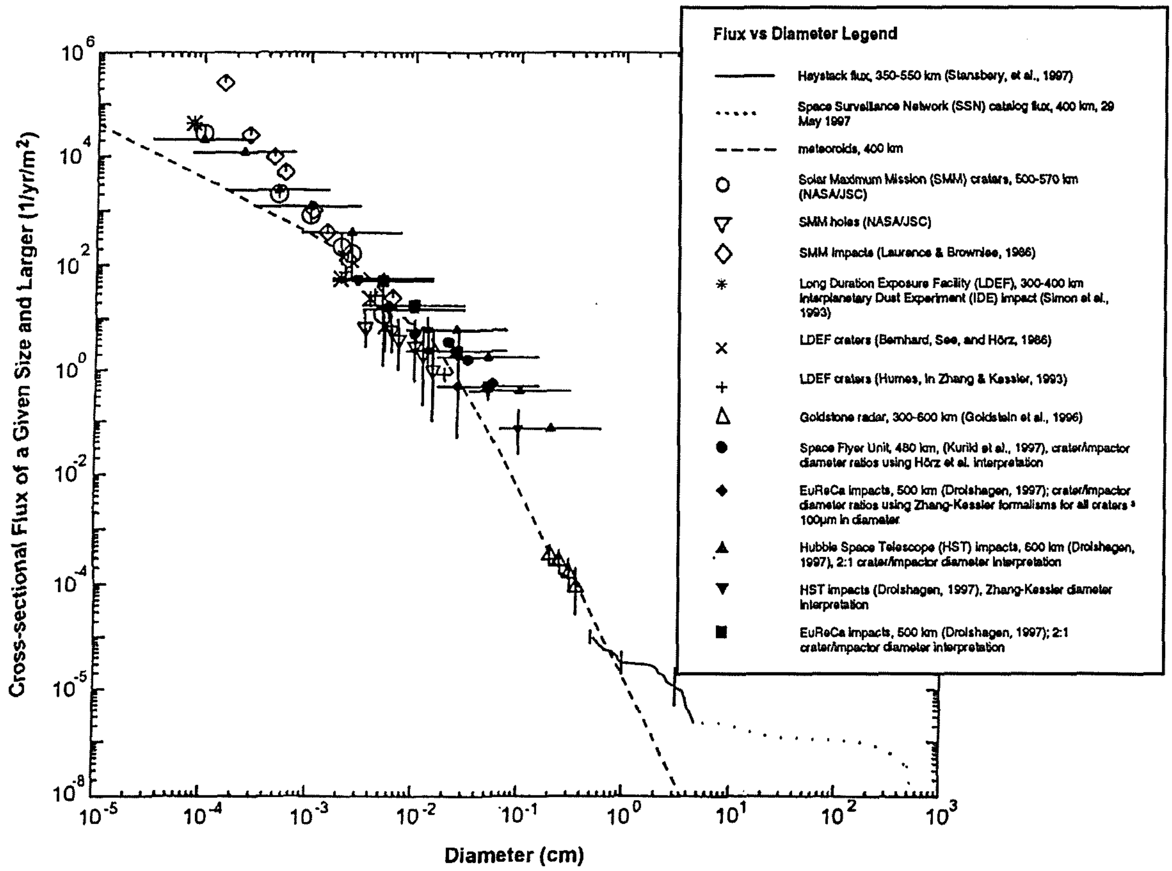


Figure 2-6 The size dependence of debris and meteoroid measurements [UNCOPUOS (1997)]

The final technique for gaining insight into the small-size debris population, in situ detection, typically uses the energy of an impacting particle to trigger a measurement. Since 1996, the GORID sensor on board the Russian Express-2 communications spacecraft has been returning data on the GEO debris and meteoroid populations in the sub-micron to millimetre size range [UniSpace Kent (2000)]. The sensor, an impact ionisation detector, measures the plasma generated by an impact to derive the mass and velocity of the particle. More recently, a standard debris detector, called DEBIE, has been developed under ESA contract to perform similar measurements. The first application of DEBIE is on board ESA's PROBA satellite, which has just been launched into a polar, low Earth orbit.

2.3.2 Modelling the Orbital Debris Environment

Based on the results from these measurement campaigns, it is generally accepted that the two most heavily utilised regions of near-Earth space - LEO and GEO - contain the highest concentrations of man-made debris. The debris populations in these regions have grown to such a level that the risk of an operational spacecraft experiencing a damaging collision can no longer be ignored. Debris in LEO are perhaps the greatest cause for concern at present for two reasons (1) they impact spacecraft with much higher velocities, and (2) the population in LEO is much larger. Therefore, the focus of the remainder of this chapter will be the LEO population.

In some regions of LEO the spatial density of the debris population may have already exceeded a 'critical density', where the rate of fragments being produced by random collisions exceeds the rate at which they are being removed by atmospheric drag [Kessler (1991)]. Consequently, the debris population may grow in an uncontrollable manner as collision fragments cause more collisions, and so on [Eichler & Rex (1990)]. This self-sustaining chain reaction, called 'collision cascading', once initiated, cannot be reversed even by reducing the number of launches. Entire regions of orbital space could become 'off limits' to future space operations, as the risk of collision would be too great. This prospect is of such concern that a number of research groups have developed models to study the long-term evolution of the orbital debris environment and the potential collision threat it poses to orbiting systems.

The evolution of the orbital debris environment depends upon many different factors as shown in Figure 2-7 [Walker (2000)]. These include:

- The rate of future launch traffic
- The rate of explosions
- Fluctuating solar activity which modulates the decay lifetimes of debris objects
- New space activities
- New debris sources
- The rate of collisions
- The efforts made by mankind to mitigate the growth of the population

In order to predict the future population over the long term, researchers have constructed complex computer models that simulate each of these debris sources and sinks in detail.

Two common approaches to debris modelling are (1) the engineering method, and (2) the semi-deterministic method. Engineering models utilise measurement data to construct empirically a mathematical representation that describes the current debris population and the associated collision risk. NASA's ORDEM96 and ORDEM2000 are the most widely used examples of an engineering model. As one might expect, results from the models correspond well with measurement data.

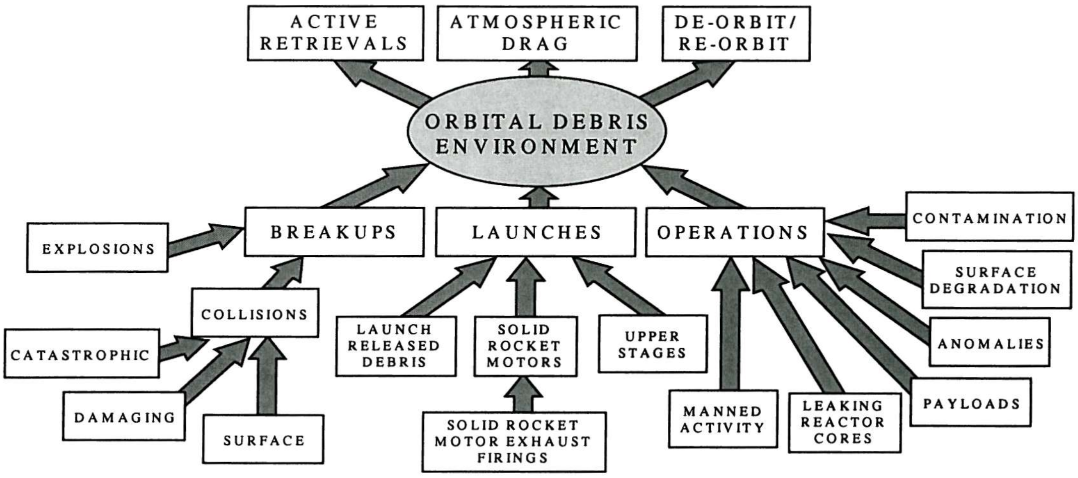


Figure 2-7 Sources and sinks of orbital debris [Walker (2000)]

Semi-deterministic models employ debris generation algorithms, such as satellite break-up models, to simulate the size and velocity distributions of each debris source as and when it occurs. The orbits of these debris objects are then propagated in time using perturbation models. Consequently, they are ideal for investigating long-term evolution trends. The most noted semi-deterministic models are MASTER, EVOLVE, IDES, and DELTA.

ORDEM96 & ORDEM2000

ORDEM96 [Kessler *et al* (1996)] uses six source components to characterise the environment. These are intact satellites, large and small fragments, paint flakes, solid rocket motor particles, and sodium-potassium droplets. Particle diameters down to 1 micron are modelled.

The model approximates the LEO debris environment by means of six inclination bands and two eccentricity families, each with its own number distribution. The two eccentricity families are: circular orbits (eccentricity < 0.2) and elliptical orbits (eccentricity > 0.2). Each of the six inclination bands has a unique altitude distribution of circular orbits, and a unique perigee distribution of highly elliptical orbits. Based on the number distributions, the model can calculate directional collision fluxes on a target satellite.

ORDEM2000 [Anz-Meador *et al* (2000)] is the successor to ORDEM96. The most recently available measurement data is used to derive the LEO debris populations. Template files store the spatial density, velocity distribution, and inclination distribution of debris particles of a given size and larger at a given latitude and altitude. The debris flux on a spacecraft is calculated from the spatial density, and the relative velocities between the spacecraft and the debris particles.

An environment evolution function is also included in the model, for long-term projections up to the year 2030. The function is based on the altitude dependent spatial density variations calculated from NASA's EVOLVE model.

MASTER

The ESA MASTER-99 model [Klinkrad *et al* (2000)] characterises both the debris and meteoroid environments, using semi-deterministic analysis and prediction techniques. It generates a high-resolution, 3-dimensional description of the debris distribution from LEO up to GEO. All of the major debris sources are modelled, including tracked objects, fragmentation debris, solid rocket motor particles, sodium-potassium droplets, paint flakes, ejecta, and aluminium oxide dust. The model is also able to trace back the characteristics of all contributing objects, e.g. in terms of source, orbit, mass, and size. However, the model is not suitable for examining the environment more than 10 years either side of the MASTER reference epoch, which is currently set at 1999, because of inherent limitations in the extrapolation algorithms.

EVOLVE

EVOLVE [Krisko *et al* (2000)] is NASA's primary model for long-term environment projections. It simulates the environment by placing each of the debris sources in their specific orbits, and calculating how these orbits change with time. From these orbits, the model is then able to calculate the orbital debris flux as a function of time, altitude, and debris size.

Past objects are launched into orbit in accordance with historical data, whereas future objects are launched based on mission models. EVOLVE places break-up fragments into the environment using the mass and velocity distributions in NASA's break-up models. Similarly, for non-fragmentation debris, other deposition models are used.

An important feature of EVOLVE is its ability to model debris mitigation measures. Scenarios have been run where future launch rates, explosion rates, and post-mission orbit lifetimes have been controlled to investigate the long-term environmental benefits.

IDES

IDES is another well-known model, which was developed by DERA (now QinetiQ) to:

1. Model the historical, current and possible future orbital debris environments in LEO.
2. Provide directional collision risk assessments for individual orbiting satellites or multi-satellite constellation systems.

IDES characterises the current debris environment by simulating the debris source and sink processes occurring within fixed time intervals from the beginning of the space age to the present day. This simulation of historical population evolution includes debris larger than 10 microns in size. The debris sources modelled by IDES are: low intensity explosions, high intensity explosions, catastrophic collisions, payloads, rocket upper stages, operational debris, liquid metal coolant droplets, secondary ejecta and paint flaking from meteoroid/debris impacts. IDES is capable of modelling all of the major orbit perturbations for debris larger than 10 microns, which include geopotential, atmospheric drag, luni-solar, and solar radiation pressure effects.

Once the current debris population is defined, IDES can use this as the initial condition for simulations of the future environment. The simulations also utilise detailed models of future launch traffic projections and explosion rates. A constellation mission model allows the introduction of one or more satellite constellation designs into the population at any time in the future. Finally, to predict catastrophic impacts for large target objects in the evolving population, IDES uses a target-centred approach for collision event prediction.

Specific information on the IDES modelling techniques, and their validation, can be found in Walker *et al* (1996 & 1999). Most recently, development activities have focused on extending IDES to model the GEO debris environment.

DELTA

The DELTA model was developed by DERA to provide ESA with a new, sophisticated simulation and analysis tool for performing long-term prediction of the LEO debris environment and its associated mission collision risks. The model allows users to investigate the long-term environmental effects of a wide variation of satellite constellation deployment scenarios and debris mitigation strategies, such as explosion prevention and post-mission disposal. This latter option includes atmospheric drag de-orbiting within a given period of time, and re-orbiting to a given storage orbit altitude band.

In many respects the structure and functionality of DELTA is very similar to IDES. However, there are notable differences. For example, DELTA uses the ESA MASTER-99 model reference population as the input for all major sources of debris larger than 1 mm in size. The evolving population within DELTA is defined by individual representative objects. A fast, analytical orbit propagator was developed to propagate the orbits of these objects, taking into account all relevant perturbations during a simulation. The objects also have full classification and origin information associated with them to enable the identification of population histories for any simulated constellation system, fragmentation cloud or debris source. This is particularly valuable in collision event prediction, as it identifies the types of impactor causing collisions, and ultimately leads to a better understanding of the collision cascading process. A more detailed description of the DELTA model can be found in Walker *et al* (2000b).

2.3.3 Predictions of the Orbital Debris Environment

An important application of debris environment models is to determine the impact that new space activities will have on the stability of the debris environment. A good example of this is the introduction of several large multi-satellite constellation systems into LEO during the next decade. These are being launched to provide various global communications services, such as mobile telephony. Two such constellations, Orbcomm and Iridium, are already operational and have added over 100 satellites and a number of associated rocket bodies to the space environment. The majority of constellations are being deployed in the regions of peak debris density where the debris collision risk is the greatest. These systems could be utilised for many years in a growing debris environment that presents a significant long-term collision hazard. It is important therefore to understand the effect the systems will have on the environment, and vice versa, particularly when one bears in mind the possible onset of the collision cascade process.

Concern over the addition of so many new objects in LEO, with their potential to exacerbate the debris problem, has prompted the proposal of a variety of debris mitigation measures. These are the steps taken during launch and mission operations in order to limit the production of different types of debris in the future. The measures include:

- Tethering launch and operations-related objects to the vehicle.
- Prevention of on orbit explosions by venting residual fuel and discharging batteries.
- Performance of end-of-life manoeuvres to de-orbit a spacecraft immediately, or reduce its post-mission orbital life.
- Performance of end-of-life manoeuvres to re-orbit a spacecraft, i.e. to raise it into a higher storage orbit.

Because of the costs associated with these types of mitigation measures, it is essential to quantify their effectiveness before they are implemented. Long-term debris evolution models offer the facility to examine different mitigation options for a variety of possible future scenarios.

Typical Study Definition

The stochastic nature of modelling future debris sources such as collisions, explosions, and launches requires models to be used in a Monte Carlo mode to represent different statistical permutations of long-term evolution. Typically, 10 Monte Carlo simulations are required for a particular scenario to obtain a reasonable statistical average of the long-term evolution results. It is useful to consider four distinct scenarios when performing predictions of the future environment. They are:

1. Business as usual (BAU), where future space activity, e.g. launch rates and explosion rates, mirrors the activity of the recent past.
2. Business as usual with constellations (BAUCONST). This is the BAU scenario, but with satellite constellations launched into LEO at various times in the future.
3. Business as usual with mitigation measures (BAUMIT). This is the BAU scenario, but with one or more mitigation measures applied at given times in the future.
4. Business as usual with constellations and mitigation measures (BAUCONMIT). Finally, constellations and mitigation measures are added to the BAU scenario.

Example Study Definition

Several researchers have used such scenarios to conduct long-term environment evolution projections, including Bendisch & Rex (1995), Reynolds *et al* (1997), Rossi *et al* (1997), and Walker *et al* (1997a&b). These studies concluded that the large increase in mass and area at critical altitudes, due to the presence of constellations, may accelerate long-term debris growth in LEO and trigger the collision cascading process much sooner than previously estimated. In particular, Walker *et al* (1997a&b) showed that the LEO space environment could not sustain the long-term operation of a 924-satellite constellation in a region of high spatial density. The greatly enhanced, localised collision risks associated with the constellation could not be entirely counteracted by the introduction of mitigation measures.

Since these studies were performed there have been some significant changes to the designs of planned constellations. In particular, the proposed 924-satellite system has been reduced to 324 satellites. Not surprisingly a change of this magnitude is likely to make a dramatic difference to long-term projections. With this issue in mind, Walker *et al* (2000a) have recently used IDES to re-examine the four above-mentioned future evolution scenarios using a current and credible set of initial assumptions, as follows:

- ❑ 'Business as usual' is defined by a future launch rate of 89 per year, and an explosion rate of 5.5 per year. These rates were derived from the average of the 8 years of activity between 1990 and 1998.
- ❑ Five constellations are launched into LEO between 1998 and 2002. The parameters of each constellation are defined in Table 2-2, and were chosen on the basis of current projections of constellation launch traffic.

System Name	Year of Launch	Altitude (km)	Inclination (degrees)	Total No. satellites	Satellite Area (m^2)	Satellite Mass (kg)
CONST1	2002	1375	85	324	12	1400
CONST2	1998	780	86	72	9	700
CONST3	1999	1414	52	56	10	450
CONST4	1998	775	45	28	9.6	42
CONST5	2002	1457	55	80	12	800

Table 2-2 Typical design parameters of each constellation in an IDES study [Walker et al (2000a)]

- Launch vehicle upper stages associated with the deployment of the five constellations are assumed to be safely de-orbited immediately after delivering their payloads into an initial parking orbit. All the constellations are assumed to operate until 2050, with expended satellites being de-orbited and replaced by spare satellites on demand. Any constellation satellites experiencing a catastrophic debris collision are also assumed to be replaced. Constellation satellites are assumed not to explode, and to be de-orbited if they fail during operation.
- The chosen parameters for the mitigation measures are: limitation of mission-related objects from 2005; prevention of explosions from 2010; immediate de-orbiting of upper stages from 2005; and end-of-life de-orbiting of satellites from 2015.

Future Population Trends

The IDES predictions in Figure 2-8 represent the projected evolution of the average LEO population (> 1 cm in size) for the four scenarios defined above. In order to put these projections into perspective, a prediction of the historical evolution of the population is included. Historically, the > 1 cm population has been dominated by fragments generated from high intensity explosions, and the liquid sodium-potassium coolant droplets leaking from the Russian RORSAT spacecraft. The 'historical' curve shows a fairly rapid increase in the number of objects from 1965 to 1988, where it reaches a peak of ~150,000. After 1988, a combination of factors rapidly reduce the population to its current predicted level of ~120,000 objects. These include the cessation of RORSAT coolant leaks, a reduction in the rate of high intensity explosions, and a greater removal rate brought about by increased atmospheric density at the time of the 1990 solar maximum.

After 1998, if no mitigation measures are introduced, then the population growth takes on what appears to be an exponential growth trend. This is the case whether constellations are present or not. The > 1 cm population is projected to increase from 120,000 objects in 1998 to nearly 200,000 objects by 2050. Interestingly, the presence of the constellation traffic has a relatively minor effect on the population growth. This is because the average number of collision-induced break-ups is predicted to be relatively low during the 52-year period. When constellations are not present, 11 catastrophic collisions are predicted to occur; when they are included, this number increases to only 13.

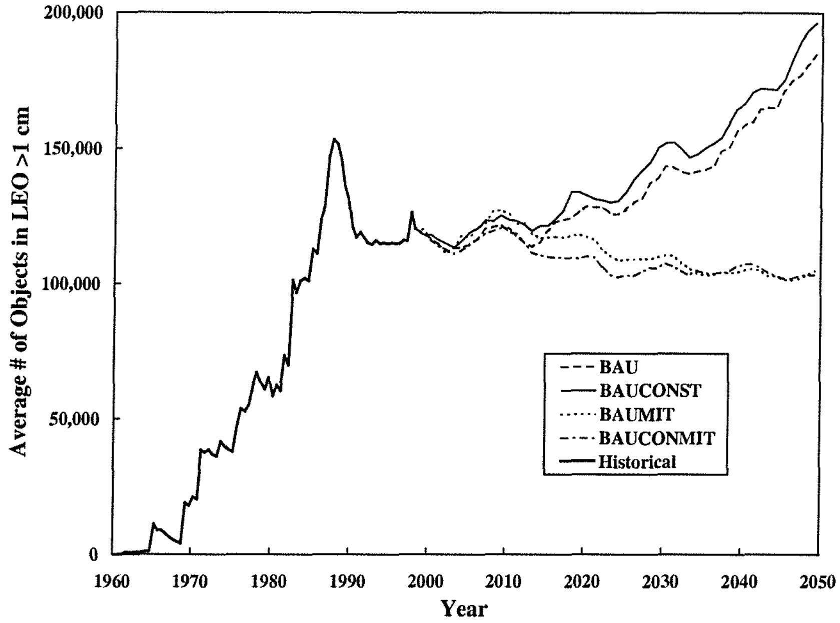


Figure 2-8 Predictions for the evolution of the > 1 cm population [Walker et al (2000a)]

The introduction of mitigation measures produces a dramatic difference in the population growth curves. Indeed, the long-term trend from 1998 to 2050 is for a reduction in the population to ~100,000 objects, i.e. half what it would be without the measures present.

Future Spatial Density Trends

An important means of understanding long-term debris environment projections is to plot the growth of spatial density versus altitude. This is usually done for a given particle size threshold and future traffic scenario. Figure 2-9 is an example of such a plot. It shows the evolution in spatial density of objects > 10 cm for a 'business as usual' scenario with constellations present, but excluding mitigation measures. Initially, in the year 2000, the main peaks in spatial density are located at 775 km, 825 km, and 1000 km altitude, with two smaller peaks located at 1400 km and 1475 km. The 775 km peak is largely due to the two constellations, CONST2 and CONST4, launched in 1998. In 2002, the addition of the CONST 5 constellation causes a slight increase in the 1475 km peak. A completely new peak emerges at 1375 km as a result of the 324-satellite constellation, CONST1. This new peak in 2002 is actually higher than all of the others.

The growth in spatial density over the 50-year period follows a linear trend at all altitudes. However, the growth rates are altitude-dependent. The highest growth rates are observed in the 800 km to 1000 km altitude band, with a four-fold increase in spatial density occurring between 2000 and 2050. This compares to only a two-fold increase at the 1400 km to 1500 km altitude band. The cause of these different growth rates is due to the significant differences in future launch traffic and explosion rates for the different altitude regions.

Figure 2-10 repeats the simulation in Figure 2-9, but this time the effects of introducing mitigation measures are shown. The spatial density versus altitude profile, and its evolution up to 2010, is identical to the previous plot. However, after 2010, there is no significant growth in spatial density at any altitude. This is due entirely to the introduction of the explosion-prevention mitigation measure in 2010.

Figure 2-11 shows the projected spatial density evolution of objects > 1 cm for the 'business as usual' plus constellations scenario. Initially, in the year 2000, the profile is dominated by the sodium-potassium droplet population between 800 km and 1000 km. This produces a spatial density peak at ~900 km, which is a factor of four greater than the underlying spatial density caused by high intensity explosions. The long-term trend for this peak at 900 km shows a net decrease over the next 25 years, due largely to the atmospheric drag effect removing the droplets. However, despite the droplet removal process continuing beyond 2025, the peak starts to rise again. This is due to another debris source dominating the region, caused by the growth in fragments from high intensity explosions and catastrophic collisions. By 2050, the spatial density at 900 km is comparable to the predicted present-day spatial density.

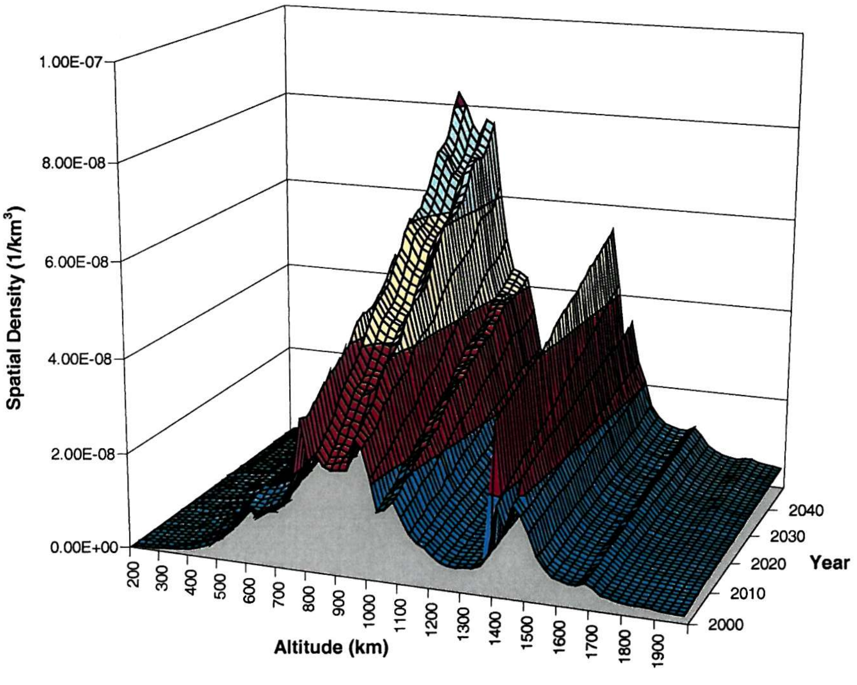


Figure 2-9 Predicted spatial density evolution of objects > 10 cm for a 'business as usual' scenario with constellations included [Walker et al (2000a)]

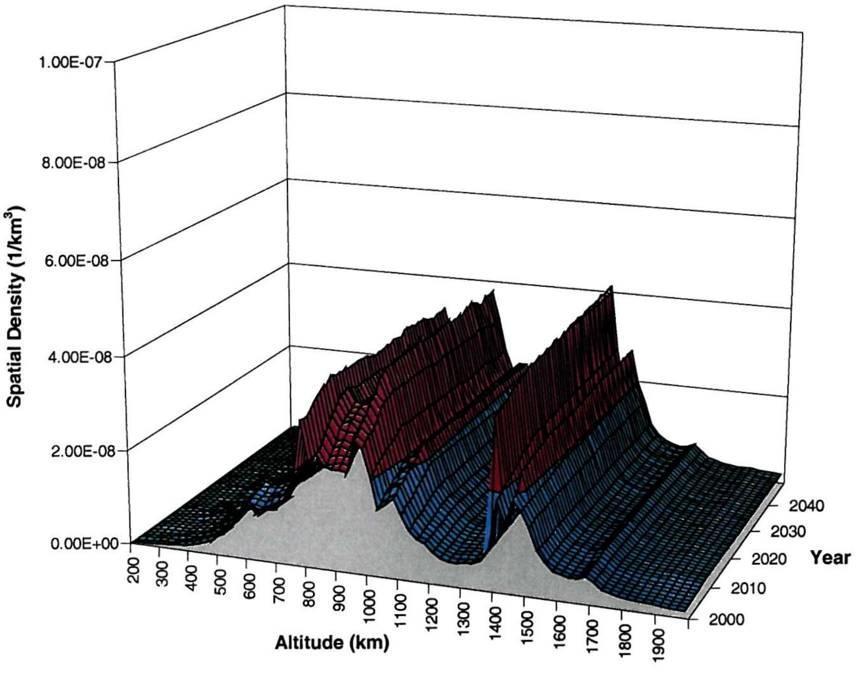


Figure 2-10 Predicted spatial density evolution of objects > 10 cm for a 'business as usual' scenario with constellations and mitigation measures included [Walker et al (2000a)]

Finally, a repeat of the simulation in Figure 2-11, but with the mitigation measures added, gives the plot shown in Figure 2-12. From 2000 to 2025 the spatial density versus altitude profile is very similar to before. Beyond 2025, though, there is a dramatic difference in the evolution of the peak density region between 750 km and 1000 km. Now, the spatial density continues to decrease through to the year 2050. This is because the mitigation measures have the desired effect of inhibiting the generation of fragmentation debris > 1 cm in size.

Studies of Mitigation Measures Effectiveness

The preceding results give a good indication of the potential for mitigation measures to stabilise and possibly control the long-term growth of the debris population. Studies are now ongoing to try to identify a package of measures that is not only beneficial to the long-term environment, but also does not impose an unnecessary or unreasonable cost burden on spacecraft operators in the future. For example, the end-of-life de-orbiting option is recognised as being the most effective mitigation measure because it mininises the growth rate of large objects, which are the most susceptible to collision due to their large masses and surface areas. However, this environmental benefit comes at a price to the operator, through increased costs to the spacecraft propulsion system.

As a compromise, it has been suggested that instead of performing immediate post-mission de-orbit manoeuvres, spacecraft could be de-orbited into a decay orbit with a lifetime lasting several years. Walker *et al* (2000c) have used DELTA to study this option for 10, 25, 50, 75, and 100-year post-mission lifetime decay orbits. It was concluded that the 75 and 100-year options may not be sufficiently effective in stabilising the environment, whereas the 10, 25, and 50-year options would be, provided they are performed by lowering only the perigee altitude to create an eccentric disposal orbit. On the face of it then, a decay lifetime of 50 years or less would seem to be a satisfactory solution. However, because the perigee altitudes of the eccentric decay orbits intersect altitudes where manned vehicles operate, large transient increases in collision risk may be experienced by these vehicles. Whether the risks are tolerable is a matter that is currently under investigation.

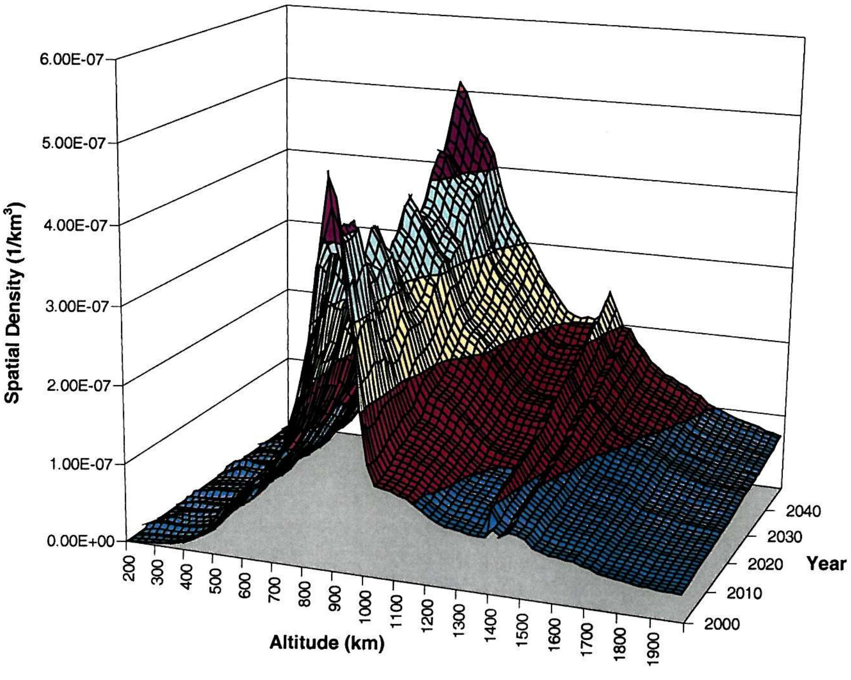


Figure 2-11 Predicted spatial density evolution of objects > 1 cm for a 'business as usual' scenario with constellations included [Walker et al (2000a)]

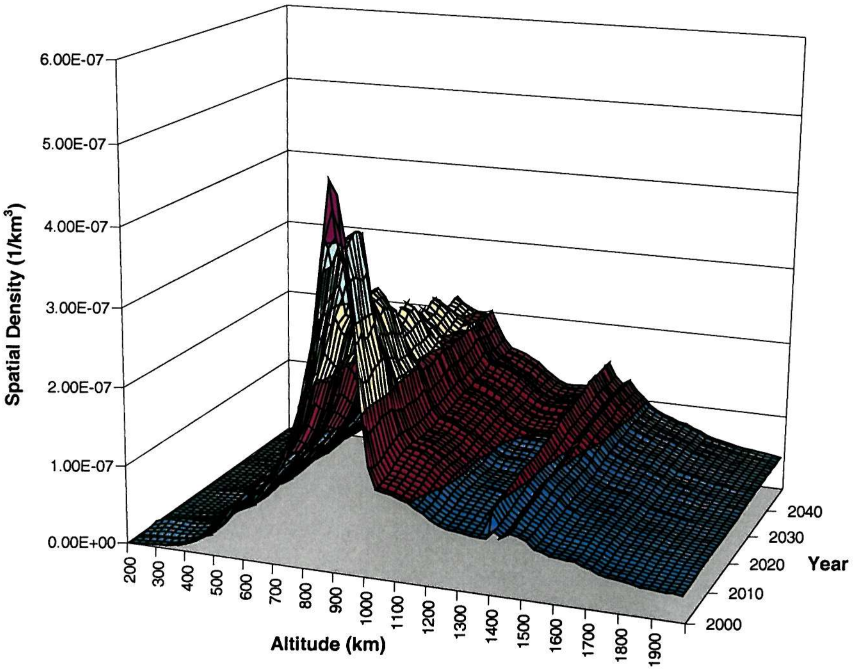


Figure 2-12 Predicted spatial density evolution of objects > 1 cm for a 'business as usual' scenario with constellations and mitigation measures included [Walker et al (2000a)]

An alternative strategy to post-mission de-orbiting, that has recently been suggested, is to re-orbit LEO satellites into higher storage orbits, e.g. at 1800 km. However, Walker *et al* (2000c) show that this is not an effective long-term solution. The lack of atmospheric drag, combined with collision activity amongst the re-orbited objects, has the potential to cause environmental instability in the storage region beyond the year 2100.

One solution that does appear to offer great potential, in resolving the long-term debris growth problem, is for future LEO spacecraft to use electric propulsion systems to perform de-orbit manoeuvres. Very short post-mission de-orbit lifetimes of the order of one or two years can be achieved at relatively low cost (Walker *et al* (2000c).

2.3.4 Orbital Debris Collision Risks

From the preceding results, it is clear that considerable uncertainty surrounds the likely growth in the debris environment. This translates into considerable uncertainty in the collision risk to spacecraft operating in high density regions for several years. Designers and operators need to quantify these risks in order to be assured that their spacecraft satisfy specified through-life survivability requirements. The only way to achieve this properly is to undertake a range of long-term collision risk projections. It is not sufficient simply to quantify the long-term collision risks on the basis of the current debris environment.

As an example of a long-term collision risk prediction, Walker *et al* (2000a) have used IDES to calculate the average collision flux of debris > 1 cm that would be encountered by a typical LEO constellation satellite over the next 50 years. For the simulation, one of the satellites from the CONST2 constellation was chosen (see Table 2-2). The same future traffic scenarios, explosion rates, and mitigation measures as described in Section 2.3.3 were also applied. The resulting long-term collision flux predictions are shown in Figure 2-13.

The plot shows an increase in collision flux on the target satellite of ~50% by 2008, just ten years after it was launched. This can be attributed to the orbital decay of the population of sodium-potassium droplets; the peak of which is currently located only ~100 km above the satellite's orbit. If no mitigation measures are introduced, then the collision flux continues to increase between the years 2010 and 2050. The rate of growth seems to follow an exponential trend, so that by 2050 the flux is approximately twice that experienced by the satellite in the year 2000. Referring back to the explanations for the long-term environment projections in Figure 2-11, it is clear to see that explosion and collision fragmentation activity in this orbital region is the main reason for the increase. However, when mitigation measures are introduced a quite different picture emerges. From 2010 a gradual decrease is observed, which brings the collision flux back down to its 1998 level. This too is consistent with the environment projections in Figure 2-12.

A designer of a constellation needs to understand how collision fluxes such as these might affect the system, particularly in terms of the number of satellites that may experience a mission-terminating failure during the operational life of the constellation. From this, the number of replacement satellites can be determined, and hence the cost associated with the collision risk.

To determine the number of satellite failures, the number of particles (of a given size) impacting the satellites must first be calculated. This is simply a matter of multiplying fluxes by satellite cross-sectional areas over specified time intervals. Then, to find the number of failures, the number of impacts is multiplied by the probability of failure associated with that size of particle impacting the satellites. In order for models such as IDES to perform this computation, values of failure probability currently have to be assumed (see Table 2-3). Typically, particles > 1 cm are always assumed to cause mission failure, whereas between 1 mm and 1 cm a range of failure probabilities is applied. The probabilities are estimated purely on the basis of engineering judgement, which is derived from an understanding of impact test results on typical satellite structures, [e.g. Turner *et al* (1999)]. Essentially, it is assumed that small debris particles produce less penetrative damage, which is also more localised. Therefore, the particles are less likely to cause critical satellite equipment to fail.

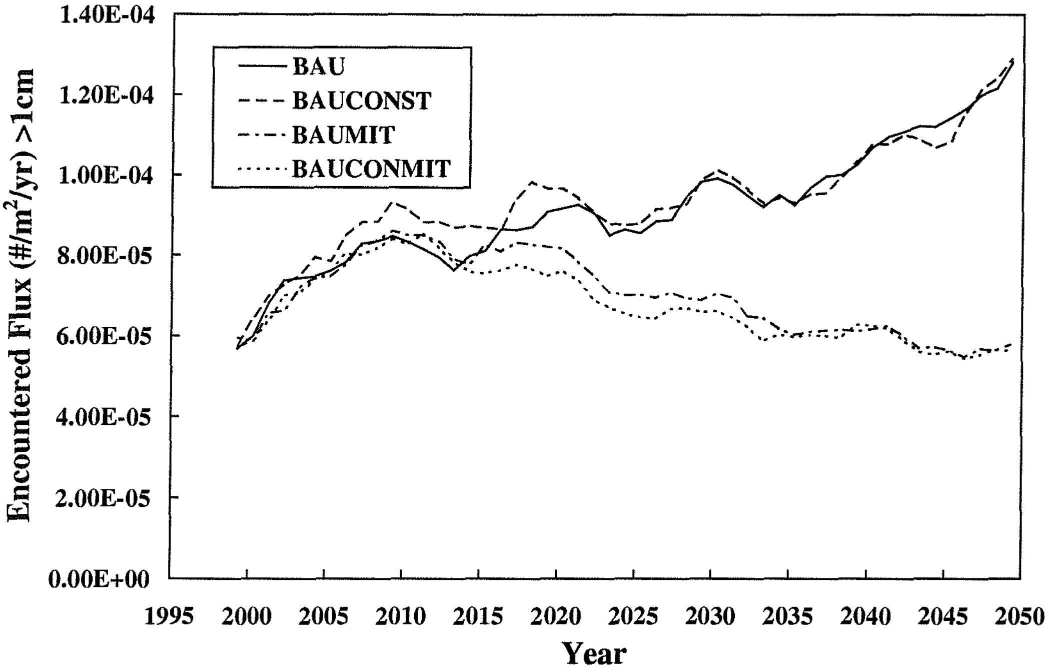


Figure 2-13 Long-term collision flux predictions for a CONST2 satellite [Walker et al (2000a)]

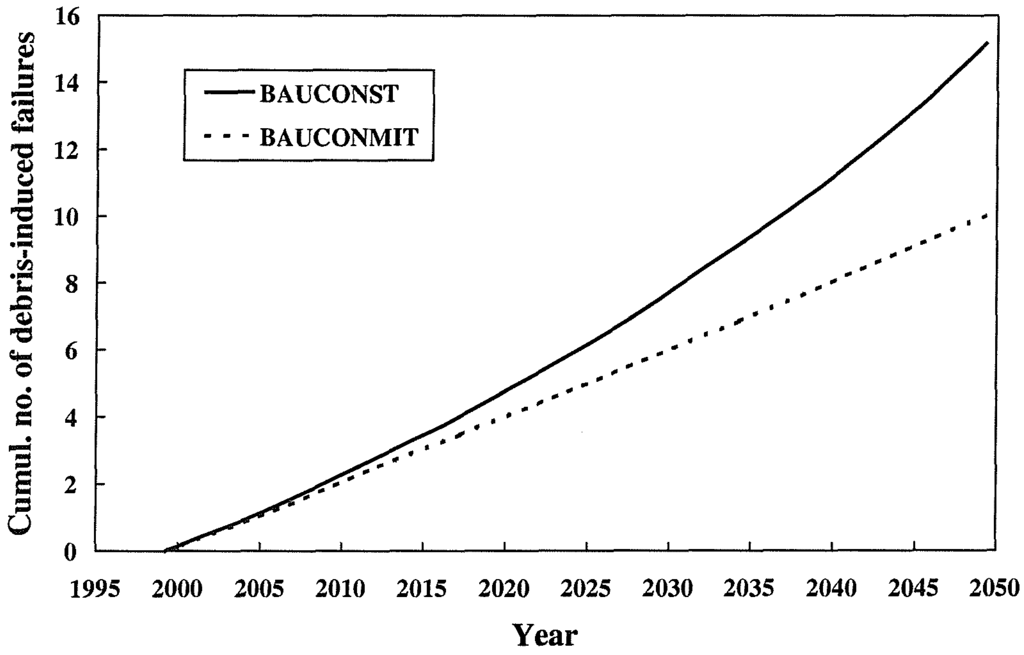


Figure 2-14 Average number of CONST2 satellites experiencing collision induced failures [Walker et al (2000a)]

Debris size range	Impact failure prob.
1 mm – 2.2 mm	10 %
2.2 mm – 4.7 mm	30 %
4.7 mm – 10 mm	60 %
> 10 mm	100 %

Table 2-3 Estimated satellite failure probability for a given impactor size [Walker et al (2000a)]

By employing this procedure, Walker *et al* (2000a) have determined the estimated cumulative number of debris-induced failures for the CONST2 system (see Figure 2-14). In the absence of mitigation measures, a total of 15 satellites are predicted to fail by the year 2050. The cumulative number of failures also appears to increase exponentially with time. However, when mitigation measures are introduced the long-term trend becomes linear, resulting in a total of 10 satellite failures by 2050.

2.4 Discussion

It is worth highlighting the most important findings from this chapter, before considering the design implications for unmanned spacecraft. The first point to note is that the collision risk from orbital debris in LEO is now considered to exceed the risk from the meteoroid environment, although the opposite is probably true at geostationary altitudes [Kessler (1993, 1994)]. In LEO, the spatial density of orbital debris is altitude and inclination dependent. Peak densities occur at 800 - 1000 km and 1400 - 1500 km. Near-polar inclination low Earth orbits are the most densely populated. It is not difficult to see why these high density regions have emerged, since many spacecraft tend to operate there. The operational popularity of the orbits is also likely to continue for the foreseeable future, thus potentially amplifying the hazard. This is particularly true when considering the introduction of several multi-satellite constellation systems in the highest density regions. However, appropriate mitigation strategies, such as de-orbiting satellites at end-of-life, appear to offer a credible means of controlling the growth of the orbital debris population.

Although there is much uncertainty in the long-term evolution of the future orbital debris environment, it is possible to predict the through-life collision fluxes on individual satellites, or systems of satellites, for a range of scenarios using models such as IDES and DELTA. This gives an estimate of the most likely fluxes that a mission will experience, as well as the upper and lower bounds. The translation of these fluxes into a meaningful assessment of the probability of failure of an unmanned satellite is identified as an area where little or no data is available, so crude assumptions have to be made. This is particularly problematic for the design and operation of a constellation, since it causes difficulty in accurately predicting the number of replacement satellites required over the life of the system. The need for research here is quite clear. Without it, designers of unmanned spacecraft do not have a sound basis for implementing cost-effective protection measures. One of the aims of this Ph.D. research programme is to address this issue.

3 Review of Spacecraft Debris Protection

3.1 Introduction

As previously discussed, the orbital debris hazard in LEO is considered to be much greater than in GEO. In LEO, not only is the debris population greater in number, but the impact velocities are an order of magnitude higher, i.e. of the order of ~10 km/s. For these reasons, there is a tendency to place more emphasis on the problem of protecting LEO spacecraft, and so this will be the focus of the Ph.D. Impact risks are still an issue for GEO spacecraft though, particularly from micrometeoroids (whose velocities can reach the order of 70 km/s).

The hazard posed to operational spacecraft by the growing debris population is now acknowledged at the very highest levels internationally, for example by the UNCOPUOS Scientific and Technical Subcommittee. In particular, it is recognised that a hypervelocity impact with even a millimetre-size particle could potentially result in partial or complete loss of a mission. More seriously, such damage may also prevent the implementation of planned passivation measures or post-mission disposal options. To mitigate the possibility of these occurring, UNCOPUOS (1999) recommends the following action.

Recommendation: Spacecraft designers should consider incorporating implicit and explicit protection concepts into their space vehicles.

It is important to understand the existing methods available to help designers respond to this recommendation, before any new protection strategies can be proposed. This chapter identifies some typical effects of debris impact on spacecraft; summarises the design and performance of state-of-the-art shielding options; reviews the current methodology and tools for assessing and reducing impact risks; and lists some general guidelines that have been recommended for the implementation of protection.

3.2 Effects of Hypervelocity Impact on Spacecraft

The effect of a hypervelocity particle impacting a spacecraft is, not surprisingly, dependent on impact energy. For the purposes of considering impact effects on a spacecraft design it is useful to distinguish three debris size ranges: < 0.1 mm, 0.1 - 10 mm, and > 10 mm.

A debris object smaller than ~0.1 mm in size represents a very low penetration hazard to a spacecraft, but because the population of such objects is so large in LEO (several orders of magnitude greater than the trackable population), multiple impacts can occur. Over the mission life these impacts can cause an accumulation of minor damage to spacecraft surfaces, such as surface pitting and erosion. Evidence for this has been gathered from several missions where exposed surfaces on spacecraft such as LDEF, EURECA, and HST have been retrieved from space and examined. These showed significant numbers of small impact craters; in the case of LDEF more than 30,000 were observed. However, there was no discernible effect on the missions.

Synergistic environmental effects are another potential cause of damage to a spacecraft [Lu & Nahra (1991)]. For example, multiple small debris impacts can erode atomic oxygen (AO) protective coatings thereby allowing AO impingement. The material released from the resulting oxygen ion sputtering could then be a source of contamination.

For debris in the 0.1 mm to 10 mm size range, significant structural damage can occur. This might include penetration of exposed instruments located on the outside of a spacecraft. Penetration of the structure and damage to internal equipment is another distinct possibility. Both effects could lead to partial or complete loss of a mission. Christiansen *et al* (1993) have also shown that a homogeneous particle impacting a spacecraft surface at a highly oblique angle (> 65° with respect to the surface normal) can cause the particle to break-up and release fragments that are projected onto other surfaces, or back into space. A mathematical model of this ejecta phenomenon has recently been derived [Rival & Mandeville (1999)].

It is generally considered unlikely that a spacecraft will survive an impact with a particle larger than ~10 mm, mainly because of the penetrative damage caused. At the very least, the transfer of momentum may cause the spacecraft to lose attitude control. If the debris object is large enough, the post-impact stress waves could carry sufficient energy through the structure to cause a catastrophic break-up of a spacecraft. For this to occur, the ratio of impact energy to spacecraft mass has to exceed a certain value. As a ‘rule-of-thumb’ a figure of 40 J / gram has been suggested [US National Research Council (1995)].

To further understand the damage effects on a spacecraft, it is instructive to examine the consequences of impacts on a selection of typical subsystems (see Table 3-1).

Subsystem	Possible impact effects	References
Honeycomb panels (Carbon Fibre Reinforced Plastic (CFRP) facesheets, Aluminium honeycomb)	Hole through facesheet: □ Delamination around hole □ Reduced electromagnetic compatibility Honeycomb blast damage: □ Thermal distortion stresses □ Thermal conductive paths disrupted Release of ejecta and cloud: □ Damage to equipment both inside and outside the spacecraft	Taylor *et al* (1997) Turner *et al* (1999)
Solar panels	Sub-millimetre impactors: □ Erosion of surface coating □ Cracking of coverglass □ Penetration of cell □ Reduction of light energy transmission □ Severed wire leading to loss of cell string or circuit □ Production of plasma cloud, possibly leading to electro-static discharge and circuit burn-out	Gurule *et al* (1992) McDonnell *et al* (1997) Caswell (1998)
Pressure vessels / tanks (For storing propellant at low pressure, or storing inert gas at high pressure)	□ Stress concentration around entry hole can cause front side failure □ Spall fragments in liquid / gas contaminate tank, pipes, pumps □ Fragments can crater and perforate rear wall, and lead to crack growth and wall failure □ Transmission of shock wave through liquid / gas can impact rear wall causing bulge or failure □ Catastrophic rupture occurs when vessel is above a critical pressure □ A very reactive liquid may ignite	Schäfer *et al* (1997) Lambert (1990) Poe & Rucker (1993)
Steering/pointing mechanisms	Spall fragments released into motor casing may cause motor to jam	
Electrical harness	Severs wires thus causing short-circuit or current isolation	
Manned pressurised modules	Failure modes due to penetration: □ Crack growth in module wall, leading to decompression □ Uncontrolled thrust from air leak, leading to loss of attitude control and structural failure □ Damage to critical internal equipment □ Injury to crew from fragments, heat, light-flash, and over-pressure □ Hypoxia-induced crew unconsciousness and eventual loss (especially if hole is large, i.e. depressurisation is rapid)	Williamsen & Schonberg (1997) Destefanis & Callea (1993)
Windows / viewports	Impact on brittle glass produces a large area of radial and concentric cracking, thus reducing visibility. Visibility improved if glass is strengthened with acrylic or polycarbonate in a sandwich configuration	Lambert (1990)

Table 3-1 Some effects of hypervelocity impact on selected spacecraft subsystems

The extent to which a spacecraft is disrupted by an impact is of course driven by a number of subsystem design factors, for example:

- The function and criticality of the subsystems
- The presence of any subsystem redundancy
- The robustness of the subsystems to withstand impacts
- The location and alignment of the subsystems relative to the debris flux direction

Any decision to incorporate protection must therefore take full account of these factors if a cost-effective solution is to be implemented.

3.3 Spacecraft Shielding Design Options

The objective of a debris shield is to absorb as much of the localised damage-producing energy from an impacting particle, thereby minimising the risk of spacecraft failure. Any decision to add such shielding requires a careful balance of the acceptable risks versus the extra mass and cost. The shielding options for manned spacecraft are different to those for unmanned spacecraft, not least because of the higher safety standards required. A penetration of a pressurised manned vehicle can have serious consequences both for the safety of the crew onboard and for the integrity of the vehicle itself, which is usually a very high cost item. The political ramifications of losing a manned vehicle in this fashion are also incalculable. By contrast, the loss of a typical unmanned satellite through collision damage does not carry the same level of concern. Therefore, lower penetration risk requirements are usually acceptable. Consequently, shield designs for manned and unmanned spacecraft are different, and so they are considered separately here. This section also briefly discusses shielding options for two important elements on typical spacecraft, namely solar arrays and pressure vessels.

3.3.1 Structure Shielding for Manned Spacecraft

Manned spacecraft debris protection can be enhanced by adding dedicated shielding. Typically, this is achieved by implementing a multiple wall shield. Such a shield is considered an efficient method of improving protection, and usually involves spacing one or more thin bumper layers in front of the spacecraft structure, or back-up, wall. When a particle impacts the outer bumper of the shield a pressure wave is generated in both materials. These waves cause the particle and part of the bumper to shatter into a cloud of small particles, which may be in a solid, liquid, or vaporous state. This debris cloud expands radially as it travels towards the next wall, thus spreading its momentum and energy over a larger area. If several bumper layers are used then there is a progressive shocking of the cloud, which further reduces the momentum per unit area. The back-up wall therefore experiences less penetrative damage than if one single large particle had impacted directly onto the wall, i.e. the wall is less likely to be perforated. However, if the shield and back-up wall is perforated, the diameter of the back-up wall hole will be much larger, which poses a greater threat to the crew [Williamsen *et al* (1999)]. Clearly, there is a trade-off here that must be addressed when designing a shield for a manned spacecraft.

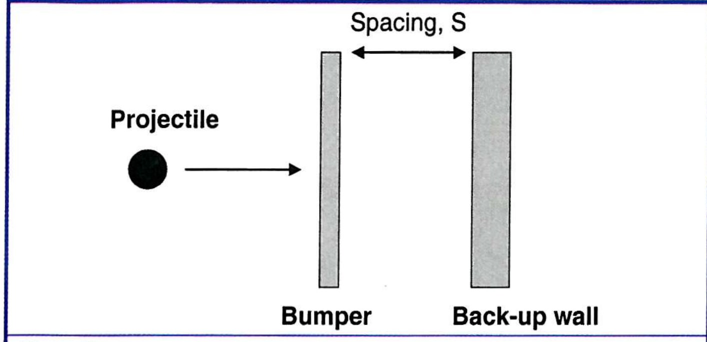
The effectiveness of a multiple wall shield is dependent on several factors, including impactor properties (mass, density, velocity, and shape), the material in each bumper, and the arrangement of the bumpers. Current shielding technology is limited to protecting against objects in LEO up to approximately 1 cm in diameter. Several multiple wall shields have been developed and tested for manned spacecraft over the years. Christiansen (1993) identifies their advantages over a single wall, as follows:

- Improved performance, i.e. the same level of protection is achieved using less weight.
- Less damaging secondary ejecta is released following an oblique impact, since smaller particles are produced.
- More efficient at converting impactor kinetic energy into internal thermal energy.
- Less sensitive to impactor shape because the multiple shocks are much more disruptive.
- Less sensitive to oblique impacts, partly due to the increased fragmentation from the multi-shocking phenomenon, and partly because the walls repeatedly slow the expansion of the debris cloud.
- Less cumulative damage to the back-up wall over the life of a long duration mission, since fewer small particles will be likely to penetrate the bumper layers, thus causing crater damage to the back-up wall.

The earliest, and simplest, multiple wall shield concept was proposed by Whipple in 1947. Since then, the desire for improved penetration resistance, particularly for the International Space Station, has resulted in several new variants. These usually fall into one of the following categories: stuffed Whipple shield, multi-shock shield, and mesh double-bumper. A common feature of these shields is their utilisation of new high strength fabric materials such as Nextel and Kevlar. A number of researchers have investigated the correct configuration and combination of materials for maximum protection. The generic designs of the four shield types are given below.

Whipple Shield

The Whipple shield comprises a single thin bumper placed a distance S in front of a thicker back-up wall, as shown in Figure 3-1.



Typical bumper materials:

- Aluminium alloy
- Kevlar
- Graphite/epoxy
- Metal matrix composite
- Ceramic matrix composite
- Corrugated aluminium

Function:

- Bumper disrupts an impactor, creating a debris cloud
- Maximum effectiveness achieved when:
  - Bumper thickness / projectile diameter ratio ~0.2
  - Spacing S is ~30 times projectile diameter
- Back-up wall resists impulsive loading

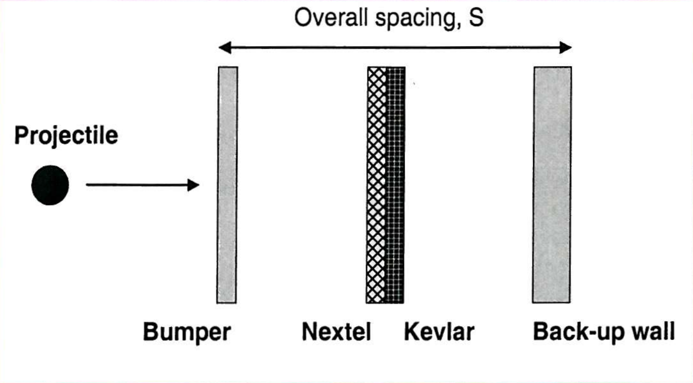
References:

- Christiansen (1993)
- Lambert (1990)
- McGill & Mount (1992)
- Robinson & Nolen (1995)
- Schonberg (1990)
- Schonberg *et al* (1991)

Figure 3-1 Whipple shield

Stuffed Whipple Shield

The stuffed Whipple shield comprises an outer bumper spaced a distance S from a back-up wall, with an intermediate bumper consisting of ceramic cloth back by high-strength cloth, as illustrated in Figure 3-2.



Typical outer bumper materials:

- Aluminium alloy
- Kevlar
- Graphite/epoxy
- Metal matrix composite
- Ceramic matrix composite
- Corrugated aluminium

Typical inner bumper materials:

- Nextel
- Kevlar
- Spectra

Function:

- Outer mesh bumper disrupts an impactor, creating a debris cloud
- Inner bumper decreases impulsive loading of debris cloud, i.e. it acts as a 'catcher'
- Back-up wall resists impulsive loading

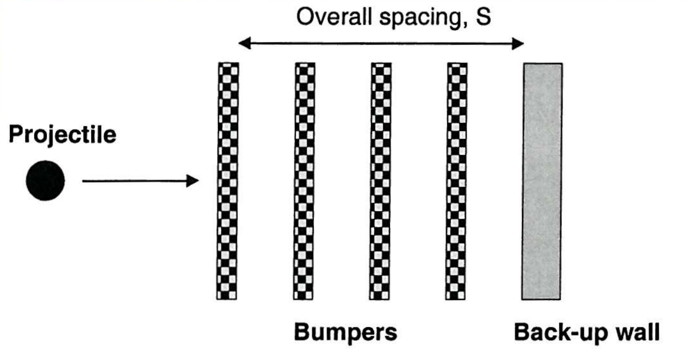
References:

- Christiansen *et al* (1995)
- Shiraki *et al* (1997)
- Beruto *et al* (1997)
- Destefanis & Faraud (1997)
- Lambert & Schneider (1995)

Figure 3-2 Stuffed Whipple shield

Multi-Shock Shield

The multi-shock shield comprises several equally spaced ultra-thin bumper layers in front of a back-up wall, as shown in Figure 3-3.



Typical bumper materials:

- Aluminium
- Nextel

Function:

- Bumpers successively shock impactor, raising its thermal state, and causing it to melt and vaporise
- Maximum effectiveness achieved when:
  - Bumper thickness / projectile diameter ratio <~0.03
- Back-up wall resists impulsive loading

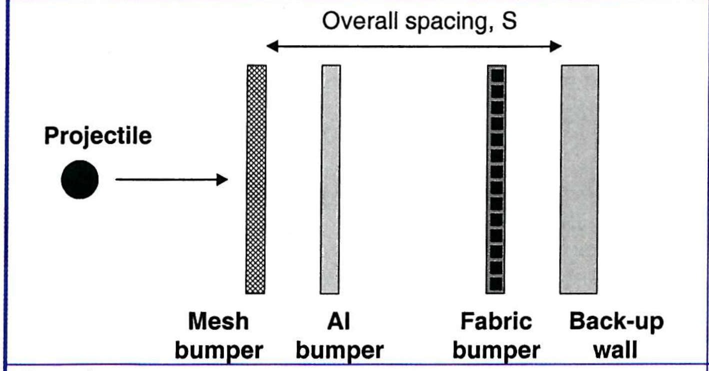
References:

- Christiansen (1993)

Figure 3-3 Multi-shock shield

Mesh Double-Bumper Shield

The mesh double-bumper shield comprises an outer mesh bumper, an aluminium second bumper, a third bumper made from high-strength fabric, and a back-up wall (see Figure 3-4).



Typical bumper materials:

- Aluminium mesh
- Aluminium
- Nextel
- Kevlar
- Spectra

Function:

- Outer mesh bumper disrupts an impactor, creating a debris cloud
- Second bumper melts / vapourises remaining fragments
- Fabric bumper decreases impulsive loading of debris cloud, i.e. it acts as a 'catcher'
- Back-up wall resists impulsive loading

References:

- Christiansen & Kerr (1993)

Figure 3-4 Mesh double-bumper shield

3.3.2 Structure Shielding for Unmanned Spacecraft

Standard Structures

Many unmanned satellites incorporate honeycomb (HC) panels and multi layer insulation (MLI) thermal blankets in their structures (see Figure 3-5). Both of these can be considered as non-dedicated shields, since their primary function is to provide structural and thermal integrity, respectively.

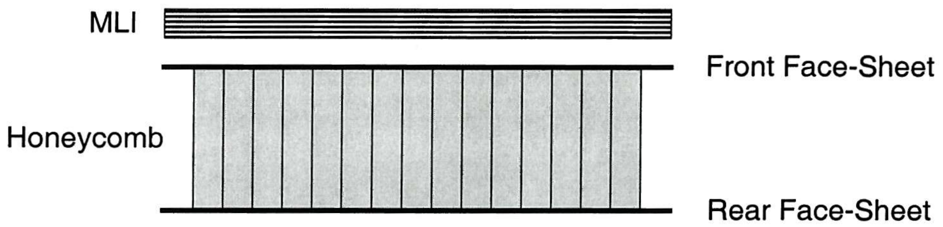


Figure 3-5 Standard single-honeycomb panel structure with MLI

The shielding performance of a honeycomb panel is highly dependent on the angle of incidence of an impactor. At oblique angles, the cells may act as a multi-shock shield, thus increasing the penetration resistance [Turner *et al* (1999)]. However, it is noted that a 'channelling' effect can also occur. That is, a penetrating particle creates a debris cloud, which is channelled by the honeycomb cells thereby increasing the areal damage to the rear face-sheet.

MLI can reduce the penetration damage from sub-millimetre size particles. Typically, it comprises 10 to 25 layers of Mylar interleaved with Dacron net, and covered with a layer of Kapton or Teflon-impregnated Beta-cloth. MLI can provide better protection than an equivalent thickness of aluminium, which suggests that it may possess a multi-shock shielding characteristic [Turner *et al* (1999)].

Enhanced Structures

The provision of additional shielding for unmanned spacecraft structures cannot rely on the techniques used for manned spacecraft, as described in Section 3.3.1. The mass, volume and cost of such shields are an order of magnitude too high. Instead, one is restricted to enhancing the design of existing honeycomb panels and/or MLI blankets. Therefore, it will not be possible to prevent 1 cm size LEO debris particles from penetrating; in fact a more achievable size limit is ~2 mm [Turner *et al* (1999)].

The wealth of information on manned shielding techniques does, however, help us to consider credible options for enhancing honeycomb panels and MLI. For example, adding layers of high-strength materials, such as Nextel and Kevlar, may provide a similar shielding effect to the stuffed Whipple shield. Table 3-2 lists the most likely options for adding shielding to unmanned spacecraft [Turner *et al* (1999)].

	Shield type	**Expected shielding benefit**
1	Change thickness of HC cell walls	Alter the debris cloud channelling effect. Angle dependency
2	Change HC cell size	Alter the debris cloud channelling effect. Angle dependency
3	Increase HC depth	Increase debris cloud dispersion
4	Vary outer face-sheet thickness	Increased projectile disruption
5	Vary inner face-sheet thickness	Increased tolerance to debris cloud loading
6	Use multiple HC layers	Similar effect to multiple wall shield
7	HC+laminate	Different shock impedances to increase projectile disruption
8	MLI+Nextel/Kevlar/Beta front layer	Multi-shock shield effect & increased projectile disruption
9	MLI+Nextel/Kevlar/Beta rear layer	Multi-shock shield effect & increased debris cloud containment
10	MLI+Nextel/Kevlar/Beta mid layer	Multi-shock shield effect & increased debris cloud disruption
11	MLI+spacer rods	Multi-shock shield effect
12	HC+Nextel/Kevlar inside spacecraft	Protects sensitive internal spacecraft equipment
13	HC+spaced bumper	Whipple bumper shield effect

Table 3-2 Shielding options for unmanned spacecraft [Turner et al (1999)]

Turner *et al* (1999) have investigated the effectiveness of several of these. In particular, it was determined that a double-honeycomb design, as shown in Figure 3-6, could potentially reduce the number of penetrating particles by a factor of ~4, at an additional 'cost' of only 1.2 kg/m^2, compared to a standard single-honeycomb panel. Double-honeycomb panels are particularly recommended for the most vulnerable spacecraft surfaces, e.g. those facing the velocity direction.

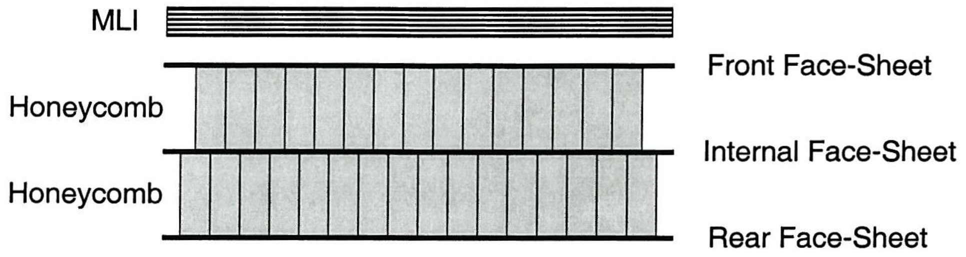


Figure 3-6 Double-honeycomb panel structure with MLI [Turner et al (1999)]

The possibility of utilising aluminium foam, as a low-cost alternative to traditional honeycomb panels has recently been suggested [Banhart (1999), Scott (1999)]. A metal foam is a "cellular solid", i.e. gas bubbles are dispersed in a molten metal, which is then allowed to solidify. A typical foam can be up to 15 times thicker than a single homogeneous wall of identical mass. Its property of high energy absorption per unit mass or volume suggests that it may be suitable as an impact resistant spacecraft structure. That is, a foam could be effective at breaking an impactor into fine vaporous particles. However, further research is required into this alternative.

3.3.3 Solar Array Shielding

Because of its large surface area and exposed location, one of the most vulnerable items on a spacecraft is a solar array. Several different shielding approaches have been suggested for deployable arrays [Turner *et al* (1999)]:

- ❑ Use thin, flexible arrays so that a penetrating particle may pass through causing relatively little damage, or
- ❑ Alternatively, use toughened and/or laminated solar cells on rigid (honeycomb) panels to absorb the impact energy.
- ❑ Consider adding MLI to the back of a panel to limit rear-side damage (provided the resulting temperature increase does not reduce power output by too much).
- ❑ Consider covering the working surfaces with a foam or gel (e.g. Aerogel). This should also have a very low dielectric constant and density. Again, power output reductions need to be addressed with this option.
- ❑ Use multiple electrical paths in the solar array design.

3.3.4 Pressure Vessel Shielding

Much research has been done recently to mitigate the potentially catastrophic consequences of an impact on a pressure vessel such as a spacecraft fuel tank. Investigations by Schäfer *et al* (1997) show that a simple measure for preventing rupture of a pressurised vessel is to reduce its working pressure. If high pressures are required, then a simple bumper plate shield spaced several centimetres in front of the vessel is an effective means of enhancing protection. Schäfer (1998) reports that shield effectiveness increases significantly when the shield / vessel spacing is increased. MLI is observed to be an excellent shield, with high ballistic efficiency. Another option for decreasing the likelihood of a fuel tank leak or rupture may be to utilise a gelled fuel [Spinks (2001)]. Composite overwrapped pressure vessels (Kevlar, graphite/epoxy, etc) reduce the size and hazard from tank fragments if rupture does occur. Internal shields might also be considered to deflect the blast from ruptured tanks away from other critical parts of a spacecraft. Finally, in terms of operational use, the most vulnerable tanks on a spacecraft should be depleted first. Once depleted, the tanks should be depressurised to reduce the risk of catastrophic rupture should an impact occur [Christiansen *et al* (1992)].

3.4 Damage Equations

The characterisation of hypervelocity debris and meteoroid impacts on a spacecraft is an important element of its environmental risk analysis. Empirical damage equations are used to determine not only the size of an impact crater or hole on a spacecraft surface (i.e. structure or shield), but also its ballistic limit. The ballistic limit is the critical size of particle that causes the structure or shield to fail. Failure can be defined in terms of perforation or spallation. To date, two distinct types of ballistic limit equation have been derived to characterise the two possible types of spacecraft structure / shield design, namely single wall and multiple wall.

Damage equations are most commonly used in damage assessment tools (cf. Section 3.5.3), particularly to assess the probability of penetration of a spacecraft structure / shield from particles that exceed the ballistic limit. Care must be taken when using the equations, particularly to understand their limitations. For example, all equations listed here have been derived for spherical impactors, i.e. particle shape effects have been ignored. Since typical debris particle shapes are likely to be more damaging than equivalent spherical particles

[Schäfer (1999)], the modification of equations to include shape effects is a possibility in the future.

3.4.1 Damage Assessment Methods

Two-stage light-gas guns (LGG) are normally used to determine the impact damage to a structure over the impact velocity range 2 km/s to 7 km/s. Typically, a LGG achieves these speeds by first burning a powder charge to accelerate a piston, which then compresses a gas of low-molecular mass, such as hydrogen. This so-called light gas is then released very suddenly, by rupturing a diaphragm, thus causing the acceleration of a projectile which is directed at a target. Several two-stage LGG facilities are currently available in Europe, including at the University of Kent, UK [Burchell *et al* (1999)]; the Ernst Mach Institute, Germany [Stilp (1987)]; and the Centre of Studies and Activities for Space, Italy [De Cecco & Pavarin (1998)]. In order to reach the impact velocities typical of debris in LEO, i.e. > 7 km/s, some facilities are developing ultra-high speed launchers. These include explosively launched projectiles [Bol & Fucke (1997)], 3-stage launchers [Bourne *et al* (2002)], and other techniques such as the LGG counterfire method [Stilp (1997)].

Hydrocodes, analytical models, semi-empirical approaches and other analysis techniques are also used to formulate and/or verify damage equations, particularly for impact velocities above 7 km/s. These physics-based approaches attempt to predict the response of a material under hypervelocity impact. The interaction is usually very complex. At the instant of impact, pressures can exceed 50 GPa with strain rates exceeding 10^6 s^{-1} [Jonas *et al* (1993)]. Under these conditions, the strength of the material can be ignored and the impact process treated as hydrodynamic. During the impact process, the extreme pressures rapidly reduce to levels where the forces are comparable to the material strength. At this point, material properties, which are usually strain rate dependent, have to be considered.

Hydrocodes are finite element and finite difference computational codes, which have been developed over the last thirty years to simulate impacts under conditions that are too difficult or too costly to reproduce using test facilities. They can encompass a number of different numerical approaches for the analysis of impact problems, such as Lagrange, Euler, Arbitrary Lagrange Euler, Shells, and Smooth Particle Hydrodynamics [Birnbaum *et al* (1996)]. Each has its own advantage and disadvantage depending on the impact application. To model the materials in a simulation, hydrocodes use Equations of State (EOS) and constitutive equations. These have been developed for commonly used spacecraft materials such as aluminium, but until recently no such equations existed for the newer lightweight shielding materials such as Nextel and Kevlar. These new equations have now been incorporated into Autodyn, which is one of the most widely used hydrocodes [Hayhurst *et al* (1998, 1999)].

3.4.2 Definition of Damage Equation Parameters

The damage equations presented throughout the remainder of this section comprise impactor dependent terms such as size and velocity, and spacecraft structure (i.e. target) terms such as material properties. Table 3-3 defines the various terms in the equations, which are subsequently presented [Lemcke *et al* (1998a)].

Symbol	Unit	Description
c_s	kms^{-1}	Speed of sound in target
d_p	cm	Particle (impactor) diameter
m	kg	Mass of particle
p	cm	Crater depth
t_t, t_B, t_s	cm	Thickness of target, back-up wall, shield (total)
$\mathrm{t_{Equiv}}$	cm	Equivalent aluminium single wall thickness
v, v_n	kms^{-1}	Impact velocity, normal component of impact velocity
D_c, D_h	cm	Diameter of crater, hole
E	J	Impact energy
F_{mx}	cm	Ballistic limit
H_t		Brinell hardness of target
K		Characteristic factor. $\mathrm{K_f}$: failure factor; $\mathrm{K_0}$, $\mathrm{K_1}$, $\mathrm{K_2}$: equation specific factors $\mathrm{K_c}$: crater factor (ratio of crater radius to crater depth)
Q	Coulombs	Charge released by an impact
S	cm	Space between shielding and back-up wall
α	degrees	Impact angle (with respect to surface normal)
ρ_t, ρ_p, ρ_s, ρ_B	gcm^{-3}	Density of target, particle, shield (average), back-up wall
τ_1^*	Pa	Yield stress of reference material 1, = 276 x 10^6 Pa (= 40,000 lb in^{-2})
τ_2^*	Pa	Yield stress of reference material 2, = 483 x 10^6 Pa (= 70,000 lb in^{-2})
τ	Pa	Yield stress of back-up wall

Table 3-3 Damage equation terms [Lemcke et al (1998a)]

The literature contains a large number of equations, which cannot possibly be listed here in their entirety. Instead, a selection of popular equations that are included in ESA's ESABASE / DEBRIS damage prediction code (cf. Section 3.5.3) are reproduced, since these are the most widely distributed throughout Europe. For further equations, the reader is invited to refer to Berthoud & Mandeville (1993), Christiansen (1993), Christiansen *et al* (1995), McDonnell (1999), Reimerdes *et al* (1993), and others.

3.4.3 Plasma Production Damage

A reliable theoretical model of the charge produced by a small hypervelocity particle impacting a spacecraft surface, such as a solar array, does not yet exist. Most investigators quote empirical models of Q, the total amount of charge released by the impact. The plasma yield from an impact has a characteristic dependence on mass and velocity of the form [Ratcliff *et al* (1995)]:

$$Q = km^{\beta}v^{\gamma} \qquad [3\text{-}1]$$

where Q is in Coulombs, m in kilograms, v in kms^{-1}, and k is a constant. β and γ are variables that depend on m and v. Typically, $\beta \sim 0.8 - 1.0$ and $\gamma \sim 3.5 - 4.8$. Ratcliff *et al* (1995, 1997) and McDonnell *et al* (1997) have conducted experiments to determine the plasma yields resulting from various impactor / target combinations.

3.4.4 Single Wall Crater Damage

A non-perforating spherical projectile impacting a ductile target produces a near-hemispherical crater of size D_c (see Figure 3-7a). The same projectile impacting a brittle target will normally produce an interior crater with diameter, D_h, inside a larger outer crater of size D_c, as shown in Figure 3-7b [Lemcke *et al* (1998a)].

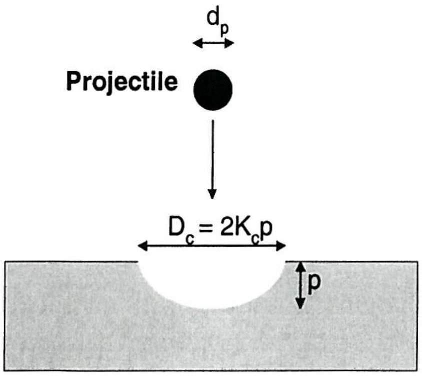


Figure 3-7a Impact on ductile target [Lemcke et al (1998a)]

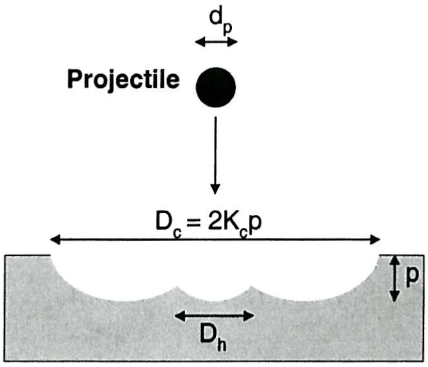


Figure 3-7b Impact on brittle target [Lemcke et al (1998a)]

Most spacecraft comprise a large percentage of aluminium alloy and other monolithic metallic surfaces, which fall into the category of 'ductile' materials. Typical 'brittle' surfaces on a spacecraft include solar arrays, mirrors on optical telescopes, and shuttle window glass.

Impact crater size on a single wall is characterised by the following equation [Lemcke *et al* (1998a)]:

$$D = K_c K_1 d_p^{\lambda} \rho_p^{\beta} v^{\gamma} (\cos\alpha)^{\xi} \rho_t^{\kappa} \qquad [3\text{-}2]$$

Typical values for the parameters in this equation are listed in Table 3-4 [Lemcke *et al* (1998b)].

Equation	K_c	K_1	λ	β	γ	ξ	κ
Ductile Materials							
ESABASE Thick Plate	2	0.606	1.056	0.519	2/3	2/3	0
Christiansen, $\rho_p/\rho_t < 1.5$	--	$10.5 \cdot H_t^{-1/4} \cdot c_s^{2/3}$	1.056	0.5	2/3	2/3	-0.5
Christiansen, $\rho_p/\rho_t > 1.5$	--	$10.5 \cdot H_t^{-1/4} \cdot c_s^{2/3}$	1.056	2/3	2/3	2/3	-2/3
Brittle Materials							
Gault	--	1.08	1.071	0.524	0.714	0.714	-0.5
Fechtig	--	6.0	1.13	0.71	0.755	0.755	-0.5
McHugh & Richardson	--	1.28	1.2	0	2/3	2/3	0.5
Cour-Palais	--	1.06	1.06	0.5	2/3	2/3	0

Table 3-4 Typical parameter values for the crater equation [Lemcke et al (1998b)]

3.4.5 Single Wall Hole Damage

For a target that is fully perforated by an impacting particle, the damage is characterised by the following hole diameter equation [Lemcke *et al* (1998a)]:

$$D = \left[K_0 \left(\frac{t_s}{d_p} \right)^{\lambda} \rho_p^{\beta} v^{\gamma} (\cos\alpha)^{\xi} \rho_s^{\upsilon} + A \right] d_p \qquad [3\text{-}3]$$

Typical values for the parameters in this equation are listed in Table 3-5 [Lemcke *et al* (1998b)].

Equation	K_0	λ	β	γ	ξ	υ	A
Maiden	0.88	2/3	0	1	1	0	0.9
Nysmith-Denardo	0.88	0.45	0.5	0.5	0.5	0	0
Sawle	0.209	2/3	0.2	0.2	0.2	-0.2	1
Fechtig	5.24E-5	0	1/3	2/3	2/3	0	0

Table 3-5 Typical parameter values for the clear hole equation [Lemcke et al (1998b)]

Investigations by Taylor *et al* (1997) show that the equivalent hole diameter, created by a hypervelocity particle penetrating a Carbon Fibre Reinforced Plastic (CFRP) / Al honeycomb panel, is related to impact energy as follows:

$$D_{eq(total)} \propto E^{1/3} \qquad [3\text{-}4]$$

3.4.6 Single Wall Ballistic Limit Equation

As a rough rule-of-thumb, a wall that is at least four times thicker than the depth of a crater produced by an impacting particle can be considered semi-infinite (or thick). In this instance, the rear side of the wall does not influence crater formation. For a finite (or thin) wall, although the crater depth is still smaller than the wall's thickness, rear-side wall damage and spallation can occur as a result of the tension waves exceeding the wall's material strength. A ductile thin wall will produce spall when the crater depth is ~1/3 of the wall thickness. Perforation occurs when the depth is ~1/2 the wall thickness [Lambert (1990)].

The single wall ballistic limit equation defines the penetration performance of a homogeneous structure such as an aluminium plate or a layer of glass. It has the following traditional form [Lemcke *et al* (1998a)]:

$$F_{mx} = K_f K_1 d_p^{\lambda} \rho_p^{\beta} v^{\gamma} (\cos\alpha)^{\xi} \rho_t^{\kappa} \qquad [3\text{-}5]$$

For implementation purposes, this is usually recast as follows [Lemcke *et al* (1998a)]:

$$d_{p_{crit}} = \left[\frac{t_t}{K_f K_1 \rho_p^{\beta} v^{\gamma} (\cos\alpha)^{\xi} \rho_t^{\kappa}} \right]^{\frac{1}{\lambda}} \qquad [3\text{-}6]$$

Typical values for the parameters in this equation are listed in Table 3-6 [Lemcke *et al* (1998b)].

Equation	K_f	K_1	λ	β	γ	ξ	κ
ESABASE thick plate	0.6	0.606	1.056	0.519	2/3	2/3	0
ESABASE thin plate	1.0	0.406	1.056	0.519	0.875	0.875	0
Pailer-Grün	1.0	0.77	1.212	0.737	0.875	0.875	-0.5
Frost	1.0	0.43	1.056	0.519	0.875	0.875	0
Naumann, Jex, Johnson	1.0	0.65	1.056	0.5	0.875	0.875	-0.5
Naumann	1.0	0.326	1.056	0.499	2/3	2/3	0
McHugh & Richardson Thick glass target	0.265	0.64	1.2	0	2/3	2/3	0.5
Cour-Palais Thick glass target	0.265	0.53	1.06	0.5	2/3	2/3	0

Table 3-6 Typical parameter values for the single wall ballistic limit equation [Lemcke et al (1998b)]

3.4.7 Multiple Wall Ballistic Limit Equation

The traditional form of the multiple wall ballistic limit equation is as follows [Lemcke *et al* (1998a)]:

$$F_{mx} = K_1 d_p^{\lambda} \rho_p^{\beta} v^{\gamma} (\cos\alpha)^{\xi} \rho_B^{\kappa} S^{\delta} \rho_s^{\upsilon 1} - K_2 t_s^{\mu} \rho_s^{\upsilon 2} \qquad [3\text{-}7]$$

In its implemented form, this becomes [Lemcke *et al* (1998a)]:

$$d_{p_{crit}} = \left[\frac{t_B + K_2 t_s^{\mu} \rho_s^{\upsilon 2}}{K_1 \rho_p^{\beta} v^{\gamma} (\cos\alpha)^{\xi} \rho_B^{\kappa} S^{\delta} \rho_s^{\upsilon 1}} \right]^{\frac{1}{\lambda}} \qquad [3\text{-}8]$$

In using this equation three velocity regions have to be considered, which are determined by the three damage mechanism scenarios characteristic of a particle impacting a multiple wall structure. At velocities below ~3 km/s, the ballistic region, the material strength of a projectile exceeds the dynamic pressure at impact, and so the projectile is poorly fragmented by a bumper, i.e. a very small number of solid particles are released. Between ~3 km/s and ~7 km/s, a projectile is broken into several pieces, which may be solid and/or vaporous. This is the shatter or transition region. Above ~7 km/s, the true hypervelocity range, pressures far exceed the material strength of a projectile thus causing it to break up into a dense cloud of numerous fine vaporous particles.

The three velocity ranges are denoted by (1) $v_n < v_l$, (2) $v_l \leq v_n \leq v_u$, and (3) $v_n > v_u$; where v_n is the normal impact velocity, and v_l, v_u are the lower and upper transition velocities between the three regions. For impact velocities in the transition region, i.e. between v_l and v_u, linear interpolation is often used to calculate the critical particle diameter, as follows:

$$d_{p_{crit}} = \left(\frac{v_u - v_n}{v_u - v_l} \right) d_{p_{crit}, v_n = v_l} + \left(\frac{v_n - v_l}{v_u - v_l} \right) d_{p_{crit}, v_n = v_u} \qquad [3\text{-}9]$$

Typical values for the parameters in the multiple wall equation are listed in Table 3-7 [Lemcke *et al* (1998b)].

Equation	v_n (km/s)	K_1	K_2	λ	β	γ	κ	δ	ξ	υ_1, υ_2	μ
ESA triple	$v_n < 3$	$0.312[\tau_1^*/\tau]^{0.5}$	$1.667 \cdot K_1$	1.056	0.5	2/3	0	0	5/3	0, 0	1
	$v_n > 7$	$0.107[\tau_2^*/\tau]^{0.5}$	0	1.5	0.5	1	0	-0.5	1	0.167, 0	0
Modified Cour-Palais (NASA ISS)	$v_n < 3$	$0.6[\tau_1^*/\tau]^{0.5}$	$1.667 \cdot K_1$	1.056	0.5	2/3	0	0	5/3	0, 0	1
	$v_n > 7$	$0.129[\tau_2^*/\tau]^{0.5}$	0	1.5	0.5	1	0	-0.5	1	0.167, 0	0
NASA shock	$v_n < 3$	$0.3[\tau_1^*/\tau]^{0.5}$	$1.233 \cdot K_1$	1.056	0.5	2/3	0	0	5/3	0, 1	1
	$v_n > 6$	$22.545[\tau_1^*/\tau]^{0.5}$	0	3	1	1	-1	-2	1	0, 0	0
NASA bumper	$v_n < 3$	$0.4[\tau_1^*/\tau]^{0.5}$	$0.925 \cdot K_1$	1.056	0.5	2/3	0	0	5/3	0, 1	1
	$v_n > 6$	$18.224[\tau_1^*/\tau]^{0.5}$	0	3	1	1	-1	-2	1	0, 0	0

Table 3-7 Typical parameter values for the multiple wall ballistic limit equation [Lemcke et al (1998b)]

3.4.8 Honeycomb Panel Ballistic Limit Equation

The ballistic limit of a honeycomb structure is a function of face-sheet thickness, cell dimensions and cell wall thickness, core depth, and the materials used. No equation currently exists that combines these honeycomb parameters. Instead, to date, researchers have concentrated on making modifications to the parameters in the multiple wall equation.

Taylor *et al* (1998) have applied the modified Cour-Palais equation to characterise the ballistic limit of a CFRP / Al single-honeycomb panel. The equation can be applied conservatively to particles impacting at angles of incidence greater than 15°. However, because of the increased channelling effect at near-normal incidence angles (i.e. < 15°), the back-up wall thickness term in the equation has to be modified by a scaling factor of 0.5.

For Al / Al single and double honeycomb panels, Turner *et al* (1999) recommend using the ESA triple wall equation in Table 3-7, but with the parameters listed in Table 3-8.

Equation	v (km/s)	K_1	K_2	λ	β	γ	κ	δ	ξ	υ_1, υ_2	μ
ESA triple	$v_n < 3$	$0.312[\tau_1^* / \tau]^{0.5}$	$1.667 \cdot K_1$	1.056	0.5	2/3	0	0	8/3	0, 0	1
	$v_n > 7$	$0.107[\tau_2^* / \tau]^{0.5}$	0	1.5	0.5	1	0	-0.5	1	0.167, 0	0

Table 3-8 Typical parameter values for the Al/Al honeycomb panel ballistic limit equation [Turner et al (1999)]

It should be noted that for velocities above 7 km/s, Turner *et al* suggest replacing the back-up wall (i.e. rear face-sheet) thickness term, t_B, in the equation with an equivalent thickness of aluminium, t_{Equiv}, for the entire panel.

3.5 Methodology and Tools for Assessing and Reducing Impact Risk

3.5.1 General Approach for Assessing and Reducing Impact Risk

The US National Research Council (1995) advocates the early consideration of impact risks, as this will minimise the costs of any necessary spacecraft design modifications. Christiansen (1998) has defined a standard risk assessment methodology, as shown in Figure 3-8.

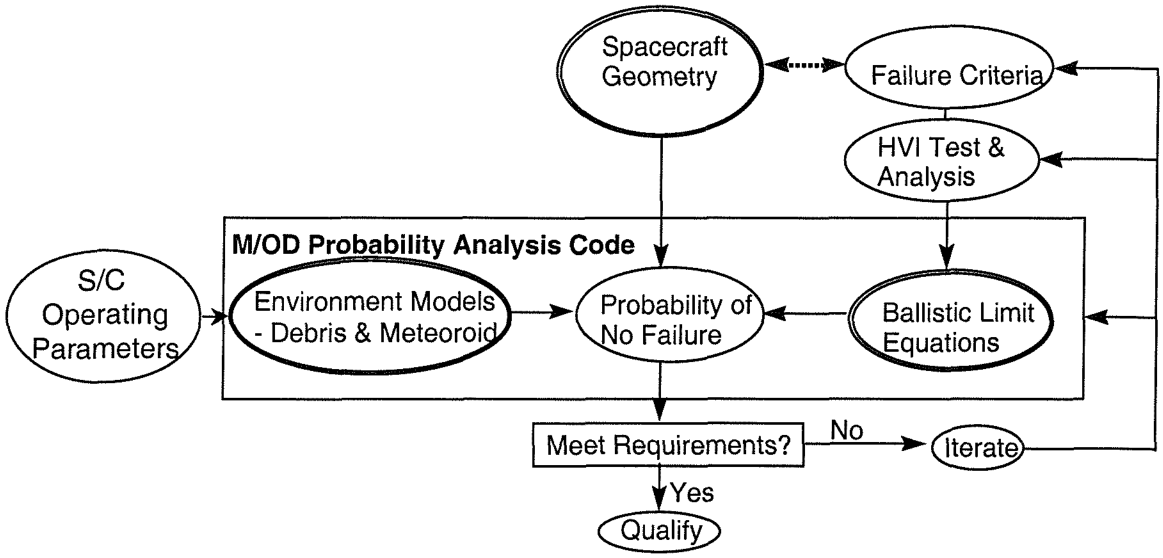

Spacecraft Geometry
Failure Criteria
HVI Test & Analysis
M/OD Probability Analysis Code
S/C Operating Parameters
Environment Models - Debris & Meteoroid
Probability of No Failure
Ballistic Limit Equations
Meet Requirements?
No
Iterate
Yes
Qualify

Figure 3-8 Standard impact risk assessment methodology [Christiansen (1998)]

Figure 3-8 shows that the procedure for assessing and reducing spacecraft impact risks is an iterative one. Each step in the procedure is briefly listed below:

- ❑ Define spacecraft design, operations, failure modes and effects.
- ❑ Identify vulnerable spacecraft components/subsystems.
- ❑ Assess hypervelocity impact damage modes for each system and component.
- ❑ Determine failure criteria for each system and component.
- ❑ Perform hypervelocity test/analysis to define ballistic limits of spacecraft.
- ❑ Conduct failure probability analysis using damage assessment tools
- ❑ Compare failure probability analysis results with goal or requirement.
- ❑ Consider updates to design, operations, analysis, test, or failure criteria, e.g.:
    - Revise analysis assumptions.
    - Refine accuracy of ballistic limit equations with additional testing.
    - Change the spacecraft design, starting with riskiest areas first.
    - Change the spacecraft orientation, so least vulnerable surface faces velocity direction.
- ❑ Update/Iterate as necessary to meet requirement.

3.5.2 Examples of Spacecraft Risk Assessment and Reduction

Although, meteoroids are routinely considered during a spacecraft design, few satellite operators fully appreciate how vulnerable their spacecraft are to orbital debris. Commercial satellite developers typically do fairly coarse impact risk assessments. However, this is starting to change as industry awareness of the problem grows. Some satellite designers now perform a 'head-on' impact analysis for their spacecraft structures to determine the penetration risk. Unfortunately, this is too simplistic since most impacts occur at oblique angles, and can potentially be more damaging [Scott (1999)].

The International Space Station (ISS) is probably the most well known example of a manned spacecraft that has followed the risk assessment methodology in Section 3.5.1, and subsequently incorporated debris protection. As a result, extensive shielding, of the types discussed in Section 3.3, is being utilised to reduce the penetration risk. Consideration has also been given to crew operations, equipment locations, and operational procedures to significantly reduce the likelihood of crew or station loss following an orbital debris penetration [Williamsen *et al* (1999)].

The Canadian Radarsat satellite is a good example of the application of the risk assessment methodology to an unmanned spacecraft. This resulted in enhancements to the structure and reconfiguration of subsystems [Terrillon *et al* (1991)]. For example, MLI was enhanced with Nextel to provide extra shielding to exposed electronics units and harnessing. Other measures included widening and thickening radiators to shield exposed boxes; thickening the walls of units considered to be vulnerable; extending honeycomb panels and aluminium plates to protect exposed hydrazine propulsion lines; and re-routing propulsion lines behind existing structures.

3.5.3 Damage Assessment Tools

The risk of failure of a spacecraft due to the debris and meteoroid hazard is determined using information on:

- □ the spacecraft geometry
- □ ballistic limit equations for the spacecraft structures and shields
- □ directional collision fluxes derived from debris and meteoroid environment models

By necessity, the complexity of this information requires the use of software models to perform the risk calculations. To calculate the probability of impact, Poisson statistics are used as follows:

$$P_{imp} = 1 - e^{(-N_{imp})} \tag{3-10}$$

where

$$N_{imp} = (flux)(area)(time) \tag{3-11}$$

That is, N_{imp} is the average number of impacts expected on a given area over a given time. The collision flux is directional. For a typical LEO satellite in a circular polar orbit, debris is most likely to strike in the orbital plane at an azimuth angle of +/- 30 - 45 degrees to the velocity vector, as shown in the example in Figure 3-9 [Walker (1998)].

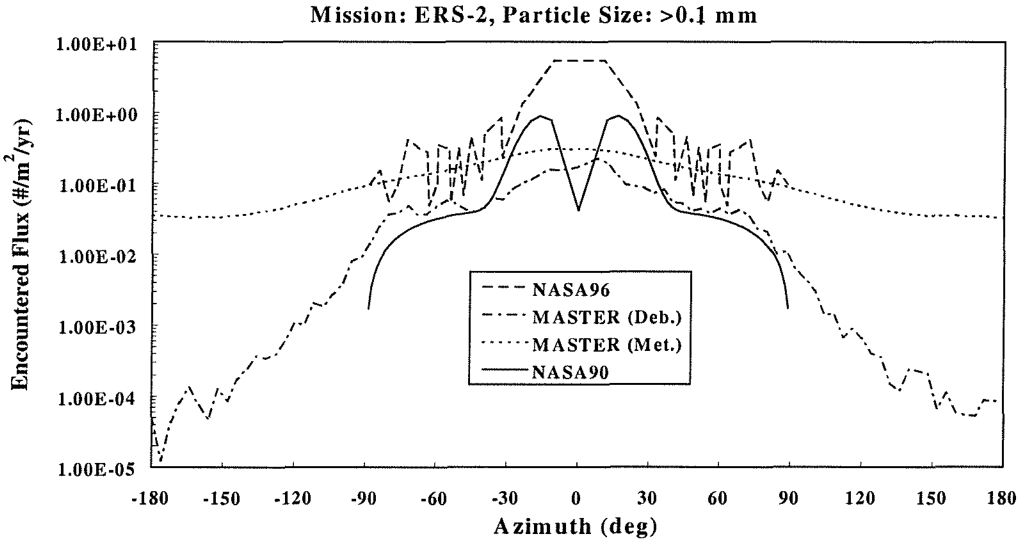


Figure 3-9 Collision flux vs. impact azimuth distributions, ERS-2, objects >0.1 mm [Walker (1998)]

'Probability of failure' is the standard measure of risk. For manned spacecraft, penetration of the structure is the usual failure criterion or 'bottom line risk'. Therefore, it is only necessary to calculate the 'probability of penetration'. However, this failure criterion does not necessarily apply to unmanned spacecraft, which, if carefully designed, should be capable of tolerating a degree of penetrative impact damage. In this instance, the probability of failure is given not only by the probability of penetration, but also the probability of lethal damage to equipment (both internal and external) given a penetration.

Several models have been developed to provide reliable and accurate risk assessments. In general, the models can determine the benefits of different shielding design options, or examine the effects of shadowing (where one part of a spacecraft provides protection to another). Three models that are currently widely used are ESA's ESABASE / DEBRIS and NASA's BUMPER and MSCSurv. These are briefly described below.

ESABASE / DEBRIS

ESABASE / DEBRIS is a popular software tool that is widely available throughout Europe. It has been developed to perform accurate numerical impact analyses and risk assessment predictions of spacecraft [Sdunnus *et al* (1997)]. To fulfill these aims the tool utilises ESABASE's sophisticated orbit propagation and spacecraft modelling capabilities. A user first enters information on the spacecraft geometry (from simple plates to complex structures), material and shielding properties, attitude, mission parameters, etc. Then, for the impact study, the user can select between several state-of-the-art environment flux models. For debris, these include ORDEM96 and MASTER. For sporadic meteoroids, a derivative of the model by Grün *et al* (1985) is used, whereas meteoroid streams are defined using Jenniskens' (1994) approach. Other parameters the user must specify for the impact simulation include the particle size range, damage type, and ballistic limit equations. Damage equations are provided in a generic, parametric form (as set out in Section 3.4), which is flexible enough to allow users to enter their own equations.

During the impact simulation, consideration can also be given to modelling secondary ejecta. This feature is particularly relevant for a highly oblique impact, where the release of large numbers of secondary particles can impact and damage other parts of the spacecraft.

On completion of the simulation, various results can be output. These include the number of impacts for the selected size range, the average impact direction and velocity, and the number of impacts that cause the specified damage. Risk assessments are limited to the calculation of probability of structural penetration, since no internal spacecraft damage model is currently included.

Finally, the program enables 2-D and 3-D graphics visualisation (as shown in the example in Figure 3-10 [Wilkinson & Stokes (1999)]), and allows previewing and plotting of the selected environment and damage models.

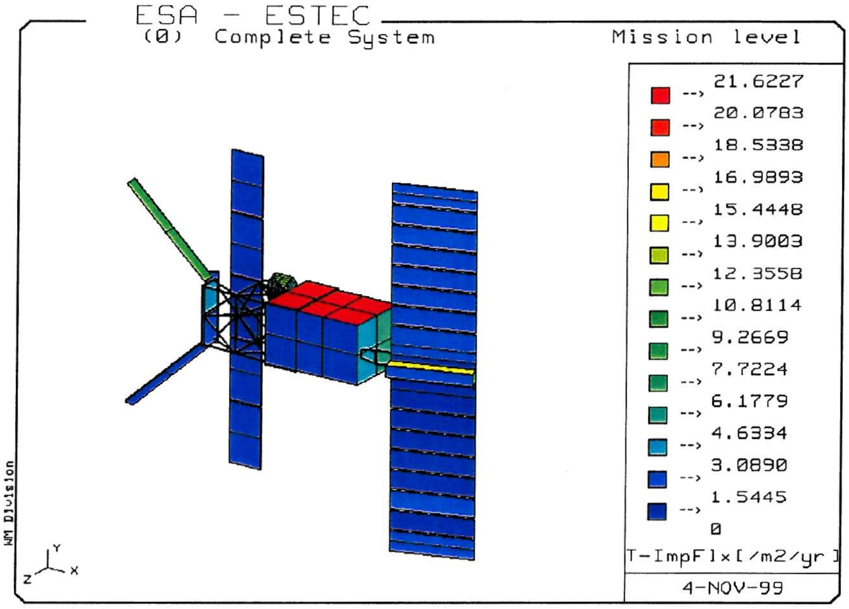


Figure 3-10 Impact fluxes on the ERS-1 satellite resulting from debris larger than 0.1 mm [Wilkinson & Stokes (1999)]

BUMPER

BUMPER is a computer model developed by NASA to assist in the analysis of shielding optimisation, particularly in terms of reducing and distributing shield weight efficiently [Christiansen & Hyde (1994)]. It uses impact damage equations derived from impact tests and analyses, in conjunction with debris and meteoroid environment models, and the geometry of the protection system under consideration. The geometry is defined using a finite element model. BUMPER operates by comparing the impact probability of each particle diameter, impact angle, and velocity in the debris environment to the criteria for whether a shield fails (i.e. the shield's ballistic limit). By summing over all finite elements the total penetration probability of a satellite or module can be estimated [Christiansen *et al* (1992)]. BUMPER has been used extensively to help define the choice and location of various shields on the ISS.

MSCSurv

To assess the likelihood of crew or station loss following an orbital debris penetration of the ISS, the NASA code MSCSurv (Manned Spacecraft Crew Survivability) has also been used [Williamsen *et al* (1999)]. MSCSurv is a Monte Carlo-style program that generates a large number of particles (> 100,000) of varying diameters and velocities from an established debris environment model. The code determines which of these particles penetrate and identifies the resulting failure mode, which can be one of the following: critical cracking, thrust-induced structural failure, critical equipment loss, crew hypoxia, and crew injury. By summing the fraction of penetrators causing damage for each failure mode, the code computes the total probability of loss given a penetration. Mathematically, this is expressed as follows:

$$P_{loss} = 1 - e^{\left(-N_{pen} \times P_{loss/pen}\right)} \qquad [3\text{-}12]$$

Using this tool, engineers have been able to assess the advantages and disadvantages of various candidate ISS designs and operational procedures.

3.6 Guidelines for Implementing Protection

ESA's European Space Debris Safety and Mitigation Standard (2000) presents fundamental safety and mitigation requirements and recommendations related to space debris. As part of this, the standard proposes measures to protect a space vehicle from the debris collision hazard. Although there is no specific protection requirement, the following design recommendations are stated.

Recommendation: Measures should be investigated and applied in order to insure the survivability of space vehicles to debris impacts (for example, shielding) and to decrease the probability of such impacts to occur (for example, avoidance manoeuvres).

The Standard notes that *"some simple rules for placing nominal and redundant devices (for example, equipment, system routing) may reduce the risk of total failure of a space vehicle. These rules are used in reliability/availability analyses."*

Furthermore, the Standard states that *"the high-speed collision between a space vehicle and space debris object involves the transformation of a large amount of kinetic energy into structural damage and release of material. The risk of total failure of the space vehicle is reduced when the recommendations given hereafter are implemented in the space vehicle design, particularly as regards the structure and the configuration."* The design recommendations continue:

Recommendation: Practices related to the connection and positioning of nominal and redundant devices (for example, an equipment and related system routing) should be optimised to maximise survivability from particle impacts.

Recommendation: If the risk due to a collision between a space vehicle and space debris exceeds project criteria, the space project should implement appropriate protection measures (for example, shields, redundancies, location, relative positioning) to reduce this risk.

In terms of operational practices, the following is recommended:

Recommendation: If the risk due to a collision between a space vehicle and space debris exceeds initial project criteria, the space project should implement appropriate additional protection measures (for example, shield augmentation, activation of redundancies) to reduce this risk.

The remainder of this section sets out suggested guidelines to satisfy these recommendations.

3.6.1 Guidelines for Implementing Structural Shielding

The following general set of criteria for the design of structural shielding on a spacecraft (manned or unmanned) have been recommended by Cour-Palais & Crews (1990), and subsequently expanded upon by Turner *et al* (1999). The list is not intended to be exhaustive or prescriptive, but is meant to be a useful guide when considering shield design issues.

- Provide the means for melting and/or vaporising particles over a large range of size and velocity. On manned spacecraft, multi-bumper shielding technologies should be considered. For unmanned spacecraft, double honeycomb panels are a viable option. Additionally, where structures are covered with MLI thermal blanketing, consider offsetting the MLI from the structure and/or enhancing it with materials such as Betacloth, Nextel, and Kevlar for extra resistance.
- Provide extra penetration resistance by using high yield strength materials such as certain aluminium alloys, laminated materials, and composites in spacecraft structures.

- Be affordable, both in terms of material and labour costs.
- Impose minimum weight penalty, i.e. minimise the addition of 'dead weight'. This implies that dedicated shields should be added judiciously.
- Be amenable to simple design and construction, e.g. by using standard manufacturing processes.
- Provide a second 'hit' capability, i.e. minimise the sacrificial aspects.
- Produce non-damaging secondary ejecta and spall. For example, a blanket enhancement material such as Nextel can generate a large number of fibres under impact, which can then cause contamination.
- Any resulting debris, spall or dust from a perforated structure should minimise the risk of subsequent failure or deterioration of spacecraft equipment.
- Avoid adding debris to the orbital environment following an impact, e.g. by trapping it.
- Provide a degree of thermal and radiation protection, e.g. cover exposed structures and shields with thermal control finishes.
- Be resistive to the effects of atomic oxygen (for low Earth orbits only). Again, thermal control finishes could be used to cover exposed surfaces.
- Be capable of surviving the normal launch and in-orbit vibration environments.
- Satisfy system requirements, such as electrical grounding and acceptable thermo-optical properties.
- Avoid interfering with the normal operation of the spacecraft, such as deployment sequences, viewing and communications requirements, etc.
- Minimise costs by considering and implementing shielding as early as possible in the spacecraft design process.

3.6.2 Guidelines for the Design and Placement of Equipment

The survivability of a spacecraft may be greatly improved by relocating vulnerable components and placing sensitive equipment behind existing vehicle structures. It may even be possible for a spacecraft to survive an impact from a centimetre-size object through special consideration of the design (e.g. redundant subsystems, pressure vessel isolation, separation of redundant units, and routing of electrical cables, fluid lines, etc.) [UNCOPUOS Scientific and Technical Subcommittee (1999)]. This is certainly the case for unmanned spacecraft, where the possibility of a degree of penetrative impact damage might be tolerable.

An integrated approach to protection is therefore needed. That is, a balance must be struck between the level of physical (e.g. structural) shielding and the arrangement of equipment, if cost-effective protection is to be achieved. Prior to this thesis, the techniques for implementing such an integrated approach had not been developed. Instead, the following general set of guidelines has been recommended for the design and placement of equipment and instrumentation on a spacecraft [Turner *et al* (1999)]:

- Identify the areas of the satellite most vulnerable to debris impact. For most unmanned spacecraft in LEO circular, polar orbits, these are surfaces facing the velocity vector direction.
- Identify mission-critical and sensitive equipment by performing a Failure Modes Effects and Criticality Analysis (FMECA). Consideration of items such as batteries, propulsion tanks / pipes, reaction / momentum wheels and gyros is especially important.
- For internal equipment, move sensitive and critical units away from vulnerable surfaces (e.g. those facing the velocity vector) and/or place them behind (relative to the vulnerable face) less critical units or internal structure. Any reconfigurations do of course need to take account of various system constraints, such as mass and thermal balance. For this reason, such an approach is best considered during the earliest design phases when there is greater flexibility. Other factors requiring consideration include:
    - Units of the same subsystem are usually located close to one another to minimise wiring. Relocating a unit may therefore result in extended cables, which would increase vulnerability to debris and add mass.
    - There is a possibility that a single impact could cause the loss of both a primary and redundant unit if they are collocated. This suggests that some physical separation may be advisable.
    - Units should not be relocated to positions near the edge of structure panels. This is to avoid strength / stress problems at the panel edges, which may already be heavily loaded.
    - When relocating certain units such as magnetorquers, electromagnetic interference effects must be avoided, e.g. by allowing sufficient space from other units.
- If possible, avoid mounting any internal equipment directly onto vulnerable surfaces, as this avoids the need for placing critical associated items, such as radiators, harness and coaxial cables, on or near those faces.

- ❑ For vulnerable items (especially externally located ones), consider covering them with MLI, enhanced MLI, or layers of high-strength fabric (subject to thermal considerations). If the item has a protective casing, consider the possibility of thickening it.
- ❑ Protect sensitive external equipment by:
  - 'Shadowing', i.e. locating it behind (relative to the debris flux) other external equipment that may be less sensitive or mission-critical.
  - Placing it on trailing or Earth-facing surfaces, again to minimise the risk of debris impact.
- ❑ If possible, avoid mounting critical external equipment on vulnerable surfaces, and prevent external instruments such as attitude sensors viewing in the directions of greatest debris flux.

3.7 Discussion

Implicit in the preceding sections is the notion that designers should attempt to implement optimised debris protection in a spacecraft. For clarity it is worth highlighting how this is currently done in practice.

Generally, a shield is designed for a specific mass / velocity range of particles. Outside this range, i.e. at too low or too high a thickness, the shield will not function properly and may actually enhance the damage risk. For example, when the shield is sufficiently thick or an incident particle is moving fast enough, the particle will be completely disrupted – typically melting or vaporising. However, if the shield is too thin, the particle may penetrate the shield largely intact with the mass of the extra shield material added (especially if the particle has a low velocity). On the other hand, if the shield is too thick, the shield material may spall off in large solid fragments thus enhancing the damage risk compared to the original particle. Therefore, there is an optimum region in which the combination of shield thickness and particle mass and velocity minimises the damage potential of any penetrating material.

Christiansen (1992) describes a method for making a preliminary estimate of the size of shield necessary to defeat a given particle threat. This is known as the 'design particle method'. A "design" particle size is calculated for each spacecraft surface based on 'probability of no penetration' requirements, and the directionality of collision fluxes relative to the spacecraft surfaces. The thickness of the shield bumper(s) and back-up wall are then sized to stop the "design" particle, which is assumed to have an average velocity, density, and impact angle.

Although the design particle method is useful for performing quick trade-off studies, it is not suitable for verifying design adequacy or for detailed examination of design options because of the simplifying assumptions made. Fortson (1992) attempted to go a step further than the design particle method by deriving a semi-empirical model that relates shield thickness to the probability of a penetration. This is achieved by performing a Monte Carlo simulation, which samples probability density functions of particle size and velocity (from environment models) to create test particles that are 'fired' at a given shield design. The shield's ballistic limit equation determines which of the test particles penetrate. Probability of penetration is then simply calculated by totalling the number of times the shield was penetrated versus the total number of impacts. The whole process is repeated for different shield thicknesses, until a graph of shield thickness versus probability of penetration can be produced. For a given probability of penetration requirement, it is then simply a matter of reading off the optimum shield design from the graph.

The advent of sophisticated software models, such as BUMPER and ESABASE / DEBRIS, has since brought about the means to perform this type of optimisation with reasonable efficiency. However, the basis of such an optimisation is the assumption that a spacecraft will fail catastrophically when a projectile passes through a shield. As noted previously, this assumption may be satisfactory for manned spacecraft, but it is not necessarily true of unmanned spacecraft. For example, Fortson (1992) notes that

"a projectile, having penetrated the shield, may have too little energy left to render the satellite inoperable. Or, it may simply fail to make contact with a vital region of the satellite. However, a thorough consideration of these factors requires detailed knowledge of the satellite under consideration, and is beyond the scope of a simple semi-empirical model."

Certainly, there is no evidence to suggest penetration automatically equals failure. Consequently, for unmanned spacecraft, the above optimisation approach is too conservative and will recommend unnecessarily massive shields. If it is accepted that there is a quantifiable probability that a satellite could survive a penetrative impact, then optimisation must be considered in the context of the probability of satellite failure, not shield penetration. By default, consideration of the arrangement and design of equipment behind a shield becomes an integral part of the optimisation problem. For example, a critical internal unit may be afforded a degree of protection from a more robust neighbouring item. The need for additional external shielding mass is therefore mitigated.

This approach immediately opens up the possibility of many new competing protection solutions to the spacecraft engineer. For example, numerous options for the design and placement of internal equipment and shielding might be considered. The difficulty of course is to determine which is the best. A technique is therefore needed that can not only distinguish between 'good' and 'bad' solutions, but also focus quickly on a selection of promising candidates. As noted by Fortson (1992), the construction of such an optimisation model is not straightforward. But this should not necessarily deter efforts.

4 A New Approach for the Implementation of Protection

4.1 Introduction

One of the key findings to emerge from the preceding chapter is the absence of a methodology or model for performing a comprehensive spacecraft protection optimisation. Current approaches tend to focus on increasing the probability of no penetration (PNP) of a spacecraft by assessing and refining its structural shielding in an iterative manner. Less attention is paid to the subtler and potentially cost-beneficial approach of improving protection by considering the vulnerability and arrangement of the internal spacecraft equipment. This may in part be because there are no widely available tools to perform this type of assessment.

Recently though, attempts have been made to investigate the advantages of relocating critical components inside an actual satellite design, specifically METOP [Turner *et al* (1999)]. METOP is a meteorological satellite that is due to be launched into an ~800 km circular, near polar orbit for a period of four years from 2005. A very simple metric was devised to assess whether moving certain items could enhance the survivability of the Service Module on the METOP satellite. Figure 4-1 shows the layout of the module's internal components, which are defined in Table 4-1. A more detailed discussion of METOP is given in Chapter 6.

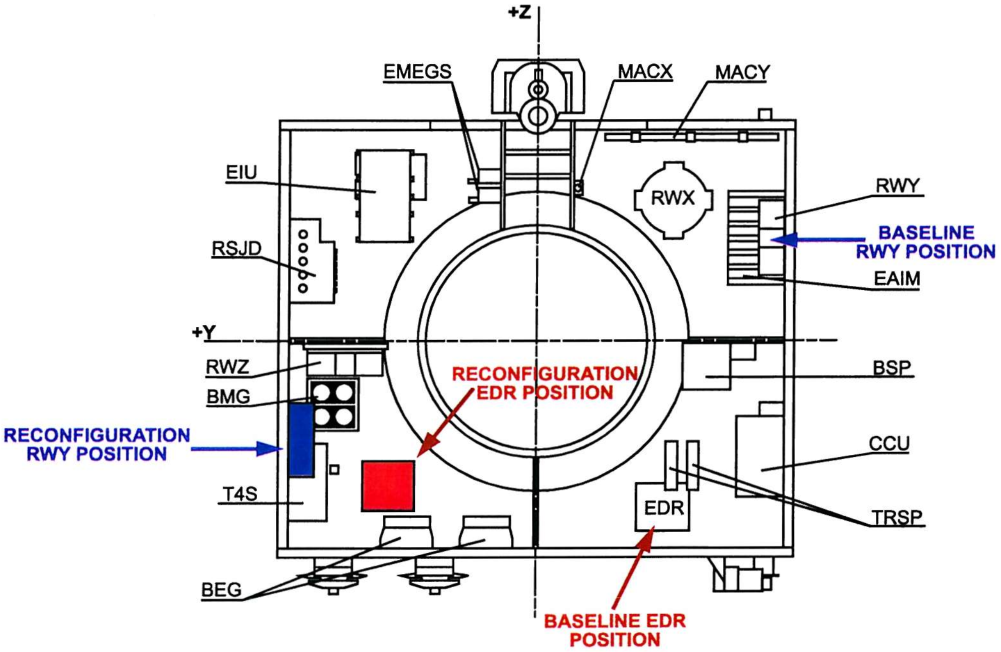


Figure 4-1 Relocation of components in the METOP Service Module [Turner et al (1999)]

Component acronym	Component name	Function
CCU	Central Command Unit	Controls the Attitude and Orbit Control Subsystem (AOCS) pointing by processing the Attitude Sensor data and creating the commands to the AOCS actuators. Controls the heating of the AOCS and batteries, and also the state of recharge of the batteries.
BSP	Boitier de Servitude Pyrotechnique	Provides control and monitoring signals to subsystems/ units without their own access to the On-Board Data Handling (OBDH) unit. The major services are supplied for heaters (duration of mission) and also pyrotechnic release controls (beginning of the mission only).
OBA	On-board Adaptor	On-board Adapter (CCU - EGSE).
EDR	Electronique de Decodage et de Reconfiguration	Receives AOCS alarm and OBDH interrupt (IT) signals; generates the Safe Mode command to the Power and AOCS for initiation of entry to Safe Mode control.
TRSP	Transponder	Each transponder includes a Diplexer, Receiver, Transmitter and a DC/DC Converter.
EAIM	Electronique de controle des Actuateurs Inertiels et Magnetiques	Provides the drive signals to the Reaction Wheels and the Magnetorquers.
RW Y	Reaction Wheel: Y Axis	Provides a torque around the Y-axis.
RW X	Reaction Wheel: X Axis	Provides a torque around the X-axis.
MAC Y	Magnetorquer	Reacts against the Earth's magnetic field to provide the required pointing of the satellite. Works in combination with the Reaction Wheels.
MAC X	Magnetorquer	Reacts against the Earth's magnetic field to provide the required pointing of the satellite. Works in combination with the Reaction Wheels.
EPRM	Electronique de controle de Propulsion et MEGS	Provides the configuration, drive demand signals and monitoring acquisition signals for the MEGS. Also provides the drive stages and transducer monitoring conditioning for the electromagnetically operated valves and sensors of the Propulsion subsystem.
EIU	Electronic Interface Unit	Provides the switching of the drive signals to the items of the Solar Array in order to achieve the correct Solar Array deployment.
RSJD	Regulateur Shunt Jonction et Distribution	Controls the main bus voltage and battery operation (e.g. state of charge, temperature control). Provides for the switch-off of nominal power lines and switch-on of the Safe Mode functions power supplies at the occurrence of a Safe Mode Order from the EDR.
MEGS-E	Mecanisme Entrainement du Generateur Solaire - Electronique	This is the electrical drive unit, which interfaces directly to the Solar Array motor and position transducer. It receives position correction demands from the EPRM.
RW Z	Reaction Wheel: Z Axis	Provides a torque around the Z-axis.
BMG	Boitier Mecanique Gyro	This unit consists of 4 gyroscopes mounted on a single reference plate. Each gyro senses 2 axes of information.
T4S	Terminal Senseur Survie et Surveillance SCAO	Provides the interface between the Attitude Sensors and the OBDH. Contains the Safe Mode software for the control of the attitude.
BEG	Boitier Electronique Gyro	This unit consists of 4 electronics boxes which are associated with the gyros.

Table 4-1 Definition of components in the METOP Service Module [Turner et al (1999)]

Although far from conclusive, the results were encouraging. Potentially significant improvements could be realised simply by repositioning one of the critical units, the RWY, away from the velocity vector face, which had the highest debris impact risk. Another critical component, the EDR, was also relocated, but this time to a different part of the vulnerable face to which it was attached. In this instance, the survivability enhancement was found to be much less significant.

The results served to underline the complexity of the problem of finding a cost-effective spacecraft protection solution, since it became apparent that this could only be achieved by considering both the shielding and equipment configuration options simultaneously. However, this was not possible for the METOP study because it would have required a very time-consuming trial-and-error assessment approach. Any changes to the design, however minor, would have had to be implemented manually. Therefore, rapid convergence on a cost-effective solution would not have been likely.

The challenge, therefore, is to overcome these limitations through the development of a new model or methodology. This must be able to compare radically different spacecraft protection solutions, identify a cost-effective choice of shielding and internal configuration, and do so in an automated fashion. Since there are many permutations for the choice and location of shielding and internal arrangement of components, such a model would need to be capable of searching through the numerous competing designs in a timely and efficient manner. These requirements suggest the need for a sophisticated search technique as a core function in a new model. Several possibilities have been identified in the literature, and these are discussed below.

4.2 Review of Modern Stochastic Search Techniques

The need for engineers to establish critical design parameters and choose among alternative concepts during the early stages of the design process, necessitates the use of a formal optimisation approach to fully explore the design [Ashley (1982)]. The standard methodology in aerospace design is to generate a parametric description of a concept and, using engineering judgement, estimate initial values for the parameters [Gage (1995)]. After defining an objective (or merit) function and minimum performance requirements, the proposed design is evaluated. Systematic changes to parameter values eventually lead to improvements in the design. A number of formal techniques now exist to assist engineers in the process of optimising a multi-variable system. These can be classified as traditional optimisation techniques and modern stochastic techniques.

The domain of engineering contains many problems that require the optimisation of a function that is dependent on many parameters [Krishnakumar *et al* (1995)]. Such problems pose a challenge because of the large parametric space from which to choose a solution, the possibility of infeasible areas within the space, and the presence of multiple local maxima. Traditionally, one had to rely on techniques such as calculus of variations, enumeration search methods or random search methods to solve these problems. Such methods have been studied for many years, both as a mathematical problem and in engineering design, and a sizeable literature is available [Keane (1994)]. However, their application tends to be limited to simple, low-order optimisation problems. This is usually for one of two reasons (depending on the search technique):

1. Inefficiency in searching the design space. Many evaluations of the objective function may be required, and important regions of the design search space may be excluded.
2. Difficulties in selecting a correct starting point. In higher dimension problems it is possible for a number of local optima to exist, each of which might satisfy the constraints of the problem. Successful convergence is highly dependent on the initial guess of the parameter values. If the chosen starting point is close to a local optimum, the algorithm will converge to it rather than the desired global optimum.

Given the above limitations of traditional methods for tackling complex design optimisation problems, it is not surprising that attention has focused on alternative methods in recent years [Ewing & Downs (1996)]. The principle aim of these new methods is to minimise the number of evaluations of an objective function that are needed to converge rapidly on a satisfactory solution [Hicks & Porter (1995)]. A search procedure is regarded as efficient if the number of evaluations required is small compared to the total that might be possible within a given parametric space. The robustness of a search procedure is a measure of how the efficiency changes when the parameters of the problem are modified. Therefore, a procedure that is much more efficient at solving one particular problem over another is not robust. In this section, modern stochastic methods will be examined to see if they might be efficient and robust enough for the protection optimisation problem.

Keane (1993) states that *"modern stochastic methods generate sequences of random points but use some overall control strategy to link the acceptance of new points to the progress of the search. Since they produce new points in this way they do not use information about the local slope of the objective function and are thus not prone to stalling in false optima. They are all essentially unconstrained methods and so make use of penalty functions to deal with constraints."*

Three of the most popular stochastic search methods have their basis in natural processes: simulated annealing, genetic algorithms, and evolutionary programming. They have all been considered for a range of optimisation tasks in engineering and, in particular, aerospace design [Ewing & Downs (1996)]. Each of these techniques will now be considered in more detail.

4.2.1 Simulated Annealing

Simulated annealing is based on the kinetics of freezing crystals, where minimum energy states are achieved through sufficiently slow cooling [Keane (1994)]. In other words, the technique aims to minimise the objective function. The method generates random points in the region of the current best point and evaluates their objective functions [Huang & Arora (1995)]. If a trial point is infeasible, i.e. it lies outside certain practical design constraints, then it is immediately rejected. On the other hand, if a trial point is feasible and the objective function is smaller than the current best value, then the point is accepted and the value of the best objective function updated. Finally, if a point is feasible and its objective function value is higher than the current best value, then the point is sometimes accepted and sometimes rejected. This 'acceptance probability' is determined using the Boltzman probability density function (PDF):

$$p(\Delta U) = \exp(-\Delta U/kT) \qquad [4\text{-}1]$$

When the PDF value is greater than a uniformly probable random number between 0 and 1, then the trial point is accepted as the best solution even though its objective function value is higher than the known best value. The temperature T in the PDF is the target value for the objective function. Initially, a large target value is selected, which reduces as the trials progress. ΔU represents the change in objective function value U, and k is the absolute value of the first, non-penalised change in the objective function.

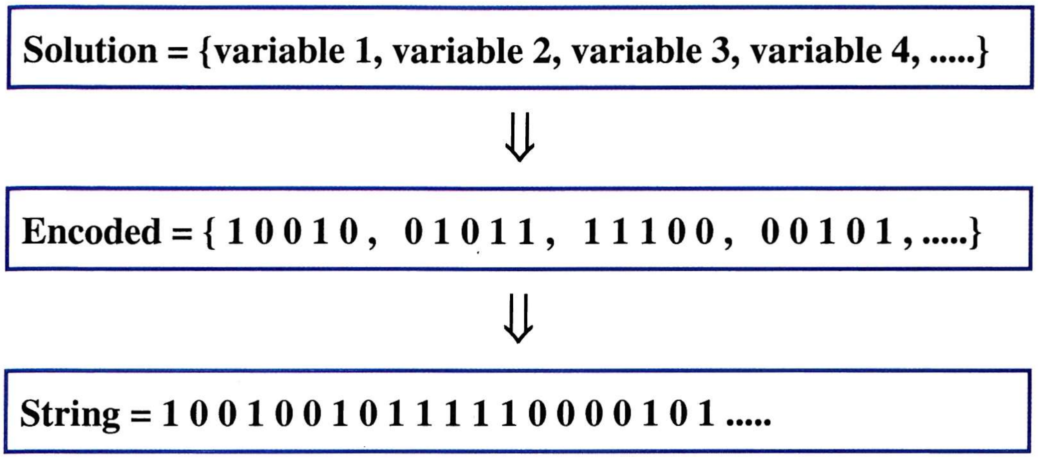
The acceptance probability steadily decreases to zero as the temperature is reduced. Therefore, it is probable that poor designs will be accepted in the initial stages, but much less so in the final stages. Such a strategy avoids the problem of becoming trapped on a local optimum. However, the method does have drawbacks, for example the unknown rate for reducing the target value T, uncertainty in the total number of trials, and uncertainty in the number of trials after which the target value needs to be reduced.

Keane (1993, 1994, & 1996) notes that, although simulated annealing is a credible alternative to genetic algorithms, its performance seems to be the worst of the three stochastic techniques. Furthermore, its sensitivity to control parameters suggests that it is not particularly robust. Inefficiencies in the application of simulated annealing have also been reported by other researchers [e.g. Hicks & Porter (1995)].

4.2.2 Genetic Algorithms

Genetic algorithms were invented in the 1970's to imitate some of the processes observed in biological evolutionary theory [Davis (1996)]. When incorporated into a computer program, genetic algorithms yield a technique for solving a difficult optimisation problem in the way that nature has done - through evolution. The principle of 'survival of the fittest' best describes the operation of genetic algorithms. Essentially, a genetic algorithm is an automated search technique for evolving better solutions to an optimisation problem.

The genetic algorithm process begins by randomly generating a population of competing solutions to a problem. Each solution is evaluated for its 'fitness', i.e. a measure of its performance in the context of the problem. This is analogous to the interaction of a biological organism with its environment. The solutions are then converted into strings of encoded information (usually binary strings) - the mathematical equivalent of biological chromosomes. Figure 4-2 provides an example to show how the design variables of a particular solution are encoded and combined into a long binary string.



Solution = {variable 1, variable 2, variable 3, variable 4, ...}

⇓

Encoded = { 10010, 01011, 11100, 00101, ...}

⇓

String = 1 0 0 1 0 0 1 0 1 1 1 1 1 0 0 0 0 1 0 1 .....

Figure 4-2 Example of a binary string or 'chromosome'

With the creation of a population of chromosomes, the algorithm can perform the three fundamental operations of evolution, namely: selection, crossover, and mutation. The selection operator is designed to choose two 'parent' chromosomes at random from the population, in such a way that the fitter chromosomes are more likely to be selected. The crossover operator 'mates' the parent chromosomes by swapping information (i.e. segments of the binary strings) to create two new 'child' chromosomes. Finally, the child chromosomes are mutated slightly to diversify the population. These three operations are repeated until a completely new generation of chromosomes has been created. After a number of generations, the fittest chromosome in the final population represents the best solution.

Essentially, genetic algorithms judiciously use the idea of randomness when performing a search. However, it must be clearly understood that they are not simply random search algorithms. Random search algorithms can be inherently inefficient due to the directionless nature of their search. Genetic algorithms are not directionless. They utilise knowledge from previous generations of strings in order to construct a new generation that will approach the optimal solution. In other words, they use past knowledge to direct the search.

So, genetic algorithms can provide robustness, efficiency and flexibility when searching a problem space for the optimum solution. To understand why, it is worth digressing here to briefly discuss some of the underlying theory. First, the concept of 'building blocks' needs to be introduced. The idea behind these is that very fit individuals in a population pass on high performance building blocks to their children. These building blocks take the form of sub-strings. Therefore, a binary string with a high fitness value must contain a sub-string that is a primary cause of the fitness. Thus, even though the crossover operation may splice the string into two, there may be a good chance that the highly fit sub-string is passed on to the children.

Next, the concept of 'schemata' needs to be understood. Schemata are templates of strings that describe similarity between certain sets of strings. In order to define a schema, the alphabet in Table 4-2 is used.

Binary Alphabet	Meaning
0	Binary 0 in String
1	Binary 1 in String
*	'Don't Care' Term

Table 4-2 Definition of a schema [Davis (1996)]

The alphabet, {0, 1, *}, can be used to represent any pattern of binary sub-strings, as shown in the example in Table 4-3.

Schema	Matching Strings
*1111	01111
	11111
*010*0	101010
	001010
	101000
	001000
**1*0	00100
	....

Table 4-3 Schema representation of a sub-string [Davis (1996)]

One or more of the schema may produce high performance sub-strings. Therefore, much of the power of a genetic algorithm revolves around its ability to process these building blocks in such a way as to create increasingly fitter strings. For example, in the selection process, fitter strings will be selected more often than weaker strings, and so fitter schemata have a greater chance of being involved in the creation of the next generation than their weaker counterparts. That is, they will occur more frequently in the next generation [Davis (1996)].

From the above discussion, it becomes apparent that the performance of a genetic algorithm can be altered by varying several of its operating parameters. For example, the size of a population is normally varied between 30 and 200 chromosomes (although a recent form of genetic algorithm called a micro genetic algorithm [Coverstone-Carroll (1996)] typically uses populations of less than 30). The probabilities of crossover and mutation can also be adjusted, as well as the number of crossover segments in a chromosome. Careful choice of these is important, otherwise they may disrupt the schemata too much, and therefore reduce efficiency. Modifying these types of parameters, to affect how the algorithm explores the search space, is known as tuning. Numerical studies [De Jong (1975)] have been carried out to optimise these parameters for maximum convergence rate.

The strength of genetic algorithms is their ability to sort through many competing design solutions, and focus on the region near the global optimum [Pinon & Fowler (1995a)]. The main drawback is that once near this region they may have difficulty focusing on the exact global optimum. However, in many optimisation problems, especially multi-dimensional problems such as protection optimisation, it is sufficient to narrow the search space and derive a near-optimum solution. That said, there has been much research to understand the behaviour of genetic algorithms, and identify the criteria for convergence to the global optimum. Numerous genetic algorithm variants have also been developed over the last decade to improve their convergence and robustness. These include:

- Different methods for crossover and mutation [Furuya & Haftka (1995)].
- Introducing an evolutionary direction operator [Yamamoto & Osamu (1995)].
- Replacing the normal fitness function with a characteristic function based on fuzzy logic [KrishnaKumar (1993)].
- Incorporating advanced operators (in addition to the usual reproduction ones), for example to protect successful solutions from extinction [Ewing & Downs (1996)].
- Implementing methods for reducing the length of chromosome strings (thereby reducing the size of a population) [Krishnakumar *et al* (1995)].
- Strategies for handling constraints in the optimisation problem [Ewing & Downs (1996), Hajela & Yoo (1995), and Seywald & Kumar (1995)].

Another approach for achieving robustness is to hybridise a genetic algorithm [Davis (1996)]. That is, to incorporate the best elements of rival optimisation techniques, or knowledge of the design space, into the genetic algorithm itself. All of these techniques have been applied with some degree of success, thus ensuring that genetic algorithms continue to be applied across a wide range of problems.

One should remember that, with genetic algorithms, designers and engineers are not doing the optimisation. They are creating conditions in which optimisation occurs, as it may have occurred in the natural world. This is what makes them so attractive for complex applications. Some of the more interesting of these include parametric design of aircraft, robot trajectory generation, multiple fault diagnosis in a communication system, schedule optimisation, composite material design, and structural design [Keane (1995)]. Clearly, engineering, by its very nature, has many potential applications for genetic algorithms. In the discipline of space engineering, genetic algorithms have already been used for trajectory optimisation [Pinon & Fowler (1995a&b), Coverstone-Carroll (1996), Carter & Vadali (1995), and Noton (1995)] and many more applications are waiting to be explored. The technique holds promise for a variety of spacecraft optimisation problems, including the design of cost-effective protection. It is not difficult to see that there is an analogy between the way nature evolves a biological organism to survive a high threat environment, and the way a genetic algorithm could evolve a satellite design to withstand the debris and meteoroid environment. Intuitively one senses that a genetic algorithm ought to be a suitable method for the protection search problem.

4.2.3 Evolutionary Programming

Keane (1993) describes a typical evolutionary programming algorithm. In this, evolution occurs by creating a mutated child from each parent in the population. The amount of mutation is related to the objective function, thereby ensuring that 'good' solutions are mutated less.

This entire approach is quite different to the genetic algorithm process where some parents may never produce children and others may produce several. Also, in the genetic algorithm approach, the child designs are derived by mixing the genetic code from a pair of parents, whereas in evolutionary programming each child comes from the genetic code of a single parent. Finally, in a genetic algorithm the amount of mutation is not connected to the objective function.

After the evolutionary program forms the objective function of each child, a stochastic process is used to rank and select the next generation of parents from the combined set of parents and children, whilst still retaining the best solution. Essentially, this means reducing the oversized population of parents and children back to its original size through a competitive process known as tournament selection. This is a stochastic process where each member 'competes' against other members selected at random. Those with the higher objective functions are more likely to win and be retained, while those that lose are discarded.

Once again, it is clear that this method of creating a new population differs considerably from the genetic algorithm method. With the exception of the 'best' parent, the genetic algorithm approach does not allow for the possibility of parents going through to the next generation of the population.

Evolutionary programming forms a credible alternative to genetic algorithms and, according to Keane (1993, 1996), its performance on a variety of search problems lies somewhere between simulated annealing and genetic algorithms. However, the sensitivity of evolutionary programming to control parameters suggests that it may not be as robust as genetic algorithms.

4.3 Prototype Model

From the preceding discussions, one can surmise that the genetic algorithm search technique is potentially the most promising. The incorporation of a genetic algorithm into a comprehensive protection optimisation model would be both innovative and challenging. However, before embarking on such an undertaking, it would seem prudent to check the suitability and efficacy of applying a genetic algorithm to this type of problem. Consequently, the author decided to develop a prototype demonstrator model [Stokes *et al* (1997)].

By necessity, the principal design requirements of a prototype are limited in scope. At the very least, the prototype model must demonstrate the ability to search efficiently through many satellite protection solutions, and converge on or near the global optimum. For this, protection options can be restricted to structural shielding on various satellite surfaces, and / or different arrangements of equipment inside the satellite. If the prototype can show that such a concurrent engineering approach is feasible, then it is reasonable to pursue a full development version of the model.

4.3.1 Overview of the Architecture Design

The prototype model was written in C programming language, and comprises some 7,000 lines of source code. The top-level architecture of the model is shown in Figure 4-3. The model starts by requiring two sets of input data from the user:

1. Simple 2-D box-shaped representations of the satellite body and its internal components. These are defined in terms of dimensions, mass, and cost.
2. A randomly generated distribution of particles that impact the external surfaces of the body geometry during its mission life. The particles are defined in terms of their size, and impact speed and angle relative to the satellite surfaces.

Once this data is entered, the simulation begins by randomly creating an initial population of competing spacecraft designs. Each solution has a unique arrangement of internal components together with a unique combination of shielding protecting the outside of the satellite body.

The positioning of components inside a given design is constrained to one plane within the satellite. That is the z, coordinate is fixed, and the x, y coordinates of the component attachment point (i.e. its centre) must be such that the whole of the component lies within the bounding body coordinates. Additionally, no two components are allowed to occupy the same space. This problem is overcome using a 'clash detection' algorithm, which is described later in this chapter (cf. Section 4.3.4).

Up to four types of shield can be selected for each of the randomly generated designs. These are single wall (i.e. the existing body structure panels), Whipple, mesh double bumper, and multi-shock shield. The values of various shield parameters, such as thickness, density, and standoff distance, are input prior to the simulation.

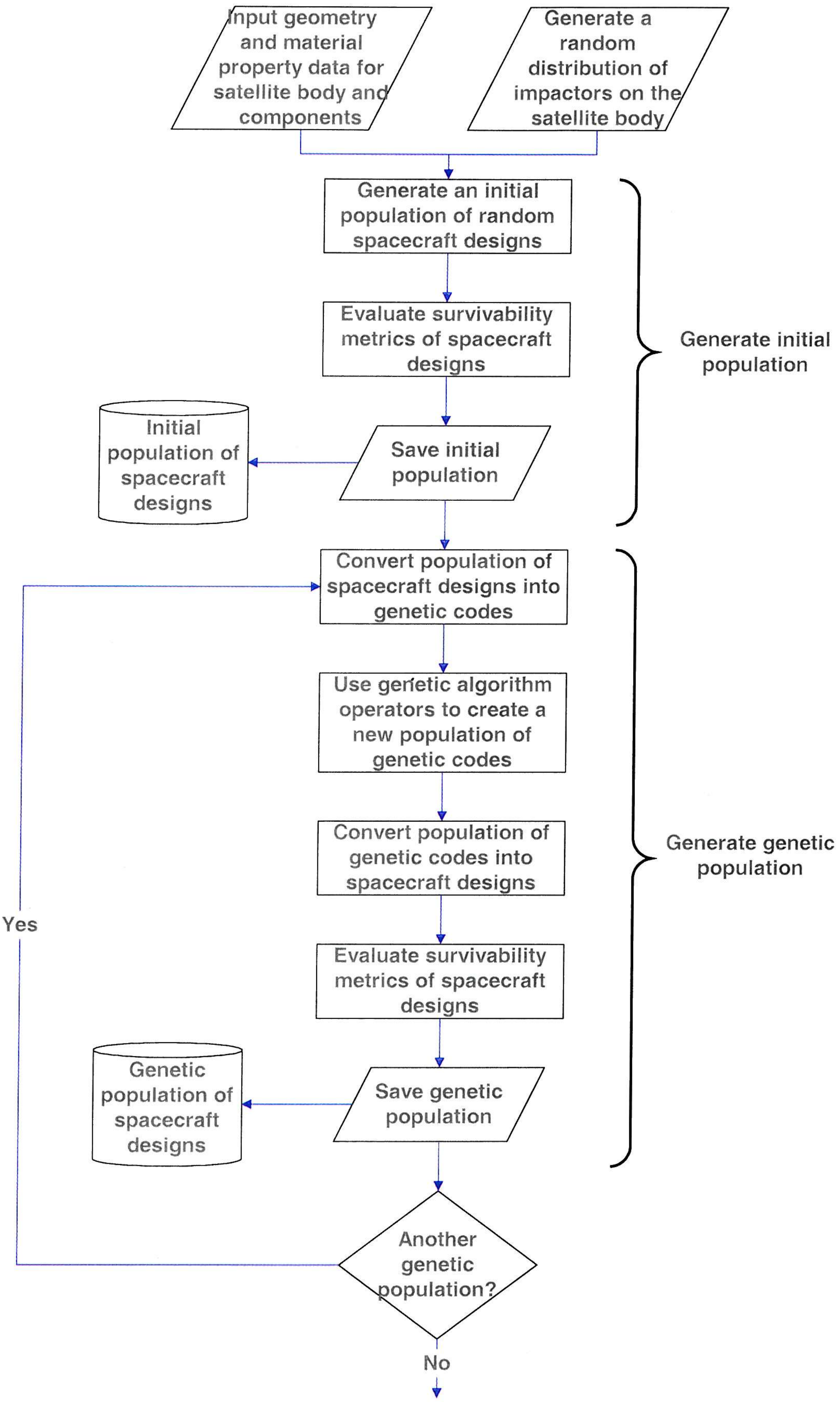

Input geometry and material property data for satellite body and components
Generate a random distribution of impactors on the satellite body
Generate an initial population of random spacecraft designs
Evaluate survivability metrics of spacecraft designs
Save initial population
Initial population of spacecraft designs
Generate initial population
Convert population of spacecraft designs into genetic codes
Use genetic algorithm operators to create a new population of genetic codes
Convert population of genetic codes into spacecraft designs
Evaluate survivability metrics of spacecraft designs
Save genetic population
Genetic population of spacecraft designs
Generate genetic population
Yes
Another genetic population?
No

Figure 4-3 High level architecture design of the prototype model

Each of the competing solutions is then evaluated to determine their survivability, or 'fitness' to survive the environment. From Chapter 3 one can see that while methods exist for determining the efficacy of different choices of shielding, there is very little in the literature for evaluating and comparing different configurations of internal components. Therefore, it was necessary to formulate a new technique.

4.3.2 A Simple Method for Evaluating Satellite Survivability

One possible approach for evaluating the survivability is to model the effects of multiple penetrating impactors on a satellite design and determine the average internal damage. A ray-trace technique could be used to determine the interactions between the clouds of penetrating fragments and internal components. Empirically derived damage equations would then calculate the resulting impact damage, and therefore the consequences to the mission. From this, the probability of failure of the spacecraft can be determined.

This is a fairly rigorous approach for assessing the survivability of a particular design. However, for the purposes of the prototype demonstrator it is perhaps too computationally intensive. The focus of the prototype is to demonstrate the concept of an automated protection optimisation process that allows simultaneous consideration of external shielding and internal satellite configuration. Therefore, a simpler evaluation approach can be adopted, which is still sufficiently representative that it will act as a guide in distinguishing between competing protection solutions. The approach chosen was to define a 'survivability metric' that could encapsulate some of the features of the more rigorous method mentioned above. It is conceivable that there may be several ways to define such a metric, and undoubtedly one could expend considerable effort exploring the alternatives. However, this would be an unnecessary digression from the prototype model development. The following metric was considered to be suitable. It is derived in three distinct steps.

Survivability Metric Derivation - Step 1

Using the appropriate ballistic limit equations, as listed in Section 3.4, a calculation is performed to identify those particles from the impact distribution that penetrate the satellite body (and any shielding). Each of the penetrators generates a cloud of secondary debris inside the body. A cloud typically comprises material from both the impactor and target surface/shield. Impactor fragments generally follow a trajectory close to the line of sight of the incident particle, whereas target fragments tend to follow a trajectory that is closer to the surface normal. The trajectory angle and spray angle of a secondary debris cloud is calculated using an empirical equation that has the following generalised form:

$$\psi = K\left(\frac{v_p}{c_s}\right)^{\eta}\left(\frac{t_s}{d_p}\right)^{\lambda} f(\theta) \qquad [4\text{-}2]$$

The various terms in this equation are defined in Table 4-4 and illustrated in Figure 4-4.

Symbol	Unit	Description
K	--	Characteristic factor
v_p	kms^{-1}	Particle impact velocity
c_s	kms^{-1}	Speed of sound in shield material
t_s	cm	Thickness of shield
d_p	cm	Particle (impactor) diameter
θ		Impact angle (with respect to surface normal)
θ_1		Trajectory angle of 'normal' spray (with respect to surface normal)
θ_2		Trajectory angle of 'in-line' spray (with respect to surface normal)
ϕ_1		Spray angle of 'normal' spray (with respect to surface normal)
ϕ_2		Spray angle of 'in-line' spray (with respect to surface normal)
Ψ		= (θ_1 / θ) or (θ_2 / θ) for trajectory angles (spherical impactor) = tan θ_1 or tan θ_2 for trajectory angles (cylindrical impactor) = tan ϕ_1 or tan ϕ_2 for spray angles

Table 4-4 Trajectory angle and spray angle equation terms

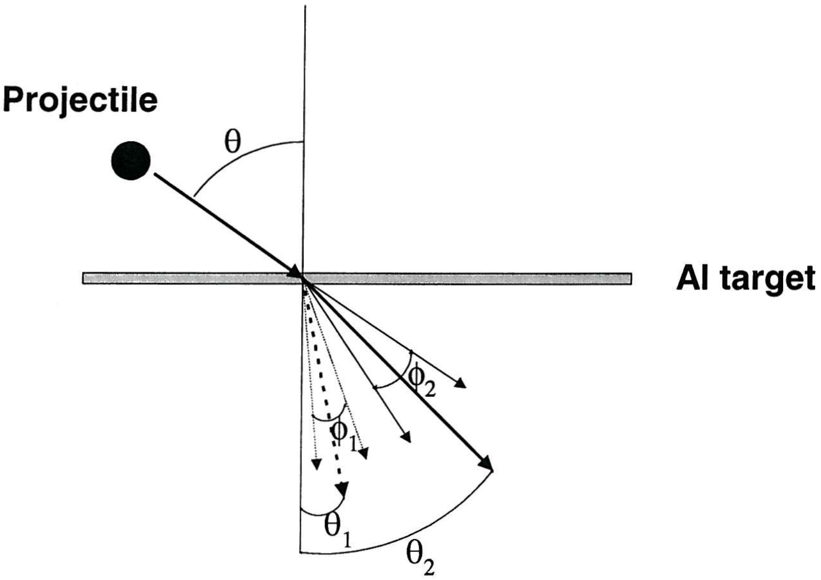


Figure 4-4 Trajectory angles and spray angles of a secondary debris cloud

Typical values for the parameters and functional forms of *f(θ)* in Equation 4-2 are listed in Table 4-5.

Equation	Validity	Ψ	K	η	λ	$f(\theta)$
Trajectory						
Aluminium sphere particle	$30° < \theta < 75°$	(θ_1 / θ)	0.471	-0.049	-0.054	$\cos^{1.134}\theta$
on Aluminium plate	$2 < v_p < 8$ kms^{-1}					
[Schonberg *et al* (1991)]	$0.064 < (t_s/d_p) < 0.684$	(θ_2 / θ)	0.532	-0.086	-0.478	$\cos^{0.586}\theta$
Cylindrical particle on	$0° < \theta < 65°$	$\tan \theta_1$	0.2216x10^{-8}	-1.710	-11.557	$\cos^{3.318}\theta$
Aluminium plate	$2.95 < v_p < 6.90$ kms^{-1}					
[Schonberg *et al* (1991)]	$0.152 < (t_s/d_p) < 0.315$	$\tan \theta_2$	0.2536x10^{-7}	-2.570	-9.952	$\cos^{1.088}\theta$
	cylinder $(L/d_p) = 1$					
Spray						
Aluminium sphere particle	$0° < \theta < 75°$	$\tan \phi_1$	1.318	0.907	0.195	$\cos^{0.394}\theta$
on Aluminium plate	$2 < v_p < 8$ kms^{-1}					
[Schonberg *et al* (1991)]	$0.064 < (t_s/d_p) < 0.684$	$\tan \phi_2$	1.556	1.096	0.345	$\cos^{0.738}\theta$

Table 4-5 Typical parameter values for the trajectory angle and spray angle equation

To a first approximation, an expanding debris cloud creates a high-risk volume inside the spacecraft, which is characterised by 'vulnerability' cones with particular trajectory and spray angles. Any spacecraft components lying within the vulnerability cones will be susceptible to damage. The exact region of space lying within a cone can be identified by applying a mesh to the satellite body, and calculating which of the grid elements lie within the cone. Each of these grid elements can then be assigned a 'vulnerability metric' to distinguish it as a region of higher vulnerability (see Figure 4-5).

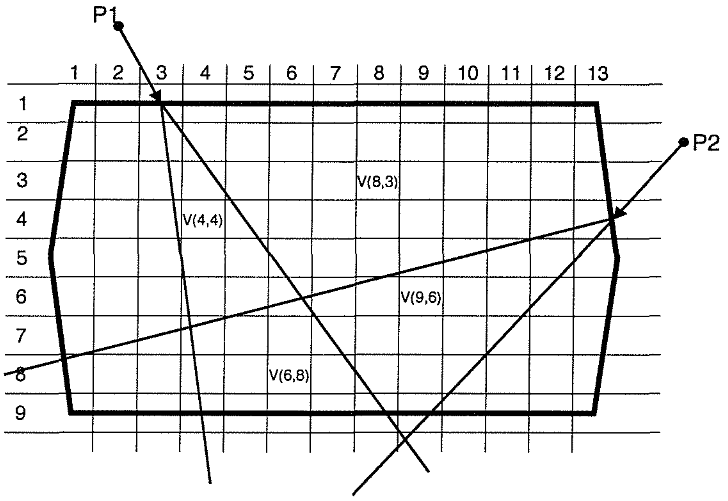


Figure 4-5 2-D illustration of vulnerability cones and metrics defining regions of high risk

The region of space occupied by grid element *(6,8)* is vulnerable to impact from two debris clouds, thus increasing its vulnerability metric. However, the element is located well away from the impact points; so only a small percentage of the cloud energies will pass through the element. Therefore, the vulnerability of element *(6,8)* may in fact be less than element *(4,4)*. Based on this logic one can define the vulnerability metric as follows:

$$V(m,n) = \sum_{i=1}^{I} E_i \left[\frac{A(m,n)}{A_i} \right] \qquad [4\text{-}3]$$

where *V(m,n)* is the vulnerability metric for grid element *(m,n)*, E_i is the energy of penetrating impactor *i*, *A(m,n)* is the mean cross-sectional projected area of grid element *(m,n)*, and A_i is the surface area of the debris cloud at grid element *(m,n)* generated by penetrating impactor *i*. Figure 4-6 illustrates this calculation for grid element *(6,8)*.

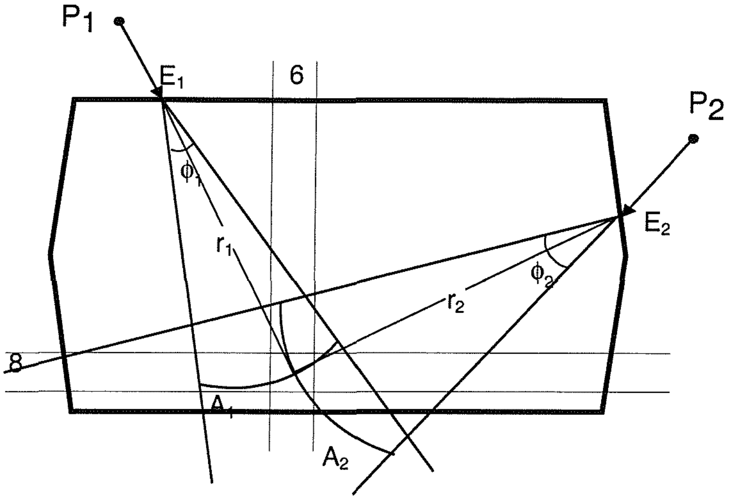


Figure 4-6 2-D illustration showing how a grid element vulnerability metric is calculated

The area A_i refers to that part of the surface of a sphere (radius r_i) contained within a vulnerability cone of spread angle ϕ_i. Therefore, it is assumed that all of the material within the debris cloud is spread evenly over this area.

Once all grid element metrics have been calculated in this manner, the result is a 2-D vulnerability matrix, or map, which characterises the most vulnerable internal regions of the satellite body.

Survivability Metric Derivation - Step 2

In the second step of the derivation, each of the components are overlaid onto the map, thereby enabling their vulnerabilities to be calculated (see Figure 4-7).

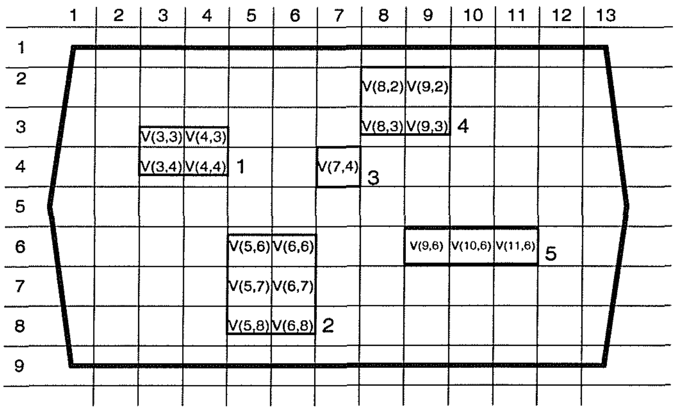


Figure 4-7 2-D illustration showing how component vulnerability metrics are calculated

For example, the vulnerability metric for component 1 is calculated by summing the metrics of the four grid elements it overlays, i.e.:

$$V_1 = V(3,3) + V(4,3) + V(3,4) + V(4,4) \qquad [4\text{-}4]$$

From the preceding definitions, one can see that this vulnerability metric is related to the total amount of impact energy that the component has to absorb assuming no other internal components are present. Of course, in practice there are probably many internal components that could absorb some of this energy. In fact, the more a component is surrounded by others, the less likely it is that it will have to absorb the energy from penetrative impactors. Therefore, one has to modify the component vulnerability metric to take this into account. To do this, a 'protection factor' is introduced. For each component, a ray is fired from its centre to the body structure, and then swept through 360° in 1° steps, rather like a radar sweep. Each time the ray encounters a neighbouring component the protection factor is incremented. One can express this mathematically as follows:

$$P_j = \sum_{k=1}^{360}\left(\sum_{l=1}^{j-1} R_l + \sum_{l=j+1}^{J} R_l\right) \qquad [4\text{-}5]$$

where P_j is the protection factor for component j, k is the angle swept out by the ray in degrees, and R_l is the amount of impact resistance provided by each neighbouring component l in the direction of the ray. The total number of internal components is J. If it is assumed that all internal components are electronics boxes with approximately equal resistance then one can normalise the value of R to 1 for each component.

Finally, a dimensionless factor called the component 'criticality' can be introduced. Clearly, a component that is very susceptible to impact should be a cause for concern if it is critical to mission success. However, a non-critical item with the same susceptibility does not merit the same level of concern. Therefore, a criticality factor provides a means of distinguishing between these two cases. The criticality factor of component j is defined as follows:

$$Q_j = \frac{X_j}{100} \qquad [4\text{-}6]$$

where X_j can be thought of as the percentage of the mission functionality or life that is lost if component j is lost. The evaluation of criticality is based largely on engineering judgement, and can be derived from reliability analysis techniques such as FMECA. For the purposes of the model, the FMECA criticality factors are defined as shown in Table 4-6.

Criticality factor	Effect on mission	Criteria
< 0.25	Negligible	❑ Component has a function that is peripheral to the main mission objective ❑ Component has a small amount of stored energy ❑ Extensive redundancy incorporated
0.25 – 0.50	Minor	❑ Component has a function that is not central to the main mission objective ❑ Some redundancy incorporated
0.50 – 0.75	Major	❑ Component has a function that is important to achieving the mission objective ❑ Little or no redundancy present
> 0.75	Catastrophic	❑ Component has a function that is vital to the mission objective ❑ Component has a very large amount of stored energy ❑ No redundancy present

Table 4-6 Definition of component criticality factors

Survivability Metric Derivation - Step 3

The combination of vulnerability metric, protection factor and criticality factor is sufficient to characterise the susceptibility of each internal component, and therefore the survivability of the overall spacecraft design. One can now define a metric to represent the spacecraft survivability as follows:

$$S = \sum_{j=1}^{J} \frac{P_j^{\mu} Q_j^{\kappa}}{1 + V_j^{\nu}} \qquad [4\text{-}7]$$

where μ, κ, and ν are coefficients that define the relative weighting, or importance, of P_j, Q_j, and V_j in the calculation of survivability S. In theory, values for the three coefficients could be determined through empirical investigation. But this is not a practical proposition within the confines of this thesis. Instead, for the purposes of investigations with the prototype model, it is not unreasonable to start by setting the values of the coefficients to 1.

If required, the spacecraft survivability metric can be adjusted to take into account the cost of applying any direct shielding to the spacecraft design. The addition of physical shielding not only increases the cost of constructing the satellite, but also increases the mass of the satellite. This in turn implies an increase in the cost to launch the satellite. To address this the following metric for cost-effective survivability was devised:

$$S = \left[\sum_{j=1}^{J} \frac{P_j Q_j}{\left(1+V_j\right)} \right] \Big/ \left[C_B + C_{Sh} + \alpha \left(M_B + M_{Sh} \right)^{\beta} \right] \qquad [4\text{-}8]$$

where C_B is the manufacturing cost of the satellite body structure, C_{Sh} is the total manufacturing cost of all external shielding, M_B is the body mass, M_{Sh} is the total mass of shielding, and α and β are dimensionless factors relating launch cost to mass. This definition of survivability is used in the model to quantify the effectiveness of a particular protection strategy, and thereby enable comparisons between competing design solutions.

4.3.3 Issues Concerning the Simple Survivability Evaluation Technique

The chosen approach for assessing the survivability of a design is potentially very fast computationally. However, some assurance is needed that the logic of the approach is sensible, and that any simplifying assumptions are appropriate. The following issues are especially pertinent:

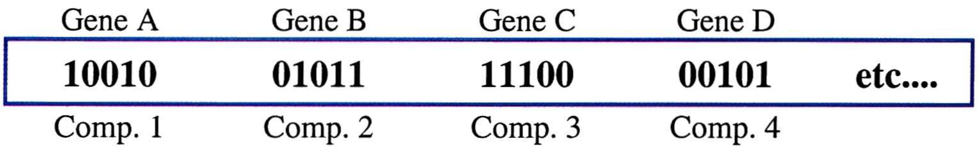
1. The number of penetrating impactors is crucial to the calculation of spacecraft vulnerability. In practice a real mission may experience few, if any, penetrators during its lifetime. Therefore, for the analysis to be statistically valid, it is necessary to derive a large distribution of impactors on the main satellite body from the impact fluxes input into the model. This in turn implies the need for longer run-times.
2. The trajectory and spread angle equations are derived for penetrating impacts on monolithic aluminium targets. Therefore, their application to penetrations of multi-layer shield designs is likely to produce erroneous results.

3. Vulnerability cones are used to represent the internal regions of greatest risk from penetrators. If the interior of a satellite body is hollow, then this is an accurate approach. Once components are added though, the situation becomes much more complex.
4. The definition of the vulnerability metric is related to the energy of the penetrating particles, rather than the energy of the resulting secondary debris clouds. This is done for simplicity. In practice, it would be better to use the cloud energy.
5. The energy within a debris cloud is assumed to be spread evenly over the surface of an expanding sphere within a vulnerability cone. There is much experimental evidence to contradict this assumption, which will be discussed in more detail later in the thesis.
6. The component vulnerability metric defines the impact risk experienced by a component in terms of impact energy. However, a component that receives the same amount of impact energy from one large impactor as opposed to, say, five smaller ones may in fact experience different levels of risk.
7. With the vulnerability of each component defined in terms of a metric, the 'hollow body' approximation is improved by introducing the 'protection factor' modification. Although this does not attempt to model the reduction in damage that a component will experience when intervening components absorb the energy from secondary debris clouds, it nevertheless gives a figure of merit to describe how well a component is protected based on its location relative to other components.

As first order approximations, all of the above assumptions and simplifications are reasonable for the prototype, as they do not detract from the main purpose of the investigation, which is to demonstrate the applicability of a genetic algorithm to the protection search problem. The approach has all the necessary features to allow rapid and potentially meaningful comparisons between competing protection solutions during the genetic algorithm search process. However, it is clear that a full development version of the model will require a more rigorous and computationally intensive approach.

4.3.4 Genetic Algorithm Optimisation Process

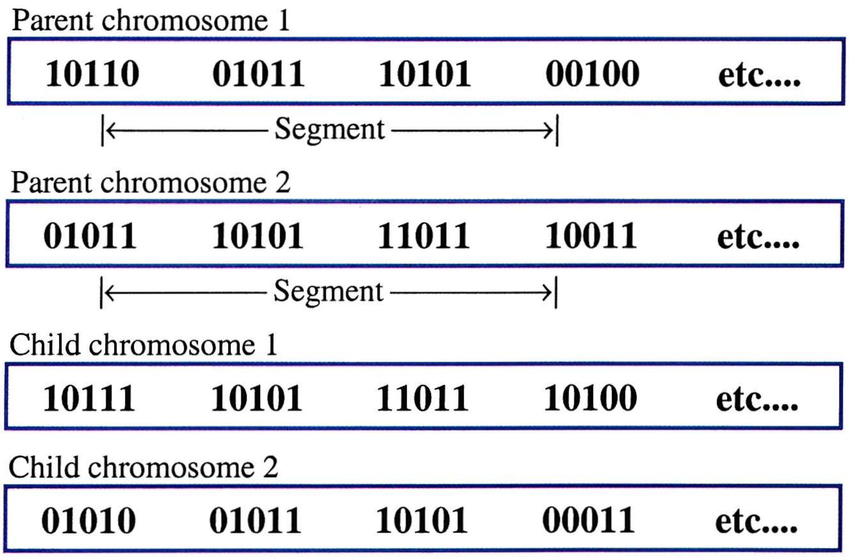
Once each spacecraft design in the initial population has been evaluated using the survivability metric method, the result is a population of competing solutions ranked according to the value of their metric. These are now converted into their evolutionary biology equivalent, i.e. a population of 'chromosomes' ranked according to their 'fitness'. A chromosome consists of a series of genes that have been spliced together. Each gene represents a variable in the design solution. For example, a gene could be the location of a particular internal component. Therefore, a design solution with ten components would be converted into a chromosome comprising a string of ten genes. The value of each gene (i.e. component location point) can be converted into a binary number, so the chromosome could take the form shown in Figure 4-8. That is, a long string of binary digits. This process is repeated for each of the design solutions in the randomly generated population.



Gene A	Gene B	Gene C	Gene D	
10010	**01011**	**11100**	**00101**	**etc....**
Comp. 1	Comp. 2	Comp. 3	Comp. 4	

Figure 4-8 Example of a 'chromosome' comprising 'genes' of design information

Three basic genetic algorithm operations are then performed on this population of chromosomes, namely selection, crossover, and mutation, to create a new population of 'child' chromosomes [Davis (1996)]. Selection occurs when two chromosomes are chosen from the population using a selection operator that is designed to pick out the 'fittest' chromosomes from the population more frequently. During crossover the two selected chromosomes are 'mated', i.e. segments of binary digits can be swapped between the chromosomes to create two new 'child' chromosomes (see Figure 4-9). The probability of crossover occurring is set at 0.7, which is a typical recommended value [Davis (1996)].



Parent chromosome 1

10110 01011 10101 00100 etc....

|← Segment →|

Parent chromosome 2

01011 10101 11011 10011 etc....

|← Segment →|

Child chromosome 1

10111 10101 11011 10100 etc....

Child chromosome 2

01010 01011 10101 00011 etc....

Figure 4-9 Example of information crossover between two chromosomes

Finally, the child chromosomes are mutated slightly to diversify the population. This simply involves randomly changing binary digits in each chromosome from zero to one, or vice versa. The probability that a digit mutates is set at 0.05, which again is a typical value.

The above sequence of operations is repeated until a completely new generation of child chromosomes has been created, equal in size to the original parent population. The two best parent chromosomes are added to this new population of children to ensure that the optimum is retained at all times. Each child chromosome is then converted back into a 2-D design configuration, e.g. the binary genes are translated into component locations, and the components arranged within the satellite body.

At this point in the process some care is needed to ensure that each child design is actually meaningful. For example, it is possible for the genetic algorithm to create a 'forbidden' design in which some of the components overlap each other and / or the satellite body structure. To overcome this, a 'clash detection' algorithm is invoked. A mesh is applied to the satellite body interior in much the same way as was used for the computation of the vulnerability map described earlier. A value is then assigned to each grid element of the mesh to indicate whether the space is empty or if it is occupied by one or more components. In the case where two components occupy at least one grid element then the components are progressively shuffled apart from each other until the grid element(s) show that the clash has been removed. The shuffling routine searches for the most efficient route to separate the components, as shown in the example in Figure 4-10. The end result is a child design that is now meaningful, and yet is still a good approximation to the original forbidden design.

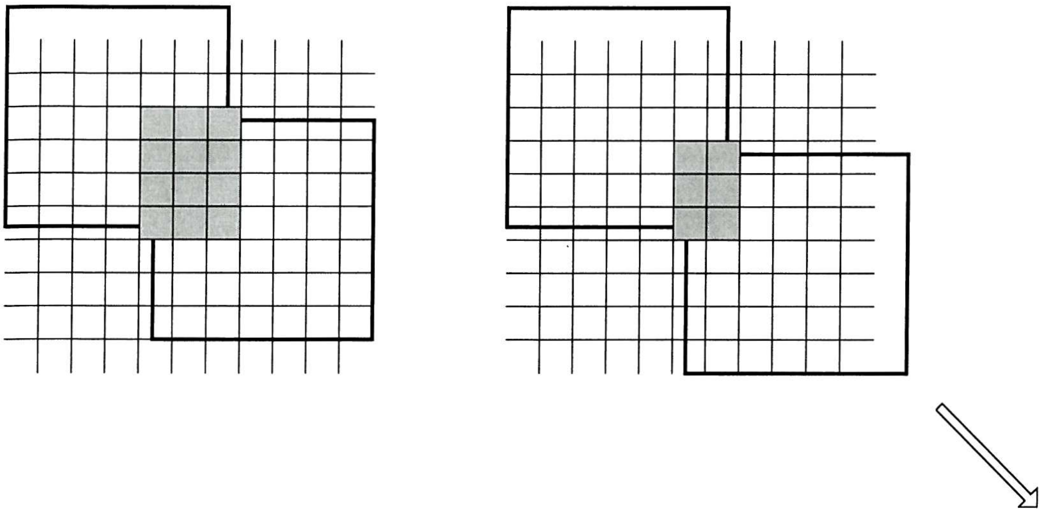


Figure 4-10 Clash detection algorithm for separation of two overlapping components

Once all child designs have been created in this fashion, then the result is a new population of competing designs that can be assessed in exactly the same manner as the initial randomly generated population. After N generations have elapsed, the final population should contain protection solutions that are close to the global optimum.

4.3.5 Results from the Prototype Model

Preliminary results from running the prototype model show that it can converge on a near-global optimum design solution within 100 generations, where each generation comprises 100 competing designs. Typical run-times are of the order of five hours on a Pentium PC.

Result 1

In the example simulation, illustrated in Figures 4-11 and 4-12, the model is optimising a ‘satellite’ comprising 9 internal components, all of which are identical with the exception that one component has a criticality factor of 0.9 (marked in red) compared to 0.1 for the others (marked in blue). All four choices of external shielding described earlier are available to protect the body, and it is assumed that the cost per unit mass of each shield option is the same. A set of 2000 impacting particles, each smaller than 10 g in size, is distributed randomly on the satellite body. That is, the distribution is not based on an environment flux model.

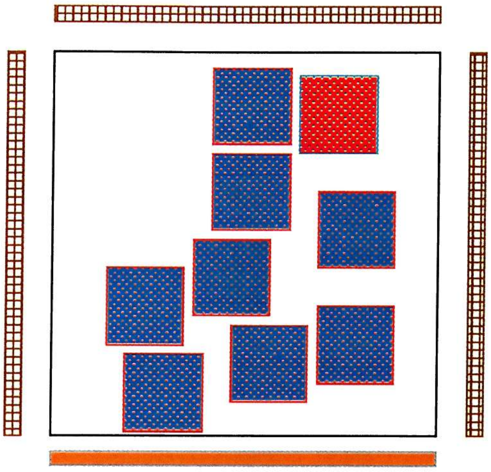


Figure 4-11 Best design in initial population

Figure 4-11 shows the best design in the initial randomly generated population, with a survivability metric of 0.0028. The external shielding comprises a mixture of mesh double bumpers (represented by the crosshatched layers in the figure) and Whipple bumper (represented by the monolithic layer). Inside the body the components are well spread out, and the critical component is quite exposed and vulnerable to impact. After 100 generations, the best design has a survivability metric of 0.0364, i.e. a factor of 13 improvement. The preferred method of external shielding on each face is mesh double bumper, and the internal components are clustered together to provide an improved degree of protection to the critical component (see Figure 4-12).

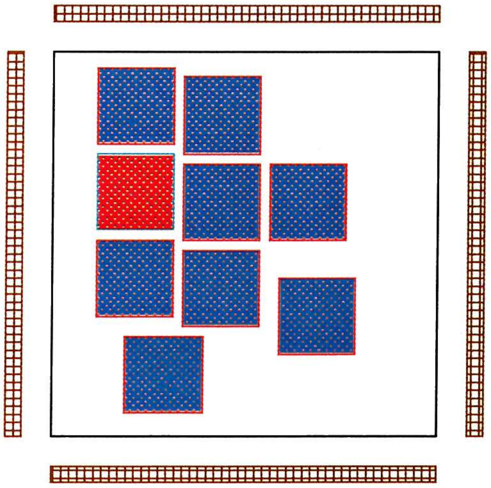


Figure 4-12 Best design in final population

One can check whether this internal arrangement is credible by examining the vulnerability map inside the satellite body (see Figure 4-13). The large pink shaded area represents a region of low vulnerability. As desired, the components in Figure 4-12 have a tendency to occupy this region.

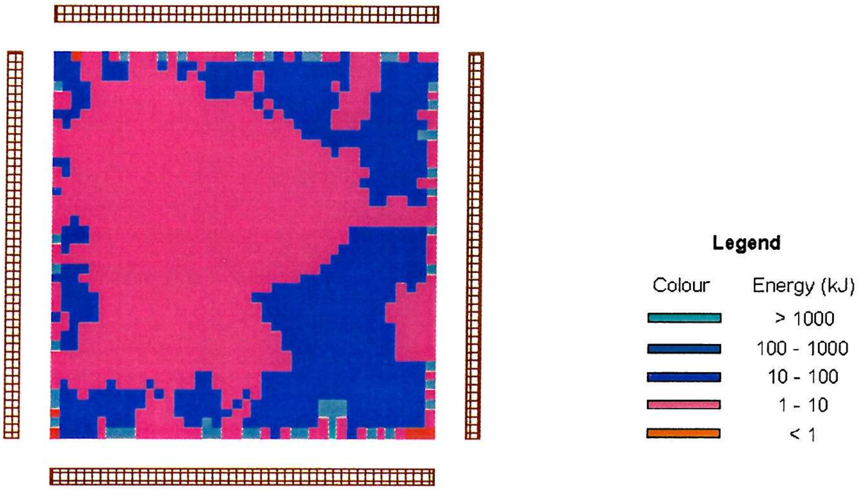


Figure 4-13 Vulnerability map inside satellite body

There is, however, a potential minor anomaly in the result that deserves mention. One might have expected the critical component to be completely surrounded by neighbours, rather than be located close to a spacecraft face. This suggests that the benefits of protection from neighbouring components may need to be enhanced in Step 2 of the survivability evaluation process. Two ways of achieving this are: first, to modify the coefficients in Equation 4-7, or second, to modify the protection factor in Equation 4-5 to include a term that would weight the importance of the protection along a given ray direction. This would strengthen the benefit derived when neighbouring components protect a critical component against the highest risk spacecraft faces.

The ease with which the model was able to focus on the near-optimum solution is illustrated by examining the growth in survivability metric throughout the generations. Figure 4-14 shows how the metric for the best design in each generation increases rapidly at first, followed by smaller increases later on. This is a good indication that the model is efficient at converging on the near-optimum solution.

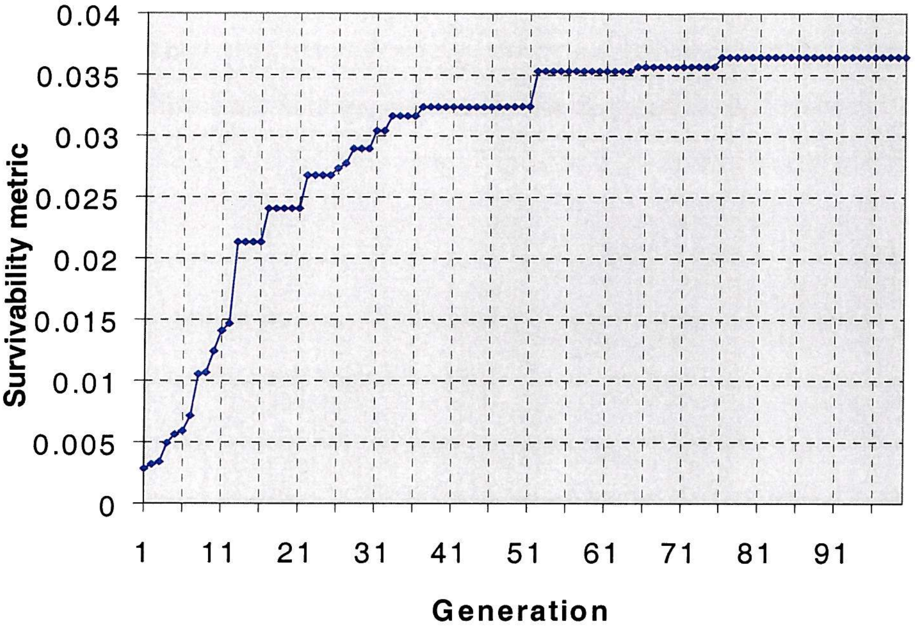


Figure 4-14 Survivability metric for best design in each generation

Result 2

In practice the design solution in Figure 4-12 would probably not be feasible because of poor mass balance and thermal balance. It is possible to consider constraints such as these by making an appropriate modification to the survivability metric. For example, to include a mass balance constraint, the metric in Equation 4-8 can be modified as follows

$$S = \left[\frac{r_{\max} - r_a}{r_{\max}}\right] \text{ x } \left[\sum_{j=1}^{J} \frac{P_j Q_j}{(1+V_j)}\right] \Big/ \left[C_B + C_{Sh} + \alpha (M_B + M_{Sh})^{\beta}\right] \qquad [4\text{-}9]$$

where r_a is the distance of the actual centre of mass of the body from the desired position, and r_{max} is the maximum possible distance that the centre of mass can deviate from the desired position. Repeating the earlier example simulation, a new optimum design solution is obtained (see Figure 4-15).

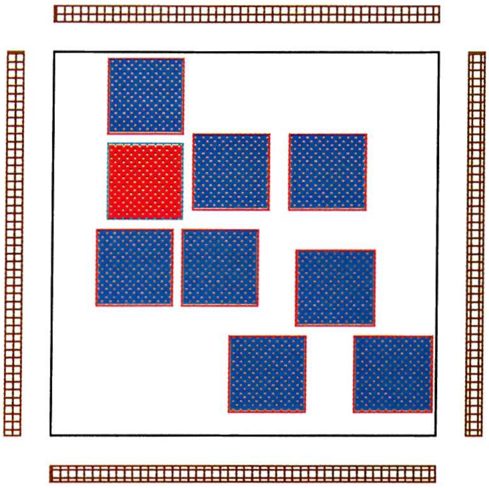


Figure 4-15 Best design in final population (subject to mass balance constraint)

The effect of including the mass balance constraint is clearly shown. The critical component is still located in the lowest vulnerability region, and is well protected by neighbouring non-critical components. However, two of the non-critical components now occupy more vulnerable regions. This improves the mass balance of the design, but causes a reduction in the survivability metric to 0.0318. The resulting optimum solution is therefore an illustration of the compromise between the need for protection and the need to satisfy other engineering constraints in the design.

Result 3

In reality, a 3-axis stabilised spacecraft in low Earth orbit will not experience a random (isotropic) impact distribution, as defined for the two results above. For example, referring back to Figure 3-9, it was shown that the ERS-2 satellite surfaces facing the velocity vector direction will be subjected to significantly higher impact fluxes than any other surfaces. The effect of such a flux distribution can be investigated here in a fairly simple manner by repeating the simulation in Result 1, but applying the random impact distribution only to one spacecraft face. All other variables in the simulation are identical to Result 1. In this case, the best design in the initial randomly generated population has a survivability metric of 0.1647, and the configuration shown in Figure 4-16.

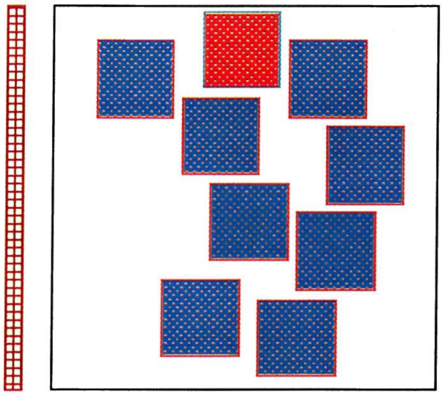


Figure 4-16 Best design in initial population

Within the first population the genetic algorithm has already identified that structural shielding is only required on one face, and that the mesh double bumper is the most effective choice. However, the internal arrangement of components is not yet ideal, with the critical unit poorly protected and placed too close to the high-risk face. One hundred generations later, though, the best design has a survivability metric of 0.3189, which is almost double that of the best in the initial population. The reason for this increase is that the internal components are clustered together to provide an improved degree of protection to the critical component, which is also located as far as possible from the high risk face (see Figure 4-17). This result is exactly what one would hope to see, and provides further reassurance that the genetic algorithm search technique is suitable for the protection optimisation problem.

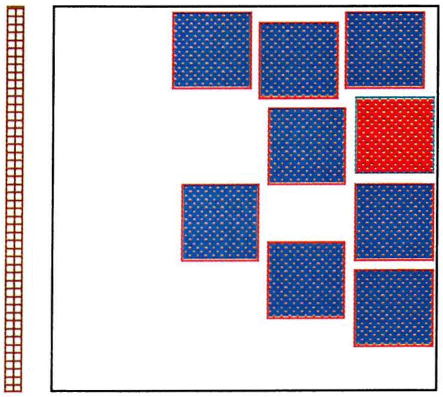


Figure 4-17 Best design in final population

As before, it is worth confirming that the internal arrangement is credible by examining the vulnerability map inside the satellite body (see Figure 4-18). The large orange shaded area represents the region of lowest vulnerability. Referring to Figure 4-17, one can see that all of the components have migrated to this region.

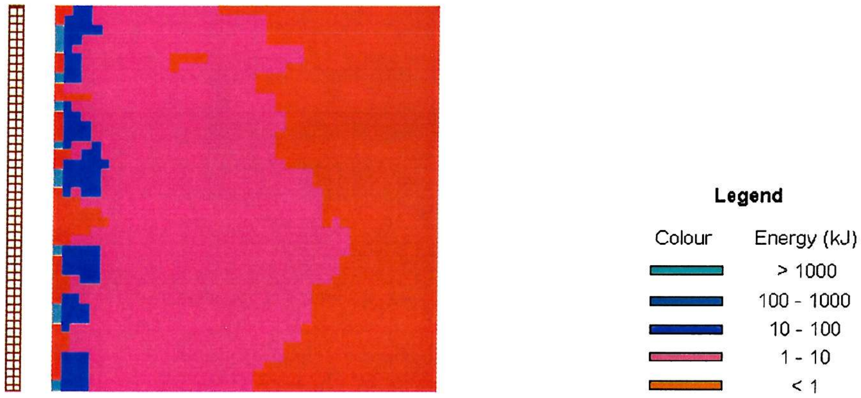


Figure 4-18 Vulnerability map inside satellite body

4.4 Discussion

The three results from the prototype model are extremely encouraging for a number of reasons:

1. The survivability evaluation technique, despite several simplifications and assumptions, was sufficiently performant to allow meaningful design comparisons. More importantly, it enabled the genetic algorithm search process to function well enough to satisfy the requirements of the prototype. Namely, to demonstrate that a genetic algorithm appears to be a fast and efficient method for searching through numerous spacecraft protection designs and converging on a credible solution.
2. The results highlight the interdependency between the choice and location of structural shielding and the vulnerability and arrangement of components inside the satellite. This suggests that the derivation of a well-considered protection solution is best achieved by simultaneously considering both of these factors. It also underlines the importance of considering appropriate protection measures during the early phases of a satellite design, rather than as an 'add-on extra' in the later stages.

3. The successful extension of the prototype to include a design constraint, i.e. satellite mass balance, suggests that the modelling approach could be used to address more complex optimisations. For example, other constraining factors such as thermal balance might also be included. Traditional design methodologies usually address each of these problems individually and sequentially until a satisfactory (not necessarily cost-effective) design solution is attained. The new approach benefits by being able to address these problems simultaneously, thus ensuring better convergence.

The pursuit of a full development version of the model can therefore be undertaken with some degree of confidence. Experience from the prototype model shows that perhaps the greatest challenge during this next phase will be the construction of a rigorous survivability evaluation methodology.

5 Description of the SHIELD Model

5.1 Introduction

The prototype model successfully demonstrated a method for searching and evaluating different satellite shielding options and internal component configurations. However, for expediency, the prototype included a number of simplifications such as: 2-D representation of a satellite; artificially created impact distributions; and a simple method for distinguishing the effectiveness of competing solutions. In the full version of the model, these simplifications must be removed and replaced by more realistic and robust methodologies. Hence, the purpose of the full 3-D model, hereafter referred to as SHIELD, is defined as follows:

- To evaluate the impact and penetration risk to an unmanned spacecraft.
- To assess the internal damage caused by particles penetrating an unmanned spacecraft.
- To determine the debris-induced probability of failure of an unmanned spacecraft.
- To identify the most cost-effective protection strategy for an unmanned spacecraft.

Impact and penetration risk computations are relatively straightforward, and have been implemented already in other models such as ESA's ESABASE/DEBRIS and NASA's BUMPER. However, the other three items on this list have not, and so there is no precedent upon which to base the SHIELD model. This represents a major challenge, and therefore requires some innovation in the software development.

The remainder of this chapter sets out to describe the SHIELD model, and explain how these issues have been addressed. A summary of the model is also given in Stokes *et al* (1999).

5.2 The Software Engineering Process

It is good practice to adopt formal software engineering standards in the development of any complex program to ensure a robust design. For the SHIELD software, ESA's PSS-05 standards [Mazza *et al* (1994)] have been implemented.

Because software projects can vary widely in size, complexity and purpose, an essential first step is to define within a software project management plan how the standards will be applied.

The SHIELD management plan [Stokes (1997a)] identifies a number of influencing factors, such as project costs and staffing, users, and software criticality and complexity. For SHIELD, the project was evaluated to be 'small' in size, and so the standards were tailored accordingly. In particular, the simple life cycle approach shown in Figure 5-1 was adopted.

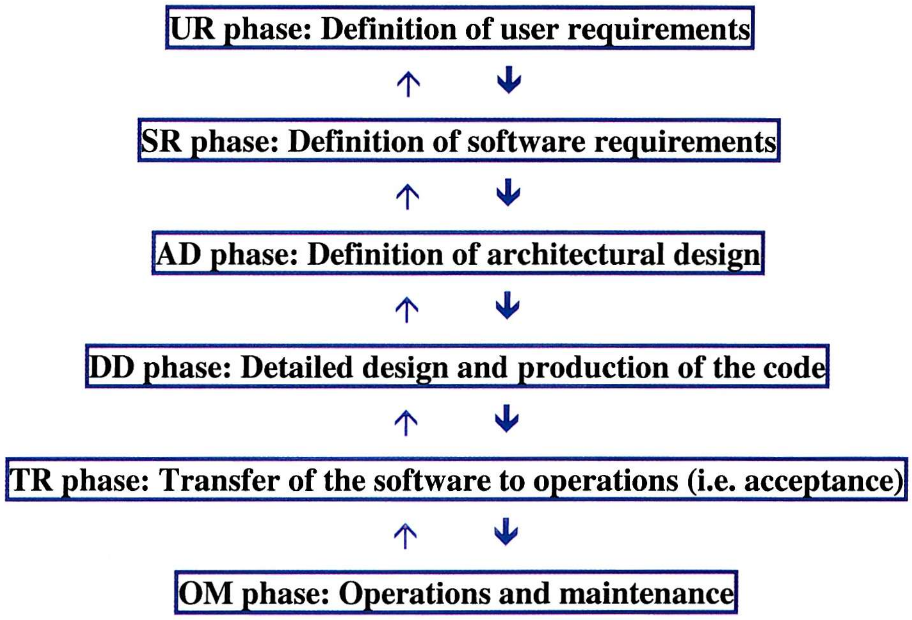

UR phase: Definition of user requirements
SR phase: Definition of software requirements
AD phase: Definition of architectural design
DD phase: Detailed design and production of the code
TR phase: Transfer of the software to operations (i.e. acceptance)
OM phase: Operations and maintenance

Figure 5-1 Software development life cycle (waterfall approach)

This is known as the 'waterfall approach' (the arrows represent permitted transitions). The phases are executed sequentially, as shown by the heavy arrows. Each phase is executed once, although iteration of part of a phase is allowed for error correction, as shown by the lighter arrows. One can see that the life cycle starts when the software product is conceived and ends when it is no longer available for use, i.e. the life cycle contains the whole of the development, operations and maintenance activities.

The UR phase is the 'problem definition phase', where the scope of the system must be defined and the user requirements captured. The SR phase is the 'analysis phase' of the project. An essential element of the activity is the construction of a 'model' describing what the software will do (as opposed to how it will do it). The purpose of the AD phase is to define the structure of the software. The model constructed in the SR phase is transformed into an architectural design by allocating functions to software components and defining the control and data flow between them. In the DD phase, the detail of the software design is defined, and the software coded, documented, and tested. The TR phase establishes whether the software fulfils the requirements laid down in the UR phase. A series of acceptance tests are conducted and, assuming these are successful, the software can be provisionally accepted. In the last (OM) phase the software enters operation and is carefully monitored. Final acceptance occurs when the software has demonstrated satisfactory operation over a sustained period of time.

The SHIELD software model development has followed this process through all stages, and the results have been documented in a User Requirements Document (URD) [Stokes (1997b)], Software Requirements Document (SRD) [Stokes (1997c)], and Architectural Design Document (ADD) [Stokes (1997d)]. A Detailed Design Document (DDD) is currently in production.

5.3 SHIELD Requirements, Architecture Design, and Operational Attributes

The complete set of user and software requirements is summarised in Tables 5-1 and 5-2, respectively.

The architecture design of a complex program is usually presented in a diagrammatic flow chart format. This is the software's 'physical model', and provides a visual summary of its functionality. The flow chart for SHIELD, at the highest level of functionality, is shown in Figure 5-2. It is derived entirely from the requirements identified in Tables 5-1 and 5-2.

After entering standard data on the project and satellite mission, the model allows the user to perform the following main functions:

- ❑ Construct and view a 3-D representation of a satellite
- ❑ Simulate debris impacts on the satellite during its mission life
- ❑ Evaluate the survivability of a specific satellite design
- ❑ Optimise the debris protection strategy of a satellite

Req. #	Requirement description
UR1	**Construct a 3-D representation of a satellite:** □ Define satellite by: ▪ a main structural body ▪ up to 50 internal components ▪ up to 10 internal shelves ▪ up to 10 internal structure walls ▪ up to 10 distinct shield types selectable for each surface of the satellite body ▪ up to 10 distinct shield types selectable for each surface of a component □ Build satellite by: ▪ defining orientation of body coordinate system with respect to satellite velocity vector ▪ selecting body and components from a library of pre-built geometries ▪ defining shape, dimensions, material properties, mass, and cost of body, components, shelves, walls, and shields ▪ defining location of each internal shelf, internal structure wall, and shield ▪ defining location and orientation of each component ▪ defining criticality and redundancy of each component
UR2	**View a 3-D representation of a satellite:** □ Utilise features such as: ▪ rotation of satellite about x, y, and z axes ▪ zoom in / out ▪ wire-frame and rendering □ Use colour scheme to visualise different elements of the satellite design □ Visualise ray-traces of impacting particles
UR3	**Identify mission-related parameters, such as:** □ Operational life □ Attitude orientation □ Life cycle cost estimating relationships, e.g. launch cost / mass
UR4	**Simulate debris impacts on satellite during mission life:** □ Access target-centred directional flux data from IDES software □ Use fluxes to generate test particles □ Use ray-trace method to determine particle intersections with satellite geometry
UR5	**Evaluate satellite survivability:** □ Create large distribution of penetrating impactors on satellite, i.e.: ▪ repeat impact simulation process several thousand times ▪ use ballistic limit equations to check which impactors penetrate body (plus any shielding) ▪ when a particle penetrates, calculate trajectory of cloud inside the body □ Determine interaction of each penetrating cloud with satellite's internal equipment: ▪ which components damaged ▪ extent of damage and consequences to mission □ Determine probability of failure of satellite □ Determine cost of satellite (including cost of any shielding) □ Determine survivability metric (for use in comparing satellite designs)
UR6	**Optimise satellite protection strategy:** □ Specify satellite design variables to be optimised (from a pre-defined list), e.g.: ▪ component locations ▪ shield choices and locations □ Use genetic algorithm (with survivability metric) to search for optimum protection strategy in an automated fashion

Table 5-1 User requirements for SHIELD model

Req. #	Requirement description
SR1	**Resources required for source code compilation:** □ Pentium PC (Minimum: 300 MHz, 1 GB hard disk, 60 MB RAM) □ Windows 95/98/NT □ Code to be written in C using MS Visual C++ □ Library of 3-D graphics functions written in C / C++
SR2	**Performance required:** □ User data entry via a graphical user interface (GUI) □ One lifetime impact simulation (all particle sizes) to execute in less than 1 hour □ One thousand lifetime impact simulations (penetrator size particles only) to execute in less than 10 hours □ Evaluation of survivability of one satellite design in less than 5 minutes □ Full genetic algorithm optimisation (10,000 satellite designs) in less than 100 hours

Table 5-2 Software requirements for SHIELD model (excluding functional requirements already identified in Table 5-1)

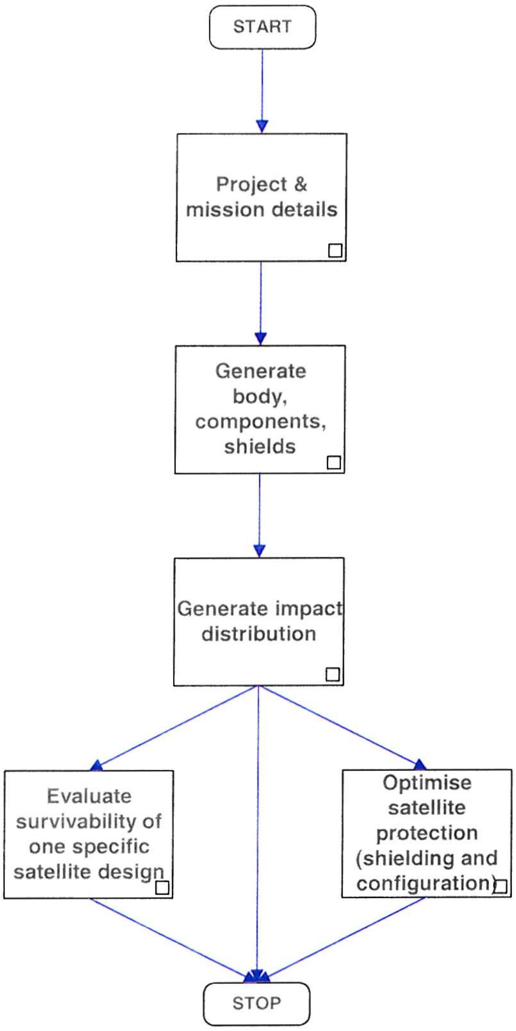

START
Project & mission details
Generate body, components, shields
Generate impact distribution
Evaluate survivability of one specific satellite design
Optimise satellite protection (shielding and configuration)
STOP

Figure 5-2 Flow chart of SHIELD software 'physical model'

The total number of C source lines of code required to implement the SHIELD design, excluding standard functions and graphics libraries, exceeds 30,000. The code is divided into 7 distinct executable files, each of which is compiled from a set of '.c' source code files. The code is highly modular, thus allowing functions in several of the '.c' files to be used in more than one executable. Table 5-3 illustrates the structure of the code.

The SHIELD software is designed to operate on a standard Pentium PC under MS Windows 95/98/NT operating systems. A description of the encoded design is provided in the following sections.

5.4 3-D Representation of a Satellite

All spacecraft perform certain basic functions in order to enable their payloads to function properly [Tribble (1995)]. A spacecraft usually has a propulsion subsystem to boost the spacecraft to its intended orbit, and to enable the spacecraft to maintain its orbit or de-orbit at the end of its mission. There must be an electrical power subsystem to provide power to the payload and other subsystems. A thermal control subsystem must maintain the spacecraft within the proper operating temperature range. An attitude determination and control subsystem is needed to orient the vehicle and ensure the payload is pointing at its desired point of reference. A data handling subsystem is required to route electrical commands around the vehicle. The transmission of data to, and receipt of instructions from, the ground requires a telemetry, tracking, and communications subsystem. Finally, a physical structure is needed to accommodate all of the above mentioned subsystems and the payload. Table 5-4 summarises the subsystems.

The first step in the model is to define a 3-D representation of a satellite that possesses all of these subsystems. The requirements in Tables 5-2 and 5-3 reveal that a user must enter various geometrical and material property data to define the satellite body, its internal structure walls and shelves, and any external shielding. Similarly, the system components inside the body must be defined in terms of their geometry, material properties, mission criticality, location and orientation.

Executables	‘.c’ Source Files	Source File Description
mission.exe	mission.c	Mission data entry
Inputs – user data	funx_mth.c	Common mathematical functions
Outputs – ‘mission.txt’ file		
body_bld.exe	body_bld.c	Satellite body data entry
Inputs – user data	funx_mth.c	Common mathematical functions
Outputs – ‘body.txt’ file	graph_3d.c	Generates GUI; 3D representation of body
comp_bld.exe	comp_bld.c	Equipment data entry
Inputs – user data	funx_mth.c	Common mathematical functions
Outputs – ‘comp.txt’ file	graph_3d.c	Generates GUI; 3D representation of components
impacts.exe	impacts.c	Calculates impact distribution on satellite
Inputs – user data	funx_mth.c	Common mathematical functions
‘mission.txt’ file	graph_3d.c	Generates GUI; 3D representation of satellite + impacts
‘body.txt’ file		
‘comp.txt’ file		
‘ides.txt’ file		
Outputs – ‘impacts.txt’ file		
seconds.exe	seconds.c	Calculates penetration distribution on satellite
Inputs – user data	funx_mth.c	Common mathematical functions
‘mission.txt’ file	graph_3d.c	Generates GUI; 3D representation of satellite + penetrators
‘body.txt’ file	ball_lim.c	Calculates ballistic limits; identifies penetrators
‘comp.txt’ file	clouds.c	Calculates secondary debris cloud properties
‘impacts.txt’ file		
Outputs – ‘seconds.txt’ file		
init_pop.exe	init_pop.c	Generates initial random population of designs
Inputs – user data	funx_mth.c	Common mathematical functions
‘mission.txt’ file	graph_3d.c	Generates GUI; 3D representation of satellite designs
‘body.txt’ file	planes.c	Calculates equation of plane of every satellite surface
‘comp.txt’ file	comp_fix.c	Attaches each component to the satellite
‘seconds.txt’ file	comp_rot.c	Correctly orientates each attached component
Outputs – ‘popn_1.txt’ file	matrices.c	Used to identify overlapping components
	clashes.c	Identifies clashes between components, etc.
	ball_lim.c	Calculates depth of particle penetration inside a satellite
	obj_fn.c	Calculates survivability of a particular satellite design
gen_pop.exe	gen_pop.c	Generates N genetic populations of designs
Inputs – user data	genetics.c	Performs genetic algorithm functions
‘mission.txt’ file	funx_mth.c	Common mathematical functions
‘body.txt’ file	graph_3d.c	Generates GUI; 3D representation of satellite designs
‘comp.txt’ file	planes.c	Calculates equation of plane of every satellite surface
‘seconds.txt’ file	comp_fix.c	Attaches each component to the satellite
‘popn_(N-1).txt’ file	comp_rot.c	Correctly orientates each attached component
Outputs – ‘popn_N.txt’ file	matrices.c	Used to identify overlapping components
	clashes.c	Identifies clashes between components, etc.
	ball_lim.c	Calculates depth of particle penetration inside a satellite
	obj_fn.c	Calculates survivability of a particular satellite design

Table 5-3 Structure of SHIELD software

Subsystem	Purpose	Key features
Attitude determination and control	Vehicle stability and pointing control	Reaction wheels Momentum wheels Sun / Earth sensors Magnetorquers
Data handling	Data and command relay to payload and subsystems	Data bus Processing Memory
Electrical power	Power generation and distribution	Solar arrays Batteries Load control electronics
Propulsion	Manoeuvre vehicle into desired orbit	Thrusters Fuel Tanks, plumbing
Structures	Integrity during launch and manoeuvres	Bulkheads Mechanisms
Telemetry, tracking, and communications	Command and data handling with the ground	Transmitters Receivers Antennae
Thermal control	Maintain temperature balance	Radiators Heat pipes Multi Layer Insulation Anodised aluminium

Table 5-4 Spacecraft engineering subsystems

First, a standard x, y, z cartesian coordinate system is constructed such that one of the axes is aligned with the velocity vector of the satellite (i.e. in the orbital along-track direction), another is aligned in the orbit's radial direction, and the third is aligned in the orbit's out-of-plane direction. The entire satellite geometry is then defined in this coordinate system, with the origin at the centre of the satellite.

To construct the satellite body, a library of pre-defined geometrical shapes is available (if the body geometry is relatively simple). Alternatively, the user can construct a more complex, irregular geometry by entering specific coordinate data for the individual surfaces. Whichever method is used, the coordinates defining each surface must lie on a plane. That is, a satellite geometry is represented in the software as a composite of many planes. This means that the software must calculate the vector equation of each plane from the coordinates of at least three points on the plane. The derivation is as follows:

Assuming that O is the origin, and the three coordinates $A(x_1, y_1, z_1)$, $B(x_2, y_2, z_2)$, and $C(x_3, y_3, z_3)$ lie on a plane (see Figure 5-3), then the following three vectors can be calculated:

$$\overrightarrow{AB} = (x_2 - x_1)\mathbf{i} + (y_2 - y_1)\mathbf{j} + (z_2 - z_1)\mathbf{k} \qquad [5\text{-}1]$$

$$\overrightarrow{BC} = (x_3 - x_2)\mathbf{i} + (y_3 - y_2)\mathbf{j} + (z_3 - z_2)\mathbf{k} \qquad [5\text{-}2]$$

$$\overrightarrow{AC} = (x_3 - x_1)\mathbf{i} + (y_3 - y_1)\mathbf{j} + (z_3 - z_1)\mathbf{k} \qquad [5\text{-}3]$$

where **i**, **j**, and **k** are unit vectors. If $\overrightarrow{ON}$ is a perpendicular vector, with length a, from O to the plane, then

$$\mathbf{n} = \overrightarrow{ON} = \kappa\mathbf{i} + \lambda\mathbf{j} + \mu\mathbf{k} \qquad [5\text{-}4]$$

where κ, λ and μ are the rectangular components of this normal vector. And so, from the scalar product of two perpendicular vectors,

$$\overrightarrow{ON}.\overrightarrow{AB} = (\kappa\mathbf{i} + \lambda\mathbf{j} + \mu\mathbf{k}).((x_2 - x_1)\mathbf{i} + (y_2 - y_1)\mathbf{j} + (z_2 - z_1)\mathbf{k}) = 0 \qquad [5\text{-}5]$$

$$\Rightarrow \quad (x_2 - x_1)\kappa + (y_2 - y_1)\lambda + (z_2 - z_1)\mu = 0 \qquad [5\text{-}6]$$

$$\overrightarrow{ON}.\overrightarrow{BC} = (\kappa\mathbf{i} + \lambda\mathbf{j} + \mu\mathbf{k}).((x_3 - x_2)\mathbf{i} + (y_3 - y_2)\mathbf{j} + (z_3 - z_2)\mathbf{k}) = 0 \qquad [5\text{-}7]$$

$$\Rightarrow \quad (x_3 - x_2)\kappa + (y_3 - y_2)\lambda + (z_3 - z_2)\mu = 0 \qquad [5\text{-}8]$$

$$\overrightarrow{ON}.\overrightarrow{AC} = (\kappa\mathbf{i} + \lambda\mathbf{j} + \mu\mathbf{k}).((x_3 - x_1)\mathbf{i} + (y_3 - y_1)\mathbf{j} + (z_3 - z_1)\mathbf{k}) = 0 \qquad [5\text{-}9]$$

$$\Rightarrow \quad (x_3 - x_1)\kappa + (y_3 - y_1)\lambda + (z_3 - z_1)\mu = 0 \qquad [5\text{-}10]$$

From the simultaneous Equations 5-6, 5-8, and 5-10, values for κ, λ and μ are easily determined. So too is the normal vector **n** (i.e. Equation 5-4). The equation of the unit normal vector is also given readily as follows:

$$\hat{\mathbf{n}} = \mathbf{n}/|\mathbf{n}| = (\kappa\mathbf{i} + \lambda\mathbf{j} + \mu\mathbf{k})\big/\sqrt{\kappa^2 + \lambda^2 + \mu^2} \qquad [5\text{-}11]$$

Now, the resolved part of vector $\overrightarrow{OA} = x_1\mathbf{i} + y_1\mathbf{j} + z_1\mathbf{k}$ in the direction of ON, i.e. the distance a, is:

$$(x_1\mathbf{i} + y_1\mathbf{j} + z_1\mathbf{k}).(\kappa\mathbf{i} + \lambda\mathbf{j} + \mu\mathbf{k})\Big/\sqrt{\kappa^2 + \lambda^2 + \mu^2} \qquad [5\text{-}12]$$

which is

$$(x_1\kappa + y_1\lambda + z_1\mu)\Big/\sqrt{\kappa^2 + \lambda^2 + \mu^2} \qquad [5\text{-}13]$$

Therefore, if **r** is the position vector of point A on the plane, the required equation is given by:

$$\mathbf{r}.\hat{\mathbf{n}} = a \qquad [5\text{-}14]$$

which is

$$\mathbf{r}.(\kappa\mathbf{i} + \lambda\mathbf{j} + \mu\mathbf{k})\Big/\sqrt{\kappa^2 + \lambda^2 + \mu^2} = (x_1\kappa + y_1\lambda + z_1\mu)\Big/\sqrt{\kappa^2 + \lambda^2 + \mu^2} \qquad [5\text{-}15]$$

That is,

$$\mathbf{r}.(\kappa\mathbf{i} + \lambda\mathbf{j} + \mu\mathbf{k}) = (x_1\kappa + y_1\lambda + z_1\mu) \qquad [5\text{-}16]$$

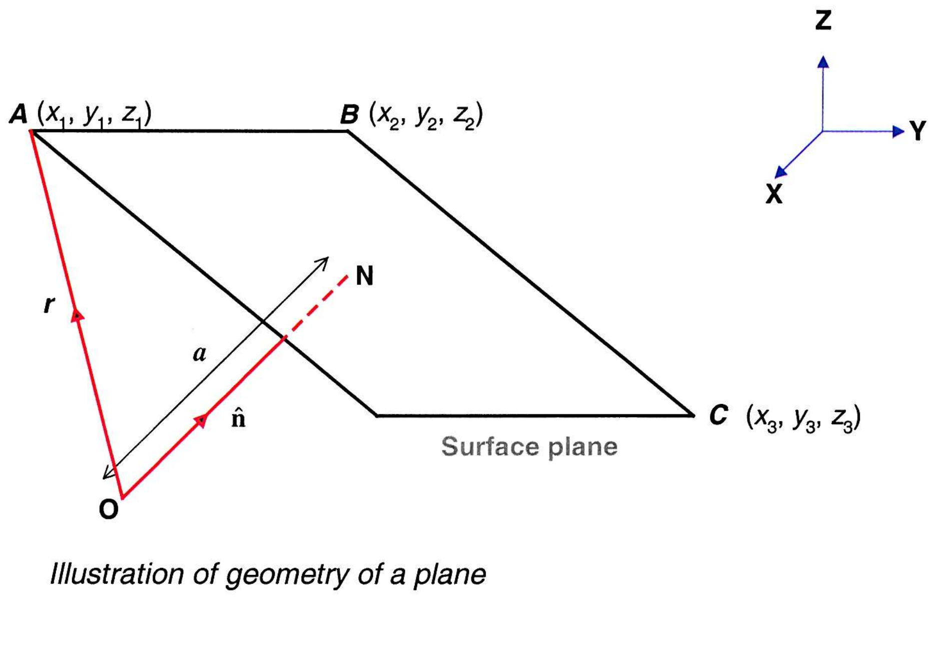


Figure 5-3 Illustration of geometry of a plane

When all of the equations and geometrical information for the body planes have been defined, they are stored in a large data structure that contains the entire body data set. The user is then required to input various non-geometrical properties associated with each of the body's surfaces. These include:

- material properties, such as density, speed of sound, brinell hardness, and thickness or areal density
- shield properties, such as type (e.g. stuffed Whipple), spacing, and thickness or areal density.

This completes the main satellite body definition. Next, any internal shelving and structure walls must be defined. In the software, walls are assumed to provide the necessary structural integrity during launch, as well as supporting shelves and internal equipment. Shelves are included primarily as a means of supporting equipment. A shelf or wall is constructed in almost exactly the same fashion as a satellite body surface, i.e. a geometric plane with material properties assigned to it. Once defined, shelves are positioned and aligned in the satellite such that they are all perpendicular to one user-defined axis in the body coordinate system. By contrast, walls can be positioned and aligned in any plane of the user's choosing. Figure 5-4 provides a simple illustration of a satellite body containing three shelves and a wall.

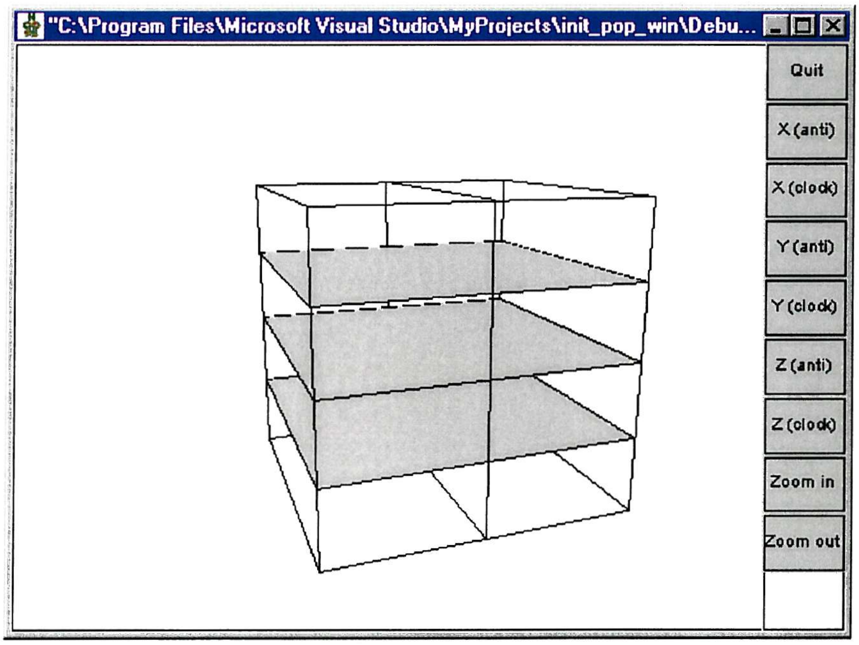

"C:\Program Files\Microsoft Visual Studio\MyProjects\init_pop_win\Debu...
Quit
X (anti)
X (clock)
Y (anti)
Y (clock)
Z (anti)
Z (clock)
Zoom in
Zoom out

Figure 5-4 Geometrical representation of a simple satellite structure

The introduction of equipment is the final step in the 3-D representation of a satellite. Essentially, equipment items are major subsystem or payload units. These are referred to as components in the model. Each component is constructed in almost exactly the same manner as the main satellite body. However, there are some notable differences, mainly due to the fact that a component, once specified, has to be attached to the satellite. These are:

- A component can be attached to any point on a body face, shelf, wall, or any user defined point in the satellite. To do this, an attachment point on the component has to be specified. This is usually the centre of one of the component's surfaces.
- Each component is defined in terms of its own coordinate system. Therefore, to attach a component to a point somewhere inside the satellite body, the software must perform a matrix transformation on the coordinates of each of the component's points. That is, the following translation is calculated:

$$(x_c, y_c, z_c, 1)T = (x_b, y_b, z_b, 1) \qquad [5\text{-}17]$$

  where x_c, y_c, z_c are the coordinates of the component prior to translation; x_b, y_b, z_b are the coordinates of the component when placed inside the body; and T is the translation matrix, which takes the following form:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ x_{ab} - x_{ac} & y_{ab} - y_{ac} & z_{ab} - z_{ac} & 1 \end{pmatrix} \qquad [5\text{-}18]$$

  where the coordinates with subscript *ac* refer to the attachment point defined for the component, and the *ab* subscript coordinates represent the body attachment point.
- In addition to a translation, the component may also need to undergo a series of rotations so that its attachment surface is coplanar with the body face, shelf, or wall to which it is attached. By defining a new coordinate system, with its origin at the component / body attachment point, and converting all of the component's points to this system, then the component can be rotated about the new origin using the following matrices:

$$R_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_x & \sin\alpha_x & 0 \\ 0 & -\sin\alpha_x & \cos\alpha_x & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad [5\text{-}19]$$

$$R_y = \begin{pmatrix} \cos\alpha_y & 0 & -\sin\alpha_y & 0 \\ 0 & 1 & 0 & 0 \\ \sin\alpha_y & 0 & \cos\alpha_y & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad [5\text{-}20]$$

$$R_z = \begin{pmatrix} \cos\alpha_z & \sin\alpha_z & 0 & 0 \\ -\sin\alpha_z & \cos\alpha_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad [5\text{-}21]$$

where α_x, α_y, α_z are the angles of rotation about the x, y, and z axes, respectively.

- Finally, the component must be correctly orientated on the body surface, shelf, or wall by the application of a further rotation.

With all components defined and positioned accordingly, it is desirable to view the full 3-D satellite representation graphically. For this purpose, the software employs a library of pre-built 3-D graphics functions called HOOPS 3D [Leler & Merry (1996)]. The functions are written in C++, and therefore can be compiled with the rest of the SHIELD C code relatively easily. Essentially, the functions transform the coordinates of the above objects into a set of 3-D images within a window. By changing certain parameters in the functions, the images can be moved, rotated, and displayed as required. To give the software a standard graphical user interface (GUI) appearance, various buttons have also been added within the window. These link to the function parameters, thereby enabling the 3-D view to be changed at the click of a mouse button.

5.5 Simulation of Debris Impacts

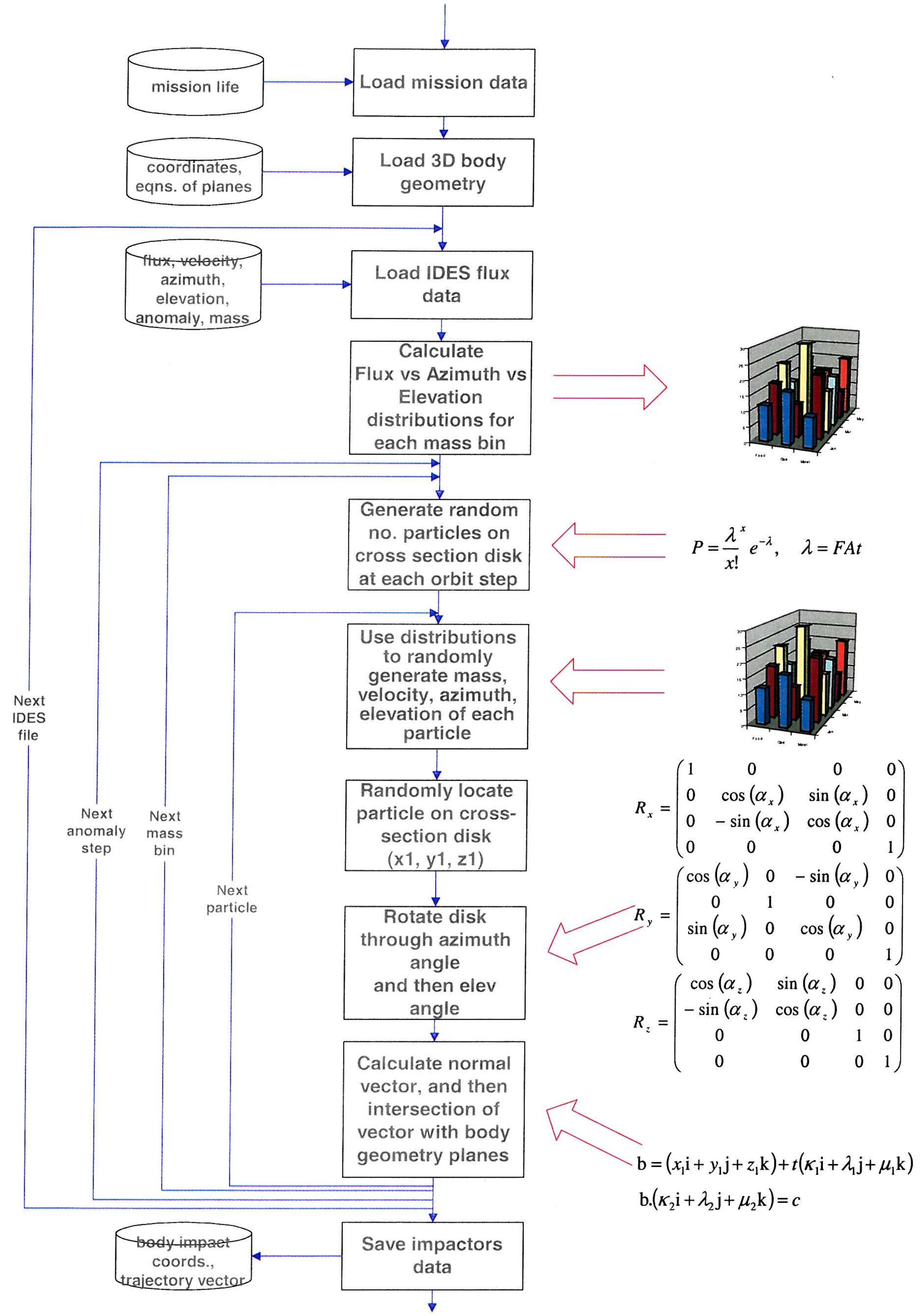
The first simulation function in the model determines the debris impacts experienced by a satellite during its mission life. That is, it computes the particle impact distribution on the satellite. A flow diagram illustrating this function is shown in Figure 5-5.

Three main inputs are required: the satellite's mission data, the external 3-D satellite geometry defined in Section 5.4 (i.e. excluding the internal components, shelves and walls), and a data-set of target-centred directional collision fluxes for the mission, which is derived from QinetiQ's IDES model.

5.5.1 Determination of Target-Centred Debris Collision Flux

The determination of debris flux is an essential element of IDES, and so the following description is distilled from Walker (2000). The IDES approach is based on a method developed by Klinkrad (1993), where the LEO debris flux environment is represented by a 3-dimensional inertial control volume, which is divided into volume cells by the spherical coordinate parameters of geocentric radius, declination and right ascension (as shown in Figure 5-6).

For any particular debris object, with known orbital elements, the method determines the true anomalies of its orbit as it intersects the respective cell borders of the inertial control volume. From this, the intersection times are readily calculated, and so too the residential probability of the object during its passage through a cell volume. Spatial density is also found relatively easily by dividing the residential probability by the cell volume. The velocity vector of the debris object is defined at the mid-point of cell passage by the velocity magnitude, azimuth, and elevation. Expressions for these are given by Klinkrad (1993), who defines azimuth as the angle between the object velocity vector in the local horizontal plane with respect to the meridian, and elevation as the angle between the object velocity vector and the local horizontal plane. The flux magnitude of each debris intersection in the cell is simply the spatial density multiplied by the velocity magnitude (and a conversion factor to change the flux units to 1/m^2/year). The flux vector of each cell intersection is defined by the spatial density, velocity magnitude, azimuth, and elevation values. To avoid storing every flux vector of every cell intersection of every debris orbit, IDES uses a technique to calculate the mean flux vector in each cell from all intersecting debris objects (which are binned according to their mass). In practice, 8 mean flux vectors per cell are required to represent fully the variation in directionality, i.e. 4 sectors in azimuth angle (-180 to -90, -90 to 0, 0 to +90, and +90 to +180), and 2 sectors in elevation angle (-90 to 0, and 0 to +90). With all cells treated in this manner, the result is a snapshot of the entire LEO debris flux environment.


mission life
Load mission data
coordinates, eqns. of planes
Load 3D body geometry
~~flux, velocity,~~ azimuth, elevation, anomaly, mass
Load IDES flux data
Calculate Flux vs Azimuth vs Elevation distributions for each mass bin
Generate random no. particles on cross section disk at each orbit step
Use distributions to randomly generate mass, velocity, azimuth, elevation of each particle
Randomly locate particle on cross-section disk (x1, y1, z1)
Rotate disk through azimuth angle and then elev angle
Calculate normal vector, and then intersection of vector with body geometry planes
Save impactors data
~~body impact~~ coords., trajectory vector
Next IDES file
Next anomaly step
Next mass bin
Next particle

$$P = \frac{\lambda^x}{x!} e^{-\lambda}, \quad \lambda = FAt$$

$$R_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_x) & \sin(\alpha_x) & 0 \\ 0 & -\sin(\alpha_x) & \cos(\alpha_x) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_y = \begin{pmatrix} \cos(\alpha_y) & 0 & -\sin(\alpha_y) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\alpha_y) & 0 & \cos(\alpha_y) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_z = \begin{pmatrix} \cos(\alpha_z) & \sin(\alpha_z) & 0 & 0 \\ -\sin(\alpha_z) & \cos(\alpha_z) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathrm{b} = (x_1\mathrm{i} + y_1\mathrm{j} + z_1\mathrm{k}) + t(\kappa_1\mathrm{i} + \lambda_1\mathrm{j} + \mu_1\mathrm{k})$$

$$\mathrm{b}.(\kappa_2\mathrm{i} + \lambda_2\mathrm{j} + \mu_2\mathrm{k}) = c$$

Figure 5-5 Flow diagram of impactor ray-trace technique

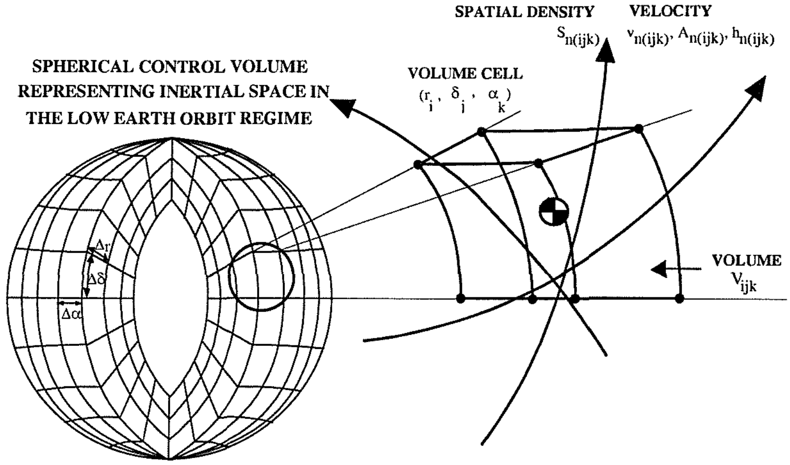

SPATIAL DENSITY
VELOCITY
S n(ijk)
v n(ijk), A n(ijk), h n(ijk)
SPHERICAL CONTROL VOLUME REPRESENTING INERTIAL SPACE IN THE LOW EARTH ORBIT REGIME
VOLUME CELL
(r i, δ j, α k)
Δr
Δδ
Δα
VOLUME
V ijk

Figure 5-6 Inertial control volume representation of LEO debris flux environment [Walker (2000)]

Now, to consider the debris flux relative to a target satellite orbiting through the LEO environment, Walker (2000) has implemented the following target-centred approach in IDES. First, the cell intersections of the target orbit are found in the same way as for the above debris flux calculations. For each cell intersection, the target's velocity components in geocentric cartesian coordinates ($\dot{x}_m, \dot{y}_m, \dot{z}_m$) are calculated together with its declination, δ_m, right ascension, α_m, and residential probability. Then, for each mass bin of the intersected cell, the velocity components ($\Delta\dot{x}_{n.m}$, $\Delta\dot{y}_{n.m}$, $\Delta\dot{z}_{n.m}$) of each of the 8 mean debris flux vectors relative to the target are derived from:

$$\begin{pmatrix} \Delta\dot{x}_{n.m} \\ \Delta\dot{y}_{n.m} \\ \Delta\dot{z}_{n.m} \end{pmatrix} = v_n \begin{pmatrix} c\alpha_m\left(-s\delta_m ch_n cA_n + c\delta_m sh_n\right) - s\alpha_m sA_n \\ s\alpha_m\left(-s\delta_m ch_n cA_n + c\delta_m sh_n\right) + c\alpha_m sA_n \\ c\delta_m ch_n cA_n + s\delta_m sh_n \end{pmatrix} - \begin{pmatrix} \dot{x}_m \\ \dot{y}_m \\ \dot{z}_m \end{pmatrix} \qquad [5\text{-}22]$$

where $c = cos$, $s = sin$, subscript n denotes the debris flux vector (from a particular mass bin), and subscript m denotes the target. A_n, h_n, and v_n are the azimuth, elevation, and velocity magnitude values of the mean flux vector. In order to express the relative velocity vector in the target-centred moving reference frame, the target right ascension of ascending node, Ω, inclination, i, and argument of true latitude, u_m, (the addition of argument of perigee and true anomaly at mid-cell intersection) are used in a matrix transformation given by:

$$\bar{v}_{n.m} = \begin{pmatrix} v^u_{n.m} \\ v^v_{n.m} \\ v^w_{n.m} \end{pmatrix} = \begin{pmatrix} cu_m c\Omega - su_m s\Omega ci & cu_m s\Omega + su_m c\Omega ci & su_m si \\ -su_m c\Omega - cu_m s\Omega ci & -su_m s\Omega + cu_m c\Omega ci & cu_m si \\ s\Omega si & -c\Omega si & ci \end{pmatrix} \begin{pmatrix} \Delta\dot{x}_{n.m} \\ \Delta\dot{y}_{n.m} \\ \Delta\dot{z}_{n.m} \end{pmatrix}$$

[5-23]

The relative velocity vector, $\bar{v}_{n.m}$, has three orthogonal components representing relative velocity for the directions u, v, and w in the target-centred reference frame, as shown in Figure 5-7. The u direction denotes the radial direction, the v direction denotes the along-track direction, and the w direction denotes the out-of-plane direction. The relative velocity magnitude between a mean debris flux vector and the target is then simply:

$$v_{rel} = \left|\bar{v}_{n.m}\right| = \sqrt{\left(v^u_{n.m}\right)^2 + \left(v^v_{n.m}\right)^2 + \left(v^w_{n.m}\right)^2} \qquad [5\text{-}24]$$

The relative flux magnitude between the mean debris flux vector and target is given by the debris vector's spatial density multiplied by this relative velocity (with a conversion factor to express the correct flux units of 1/m^2/year). Finally, the target-centred encounter angles of azimuth and elevation are determined by:

$$A_{rel} = \tan^{-1}\left(\frac{v^w_{n.m}}{v^v_{n.m}}\right) \qquad [5\text{-}25]$$

and

$$h_{rel} = \sin^{-1}\left(\frac{v^u_{n.m}}{v_{rel}}\right) \qquad [5\text{-}26]$$

Figure 5-7 shows these angles in relation to the reference frame.

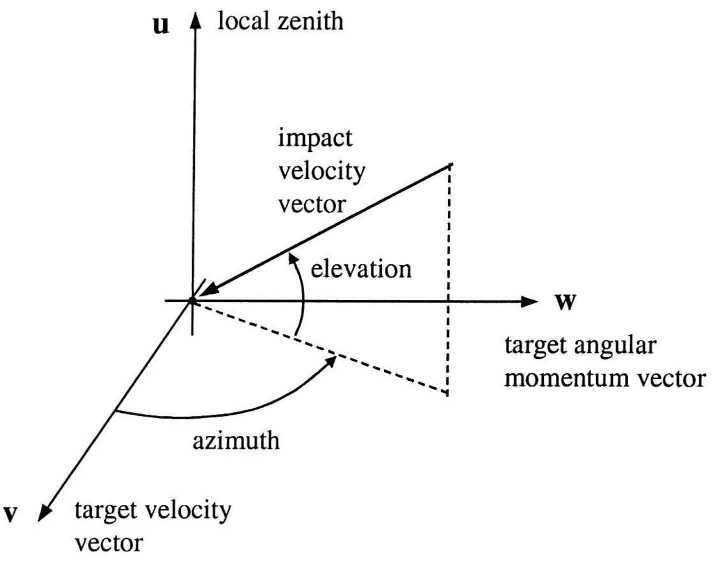


Figure 5-7 Target-centred moving reference frame for the determination of directional debris impact flux relative to an arbitrary target orbit [Walker (2000)]

The contributions of relative flux magnitude, from each mean debris flux vector encountered, are used to derive distributions of relative flux over encounter velocity, azimuth, elevation, and target true anomaly for debris larger than a user defined threshold. These distributions are output as a file with the format shown in Table 5-5. For illustration purposes, example values are used in the table. Typically, IDES partitions the debris fluxes into 13 distinct mass bins. The number of true anomaly steps, into which the target orbit is subdivided, depends upon the number of volume cell transitions of the orbit.

True anom. (degrees)	Debris mass bin	Vector No.	Flux ($1/m^2/year$)	Velocity (km/s)	Azimuth (degrees)	Elevation (degrees)
1.329	1	1	0.11701E-02	14.276	21.485	0.164
		2	0.25252E-02	9.707	-50.830	0.317
		etc…..	…..	…..	…..	…..
		8	0.24775E-02	10.224	48.158	-0.181
	Etc…..					
	13	1	0.52128E-07	13.547	27.528	0.095
		2	0.00000E-00	7.627	0.000	0.000
		etc…..	…..	…..	…..	…..
		8	0.97178E-07	9.754	51.996	-0.833
8.067	Etc…..					
Etc…..						
355.576	Etc…..					

Table 5-5 Example IDES output file of target-centred directional debris fluxes

5.5.2 Determination of Target-Centred Meteoroid Collision Flux

The above target-centred collision flux approach can also be repeated for meteoroids using the relationship developed by Grün *et al* (1985) to express the interplanetary meteoroid flux, F_{ip}, as a function of mass, m. In order to derive the sporadic meteoroid flux, F_{spo}, to a stationary target at LEO altitudes, the interplanetary flux must be multiplied by the Earth focusing factor, G_E, and the Earth shielding factor, ζ. The required equations for these computations are shown below:

$$F_{ip} = c_0\left(\left(c_1 m^{0.306} + c_2\right)^{-4.38} + c_3\left(m + c_4 m^2 + c_5 m^4\right)^{-0.36} + c_6\left(m + c_7 m^2\right)^{-0.85}\right) \quad [5\text{-}27]$$

where

$c_0 = 3.15576 \times 10^7$ $\quad c_1 = 2.2 \times 10^3$ $\quad c_2 = 15$

$c_3 = 1.3 \times 10^{-9}$ $\quad c_4 = 10^{11}$ $\quad c_5 = 10^{27}$

$c_6 = 1.3 \times 10^{-16}$ $\quad c_7 = 10^6$

$$G_E = 1 + \frac{R}{R + H_{alt}} \quad [5\text{-}28]$$

$$\zeta = \frac{1 + \cos\Theta}{2} \quad [5\text{-}29]$$

where

$$\sin\Theta = \frac{R + 100}{R + 100 + H_{alt}} \quad [5\text{-}30]$$

$$F_{spo} = F_{ip} G_E \zeta \quad [5\text{-}31]$$

Here, F_{ip}, is the interplanetary meteoroid flux at 1 Astronomical Unit (AU) in units of 1/m^2/s, R is the mean Earth radius (km), and H_{alt} is the target orbit altitude (km).

To compute the sporadic meteoroid flux relative to a moving target satellite at steps of true anomaly around its orbit, the above stationary flux must be scaled by a k correction factor. This factor is determined using the target velocity, v_t, and an average sporadic meteoroid velocity, v_m, to account for the bias in directionality introduced in the velocity-vector direction by the target's motion. It gives the mean relative impact velocity at each true anomaly step after numerical integration of the impact angle, α_{imp}, over 2π (due to isotropic sporadic meteoroid velocity vectors). The k correction factor is calculated by:

$$v^* = \sqrt{v_m^2 - v_t^2 \sin^2 \alpha_{imp}} \qquad [5\text{-}32]$$

$$v_{imp} = v_t \cos\alpha_{imp} + v^* \qquad [5\text{-}33]$$

$$k = \frac{v_{imp}^3}{v_m^2 v^*} \qquad [5\text{-}34]$$

Although IDES is purely a debris environment model, and therefore does not incorporate the above Grün calculations, other models such as MASTER do. Thus the desired directional meteoroid fluxes can be generated for use by tools such as SHIELD.

5.5.3 Generation of Test Particles

Both IDES and MASTER provide the required target-centred directional flux data, from which it is possible to generate test particles and identify the impact distributions on a satellite. Since the emphasis of this research is debris protection, an interface was developed in SHIELD to read data only from IDES. However, beyond the scope of the Ph.D., it is the author's intention to develop an interface with MASTER.

In practice, it is necessary for SHIELD to access more than one IDES output file to accurately determine the impact distributions on a target satellite over its entire mission life. This is because:

- The satellite's orbit does not remain fixed relative to the background debris environment, i.e. the orbit evolves over time.
- The debris environment is highly dynamic thus causing significant variations to the debris fluxes experienced by a satellite over its mission life.
- The debris environment in the long-term is predicted to exhibit an exponential growth trend (assuming no mitigation measures are introduced).

So, the IDES target-centred flux analyses must be generated at regular epochs throughout the satellite's mission life. Typically, one-month intervals are used, thus giving a total of 120 files for a ten-year mission. SHIELD accesses these files one at a time. That is, all flux data is extracted from a file to calculate the impacts on the target satellite during that one-month interval. Only when the computation is complete does the software access the next file, and so on.

The process of generating debris test particles from the flux data in an IDES file follows a number of steps:

Test Particle Generator - Step 1

First, for a particular mass bin, SHIELD extracts the fluxes and velocities at each interval of true anomaly to produce the following orbit-integrated frequency distributions:

1. Total flux-versus-azimuth, i.e. the summation of all fluxes lying within each 5-degree azimuth bin.
2. Total flux-versus-azimuth-versus-elevation, i.e. the summation of all fluxes lying within each 5-degree azimuth-elevation bin.
3. Velocity distribution-versus-azimuth-versus-elevation, i.e. the distribution of all velocities lying within each 5-degree azimuth-elevation bin.

Next, the flux-versus-azimuth distribution is converted into a cumulative flux-versus-azimuth distribution, which is normalised by the total cumulative value. All of these distributions are then stored for later use.

Test Particle Generator - Step 2

In this step, the software returns to the IDES file and calculates the total flux, F, on the target satellite (for a given mass bin) at a particular true anomaly in the orbit. Next, a disk of known cross-sectional area, A, is constructed that completely bounds the satellite. Given that the time interval, Δt, (e.g. one month) is also known, it is then a straightforward matter to calculate the expected mean number of impacts, λ, on the disk at that point in the orbit, as follows:

$$\lambda = F\,A\,\Delta t \qquad [5\text{-}35]$$

Knowing λ, the predicted number of impacts on the disk can be statistically derived using the Poisson probability density distribution from $x = 0$ to X event probabilities, P_x, where

$$P_x = \frac{\lambda^x}{x!} e^{-\lambda} \qquad [5\text{-}36]$$

Values of P_x are accumulated to give an overall probability, P_X, and then each event probability is normalised by this value. All normalised event probabilities lie in the range from 0 to 1. A Random Number Generator (RNG) then creates a number between 0 and 1. The predicted number of impacts corresponds to the maximum normalised event probability that has been exceeded by this random number.

It is worth mentioning that the Poisson distribution is used here because of its convenience in deriving impact probabilities. The function has also been accepted throughout the community as the *de facto* standard for orbital impact calculations, particularly for situations involving small probabilities (i.e. small values of λ). However, its applicability to large impact probability situations, such as a very large spacecraft with a long mission life in a high flux orbit, may be questionable. Maclay *et al* (1996) note that the Poisson distribution is an empirical model rather than a physical model. Its ability to characterise physical systems is explained in terms of one of two attributes, namely (1) coincidence that the mathematical form of the Poisson curve fits a particular data set and (2) its similarity to the binomial distribution. Unlike the Poisson distribution, the binomial distribution is a physical model that directly describes the behaviour of a well-characterised system as the compound probability of individual trials. The debris impact problem is an example of an inherently binomial distribution. In view of this, one might expect the binomial to be the natural choice for impact simulation software. However, it is too cumbersome compared to the Poisson empirical approach to be implemented in practise here.

The binomial is defined as the distribution of the random variable X, representing the number of successes in n independent trials, each with a success probability p and a failure probability $q = 1 - p$. Expressed mathematically, it is

$$b = \frac{n!}{x!(n-x)!} p^x q^{n-x} \qquad [5\text{-}37]$$

An important property that ties the binomial and Poisson distributions together is that if $\lambda = np$ is held constant, the binomial approaches the Poisson in the limit of large n (or small p). Because of this convergence, the Poisson distribution can under the right circumstances be used as an approximation to the binomial. As noted above, this is when the value of λ is sufficiently small, typically no more than 5 or 10. A quick calculation shows that this limit can be satisfied quite easily for typical unmanned spacecraft. For example, the collision flux of all debris particles > 0.1mm in size in the ISS orbit is of the order 7 $m^{-2}yr^{-1}$ (cf. Chapter 6). Assuming the IDES model divides this orbit into 50 true anomaly steps, then at any given interval the flux F is 0.14 $m^{-2}yr^{-1}$ (on average). If the bounding cross-sectional disk has an area A of 100 m^2, and the time interval Δt is one month, then a typical value of λ is of the order 1. Therefore, for the kinds of risk assessment studies that will be performed using SHIELD, the Poisson distribution is quite adequate.

Test Particle Generator - Step 3

If one or more particles are predicted to impact the disk, then the direction and velocity of these particles is determined by sampling the distributions in Step 1. The azimuth angle of a particle is determined by sampling the normalised cumulative flux-versus-azimuth distribution. Again, this requires the generation of a random number between 0 and 1. Next, the flux-versus-elevation values corresponding to this azimuth value on the flux-versus-azimuth-versus-elevation distribution are extracted. A normalised cumulative flux-versus-elevation distribution is constructed, which is sampled with a random number to derive the desired elevation angle of the particle. Finally, knowing the azimuth and elevation, the particle's velocity is obtained by randomly sampling the velocity distribution-versus-azimuth-versus-elevation.

Steps 2 and 3 must be repeated for each true anomaly interval to derive the total number of particles (within a given mass bin) impacting the bounding disk over the entire orbit. The three-step process must of course be repeated several times to characterise the impacts from particles in the other mass bins. Once complete, the software starts the whole process again with the next IDES file in the series.

5.5.4 Intersection of Test Particles with Satellite Geometry

Each of the generated test particles can now be evaluated to determine whether they will impact the target satellite geometry, and if so where. The software has already determined above that each particle has impacted somewhere on a bounding cross-section disk that intersects the centre of the satellite (which is also the origin of the satellite coordinate system). Therefore, for a given particle, the next step is to randomly choose a point on the disk representing its precise impact location. For this, the disk is assumed to be orientated with its plane normal to the satellite velocity vector direction, thus enabling the exact x, y, z coordinates of the impact point on the disk to be easily calculated. The disk is then rotated twice, first by the particle's elevation angle and then by its azimuth angle, so that the plane of the disk is normal to the particle's direction of impact. In doing so, the impact point on the disk is also rotated twice so that it now has new x, y, z coordinates. Standard rotation matrices are used for this operation, as previously identified in Equations 5-19 to 5-21.

With the new coordinates of the disk impact point calculated, and knowing the disk normal vector (derived from the vector equation of the disk plane), the vector equation of the particle's trajectory can be defined. For example, if the disk normal vector is $(\kappa_1\mathbf{i}+\lambda_1\mathbf{j}+\mu_1\mathbf{k})$ and the impact point is $A(x_1, y_1, z_1)$, then the equation of the particle trajectory is

$$\mathbf{r}=(x_1\mathbf{i}+y_1\mathbf{j}+z_1\mathbf{k})+t(\kappa_1\mathbf{i}+\lambda_1\mathbf{j}+\mu_1\mathbf{k}) \qquad [5\text{-}38]$$

where $\mathbf{r}$ is the position vector, and t is a parameter (not to be confused with time).

The next part of the calculation is essentially a ray-trace method, where the point of intersection of the particle trajectory vector is calculated for each of the satellite's surface planes. For example, if a point $B(x_2, y_2, z_2)$ represents the intersection of the trajectory vector and one of the surface planes, its position vector $\mathbf{b}$ must satisfy the following two equations:

$$\mathbf{b}.(\kappa_2\mathbf{i}+\lambda_2\mathbf{j}+\mu_2\mathbf{k})=c \qquad [5\text{-}39]$$

$$\mathbf{b}=(x_1\mathbf{i}+y_1\mathbf{j}+z_1\mathbf{k})+t(\kappa_1\mathbf{i}+\lambda_1\mathbf{j}+\mu_1\mathbf{k}) \qquad [5\text{-}40]$$

where $(\kappa_2\mathbf{i}+\lambda_2\mathbf{j}+\mu_2\mathbf{k})$ is the normal vector of the surface plane. Taking the dot product of both sides of Equation 5-40 with this normal vector

$$\mathbf{b}.(\kappa_2\mathbf{i}+\lambda_2\mathbf{j}+\mu_2\mathbf{k})=(x_1\mathbf{i}+y_1\mathbf{j}+z_1\mathbf{k}).(\kappa_2\mathbf{i}+\lambda_2\mathbf{j}+\mu_2\mathbf{k})+t(\kappa_1\mathbf{i}+\lambda_1\mathbf{j}+\mu_1\mathbf{k}).(\kappa_2\mathbf{i}+\lambda_2\mathbf{j}+\mu_2\mathbf{k}) \quad [5\text{-}41]$$

From Equation 5-39, and evaluating the dot products, then

$$c=(x_1\kappa_2+y_1\lambda_2+z_1\mu_2)+t(\kappa_1\kappa_2+\lambda_1\lambda_2+\mu_1\mu_2) \quad [5\text{-}42]$$

which gives a value for *t*. Therefore, this can be substituted into Equation 5-40 to give the required position vector of the impact point on the surface plane.

Since trajectory vectors and planes stretch to infinity, it is necessary to perform one final calculation to check if the impact point lies outside the boundaries of the surface. Referring to Figure 5-8, this is determined by simply calculating the vector between the centre of the surface, *C*, and the impact point *B*. If the distance between these points exceeds the distance between *C* and boundary intersection point *D*, then the surface has not been impacted.

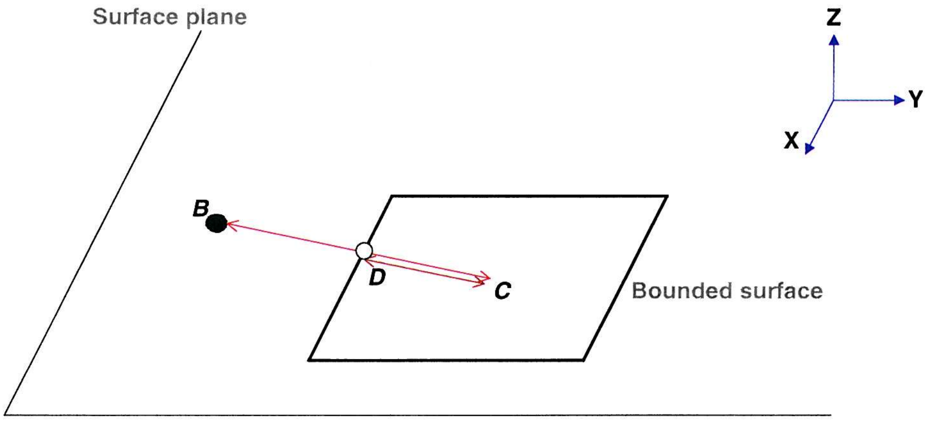


Figure 5-8 Impact point that lies outside the boundaries of a surface

In some cases, the vector may not intersect any part of the satellite. Alternatively, it may intersect one or more surfaces. Figure 5-9 illustrates these examples. If the vector intersects two or more surfaces, the furthest intersection point from the disk is chosen, as this represents the initial point of impact on the satellite. Assuming the particle does impact the satellite, then the impact coordinates are recorded together with various parameters relating to the particle.

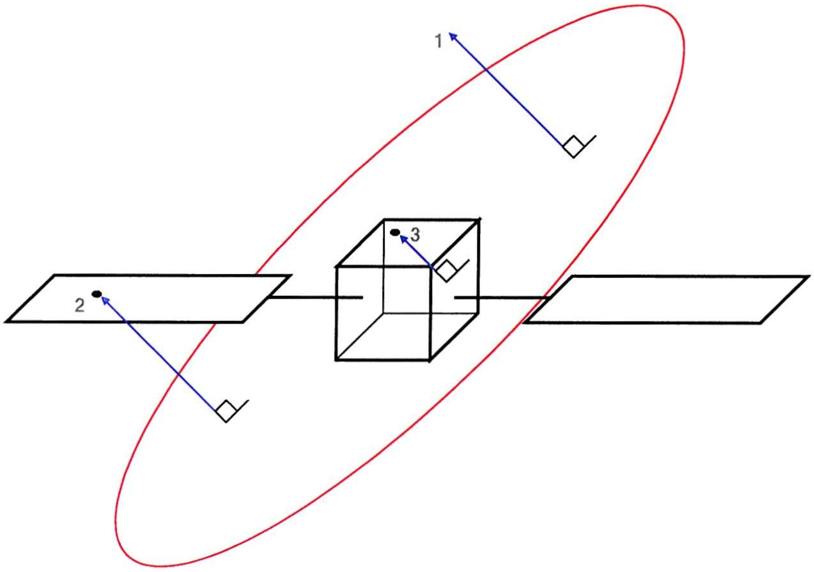


Figure 5-9 Schematic of impactor ray-trace technique

A repeat of this entire process for all the other disk-impacting particles gives the desired impact distribution on the target satellite. Up to this point, the software has a very similar assessment capability to other models such as ESA's ESABASE/DEBRIS and NASA's BUMPER code.

5.6 Evaluation of Satellite Surivability

Previous chapters have already alluded to the desire for characterising the survivability of an unmanned spacecraft in terms of its probability of failure due to debris impact, as opposed to simply the probability of penetration. This is recognition of the possibility that a penetrating particle need not necessarily result in termination of a mission. Therefore, a satellite protection strategy that focuses entirely on adding shielding, rather than considering the spacecraft design as a whole, will tend to be over-engineered and unnecessarily costly. The computation of failure probability is therefore crucial in comparing radically different protection strategies. Not only that, it sets out the debris-induced failure risk in the context of all the other (non-debris related) spacecraft failure mechanisms.

To quantify the probability of failure of a spacecraft design with reasonable accuracy, a technique is required that is not only rigorous but also rapid to execute. Speed and accuracy are essential if a user wishes to compare and quantify many radically different protection design solutions. This is by no means straightforward. To give a simple example of why, Figure 5-10 illustrates three options for protecting a critical unit in a satellite that has a high debris impact flux on one face. In the first option, additional shielding can be placed on the high risk face. In the second, the additional shielding can be applied to the critical unit itself. Finally, in the third option, the critical unit can be placed behind a less critical item. The question the software must answer is - which is the most cost-effective solution?

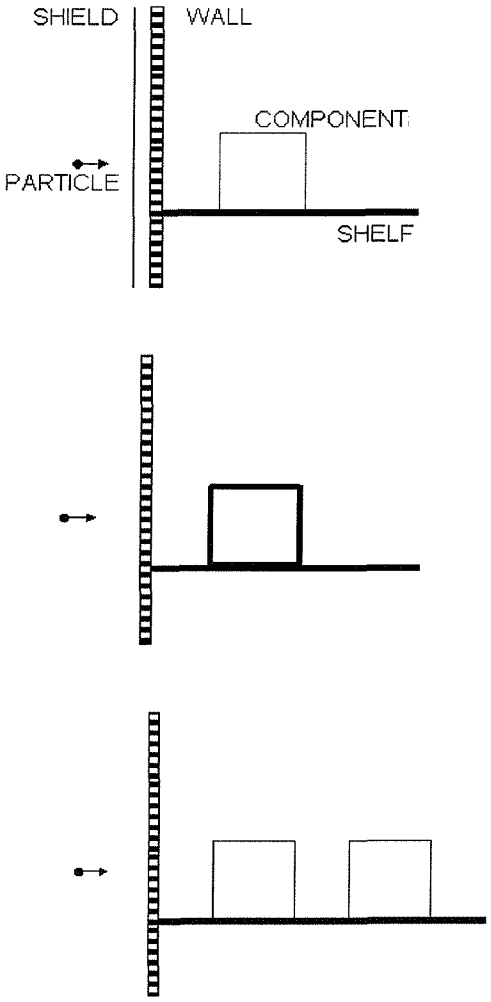


Figure 5-10 Three different options for protecting a critical satellite unit

Even for this simple example the answer is not obvious. It is not difficult to see that the problem becomes that much more complicated when considering a representative spacecraft design. Clearly, the inclusion of a methodology in SHIELD that can address this problem represents a major challenge. The remainder of this section sets out to describe the methodology that has been constructed for version 1.0 of the software.

5.6.1 Simulation of Debris Penetrators

The basis of the failure probability evaluation technique is to generate a large distribution of impacting particles that can penetrate the structure of the satellite body, and any external shielding. This is necessary to build up a statistically meaningful picture of the level of damage to internal equipment, and hence the survivability of the mission. Obviously, such a picture cannot be obtained by looking at just a small number of penetrators in detail.

Performing just one debris impact simulation, as described in Section 5.5, will probably produce few, if any, penetrating impactors on the body during its life. To determine exactly how many penetrate, the ballistic limit of each particle-surface interaction must be evaluated in turn (see flow diagram in Figure 5-11). This is done by calling up an appropriate ballistic limit equation, such as one of those listed in Chapter 3, to calculate whether the diameter of a particular impactor exceeds the critical particle diameter necessary for penetration to occur.

First though, because all of the impacting particles have their sizes expressed in terms of mass (as derived from the IDES data files) rather than diameter, a mass-to-diameter conversion is necessary. NASA has collected data to derive the following expressions [Space Station Freedom Program Office (1991)]:

$$M = 46.81\, d^{2.26} \qquad \text{if } d > 0.0062 \qquad [5\text{-}43]$$

$$M = 2094\, d^{3} \qquad \text{if } d \leq 0.0062 \qquad [5\text{-}44]$$

where M is the mass of a particle in kg, and d is its diameter in metres.

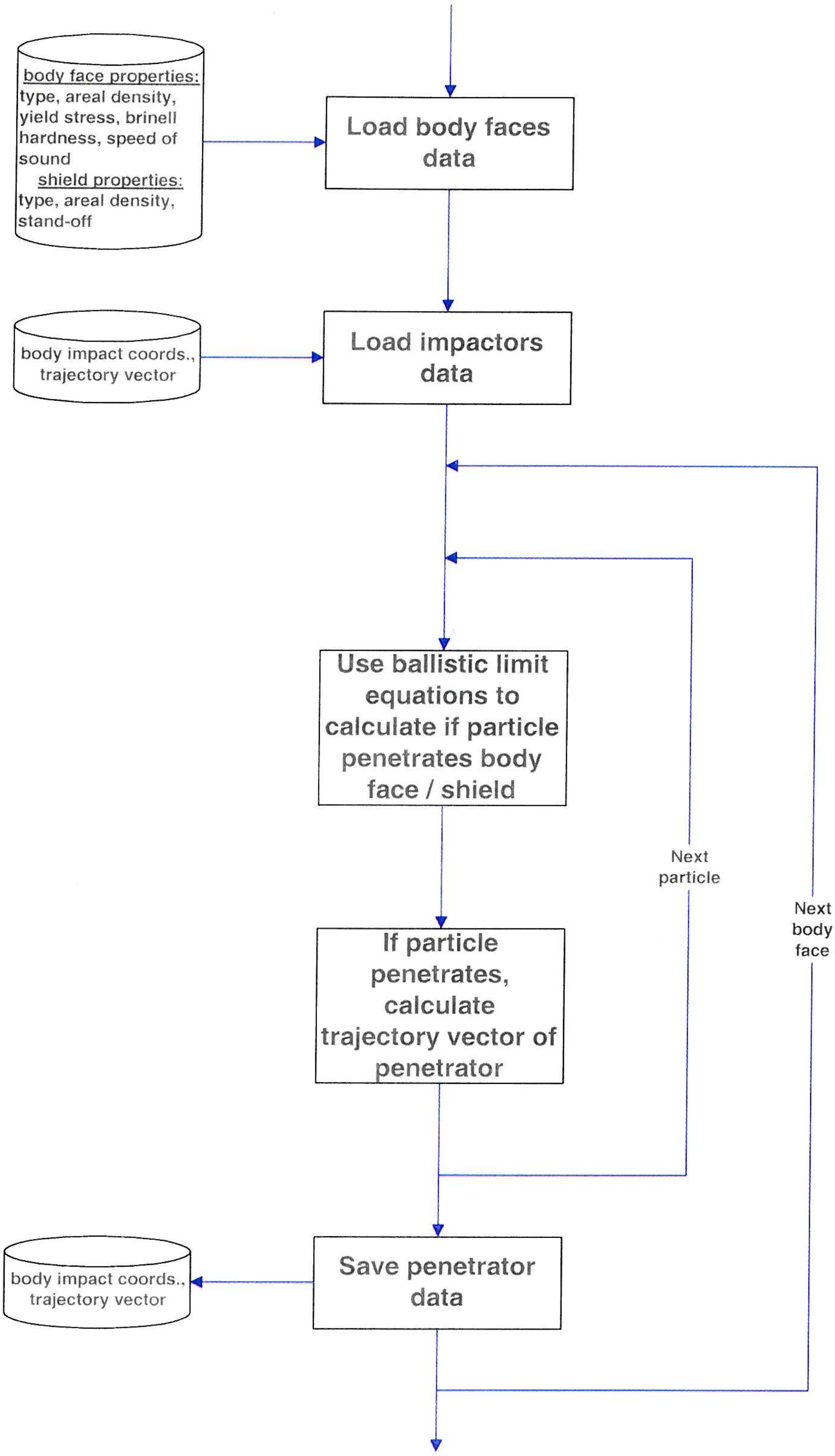

body face properties:
type, areal density, yield stress, brinell hardness, speed of sound
shield properties:
type, areal density, stand-off
Load body faces data
body impact coords., trajectory vector
Load impactors data
Use ballistic limit equations to calculate if particle penetrates body face / shield
Next particle
Next body face
If particle penetrates, calculate trajectory vector of penetrator
body impact coords., trajectory vector
Save penetrator data

Figure 5-11 Flow diagram of penetrator determination process

With the ballistic limit calculations of all particle-surface interactions complete, any particles identified as penetrating are stored for the subsequent internal damage simulations. If, as mentioned above, there are insufficient penetrators from this one lifetime impact simulation, then a larger penetrator distribution can be generated by repeating the impact simulation as many times as required. Typically, several thousand lifetime simulations may be necessary (with each simulation initiated using a different random number seed to compile a statistically significant sample). Effectively, this is a Monte Carlo simulation.

5.6.2 Possibilities for Assessing Internal Damage

Up to this point, the aim of the simulations has been to determine the impact and penetration damage on the main satellite body structure. Now, the focus shifts to determining possible damage inside a spacecraft from the penetrating debris clouds. Before choosing an appropriate assessment methodology for implementation in SHIELD, it was necessary first to explore some of the underlying theory, issues and possible approaches. These are discussed below.

The Impact Process

The behaviour of a solid material subjected to an impulsive loading exhibits one of four responses, namely:

1. Elastic deformation, where the induced stresses are less than the material strength. There are no destructive effects, such as fractures or irreversible dislocations.
2. Plastic deformation, where the induced stresses exceed the material strength. Atoms and molecules are permanently displaced from their original positions, and the result is cratering or perforation.
3. Hydrodynamic, where the material's yield strength is much less than the induced stresses, causing the material to behave like a viscous fluid.
4. Hypervelocity, where the impact pressures far exceed the material's strength, causing it to exhibit hydrodynamic flow and behave like a compressible fluid.

For typical spacecraft structure materials such as aluminium, the hypervelocity phenomenon occurs at impact velocities of ~5 - 7 km/s. Since the relative velocities of objects in LEO typically exceed these figures, then most impacts between LEO satellites and debris will occur at hypervelocities. Therefore, it is particularly important to understand the effects of this type of impact on a spacecraft.

A hypervelocity particle impact on a spacecraft structural material of semi-infinite thickness (i.e. one that is thick relative to the particle diameter) will produce extensive plastic deformation, melting, vaporisation, ionisation, and light flash. As noted in Section 3.4.6, a crater is formed, but no perforation occurs. The need for additional shielding is therefore not an issue.

However, a hypervelocity particle impacting a thin spacecraft structure may produce spallation from the rear side, or perforate completely, thus potentially damaging sensitive equipment behind the structure. It is clear therefore that it is important to understand this penetration process and the resulting debris cloud formation.

At the instant of impact, the projectile and target are shocked into a fluid-like state. Shock waves S_1 and S_2 are generated at the contact surface, and propagate into the projectile and target respectively, resulting in a state of compression (see Figure 5-12a). At the same time, rarefaction waves, R_1, from the free boundaries propagate into the compressed region, thus changing the stress state from compressive to tensile. As a consequence, some fragments are immediately ejected from the tension zone.

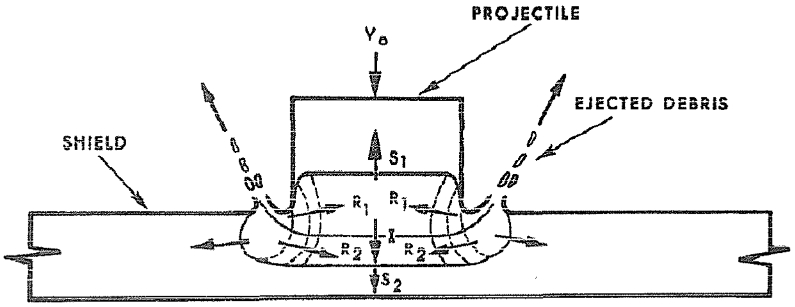


Figure 5-12a Thin target hypervelocity impact immediately after impact [Maiden (1963)]

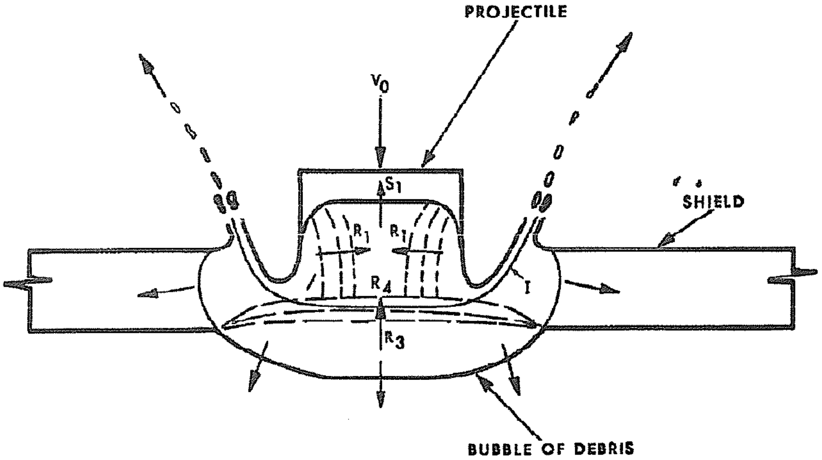


Figure 5-12b Thin target hypervelocity impact after the S_2 shock wave reflects from the free surface [Maiden (1963)]

If the target is thin relative to the projectile diameter, the shock wave S_2 is reflected as a rarefaction wave, R_3, from the back surface of the target, as shown in Figure 5-12b. When R_3 reaches the interface between the projectile and target, a reflection and transmission of the wave takes place, resulting in the transmission of a new rarefaction wave, R_4, into the projectile. With such large amounts of tensile energy imparted into the material, catastrophic fragmentation can take place in these regions. As R_4 continues to propagate into the projectile, depending upon the dimension of the projectile and its material properties, one of the following three situations can occur:

1. Shock wave S_1 is reflected as a rarefaction wave, R_5, from the projectile free boundary. The interaction of R_4 and R_5 puts the entire region in a state of tension, causing the projectile and target to shatter into a cloud of fragments.
2. R_4 catches up with S_1 creating an attenuating wave. The diminishing strength of this wave may reduce the extent of fragmentation, or produce fragments of larger size.
3. The R_1 waves meet at the axis of symmetry of the projectile, thereby unloading it, and causing the projectile to continue its motion as a rigid body.

Observations show that a debris cloud, which has formed behind a thin target, appears as a growing, hollow bubble, with most of the projectile and target material mass concentrated in a thin shell at the front of the bubble surface, close to the trajectory axis. The shape of the debris cloud's shock front, and hence its potential to cause damage, is influenced by the shape of the impacting projectile. A spherical projectile produces a hemispherical shock front, whereas the shock front from a cylindrical impactor is planar near the axis. The dispersion of the debris cloud is greater for a spherical impactor.

The size of the fragments and the state of the debris cloud are dependent on the shock intensity. Fragments diminish in size and increase in number as impact velocity increases, or target thickness decreases. The state of the cloud progressively changes from large, solid fragments to molten particles, and then to a fine vaporous spray, as impact velocity is increased. In the case of a spacecraft structure that has been perforated, solid fragments will have the greatest potential for penetrating deep inside the spacecraft. On the other hand, although a fine spray will not penetrate as far, its dispersion may make it more likely to encounter a larger number of sensitive spacecraft components, such as electrical harnessing, with equally hazardous consequences.

It is not difficult to see that the damage potential of a debris cloud will be governed by the stand-off distance between the perforated structure and the underlying substructure (e.g. equipment). At small enough distances, a cloud of solid or molten particles will not have dispersed much, and so the resulting impact craters on a substructure will overlap. This causes synergistic effects between adjacent craters, and leads to increased damage intensity.

Physical Models of a Debris Cloud

Models to characterise the formation and structure of a debris cloud have been developed by a number of researchers, who are concerned with determining the survivability of underlying structures. Generally speaking, the models are derived from physical considerations rather than from empirical data.

Richardson (1969), Swift *et al* (1982), Passman (1987), and Lawrence (1987, 1992) have all used momentum and energy arguments to estimate the characteristics of a cloud. The theories assume that the projectile and target rupture under the high impact pressures to generate a finely divided material that is uniformly distributed in a thin spherical shell (see Figure 5-13). From momentum and energy conservation considerations of the projectile and debris cloud, expressions have been derived for the cloud's centre-of-mass velocity, expansion velocity, and spray half-angle. However, there is a wealth of evidence from experiments and hydrocode simulations to show that cloud fragments are not distributed uniformly in a thin spherical shell, which contradicts the fundamental assumptions of these theories.

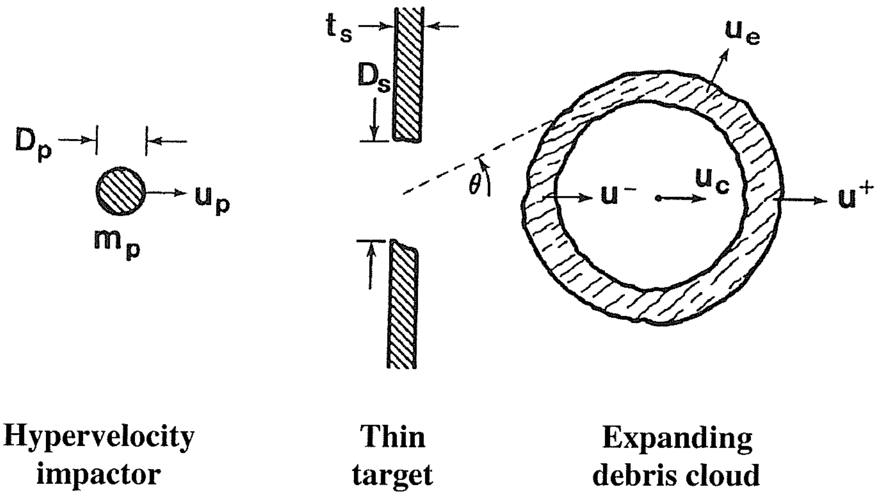


Figure 5-13 Model of debris cloud produced after penetration of a thin target [Lawrence (1992)]

Another theoretically based debris cloud model has been developed by Yew *et al* (1993). The model is based on the one-dimensional shock wave theory developed by Maiden in 1963 (reproduced in Figures 5-12a&b), together with Grady's (1987) catastrophic fragmentation theory. Since the material properties of the projectile and target, as well as their fragmentation characteristics, are taken into consideration, the model may be considered to be an improvement over the above-mentioned spherical expansion model. The derived equations in the model enable both the fragment velocity at the outer perimeter of the cloud and the mean fragment size to be calculated. From these, the momentum and spread of the cloud can then be determined. Results from the model were compared with experimental data obtained by Piekutowski (1990), and showed reasonably good agreement. However, the model is limited, as it cannot provide the exact shape of the cloud or the sizes of fragment particles. For these to be determined, a two-dimensional shock wave analysis would be required together with the equations of state for materials both in compression and in tension.

A first principles based scheme has also been developed by Schonberg (1995), who was concerned primarily with determining the amount of material in each of the three states of matter (i.e. solid, liquid, and gas) in a debris cloud created by a hypervelocity impact on a thin plate. Schonberg starts with an equation of state to calculate the residual energy in the projectile and plate materials upon release from their respective shocked states. Then, elementary thermodynamic principles are used to determine the percentages of shocked and released projectile and plate materials that were melted and/or vaporised during the release process. Assuming certain projectile and plate geometries, these percentages are used to calculate the mass of the projectile and plate materials in solid, liquid, and gaseous form. Finally, the principles of momentum and energy conservation are once again introduced to calculate the angular spread and various debris cloud velocities, as shown in Figure 5-14.

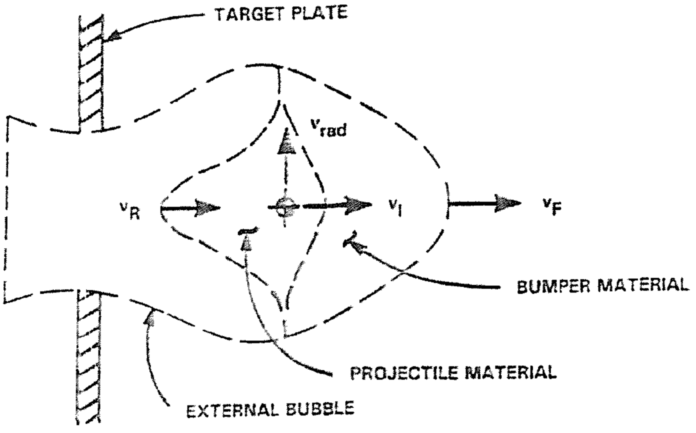


Figure 5-14 Model of debris cloud produced after penetration of a thin target [Schonberg (1995)]

As with the above model by Yew *et al*, Schonberg's approach is not extended to give the mass and velocity of individual cloud fragments. This is because the primary purpose of the model is to examine the impulsive loading on a spacecraft structure following penetration of a bumper plate. In this situation, the separation distance between bumper and structure is usually small enough that a cloud will not have expanded sufficiently to warrant consideration of the effects of individual fragments. Essentially, the cloud is still concentrated enough to be treated as one entity, rather than a collection of discrete particles.

Empirical Model of a Debris Cloud

For the purpose of assessing the interaction of a penetrating debris cloud with the interior of a spacecraft, the above-mentioned models are inadequate. This is for several reasons. First, the models do not consider the cloud resulting from the penetration of a honeycomb sandwich panel, which is a typical structure for an unmanned spacecraft. Second, because the interior of a satellite can be quite spacious, the cloud may expand sufficiently to be considered as a set of individual fragments. Each of these fragments may interact with different items of equipment inside the satellite. The above models are only viable in the case when an internal equipment item is situated reasonably close to the impact point on the satellite structure. Third, there is a possibility the models will be too cumbersome for implementation in SHIELD, since the software is required to evaluate rapidly the interactions of many penetrating clouds inside a spacecraft.

Therefore, the author proposes an entirely different approach, in which an empirical debris cloud model is constructed based on data derived from impact test programmes. The following types of data should be considered in this respect:

- The distribution of impact craters on a witness plate situated behind a target structure.
- The distributions of impact holes on a series of thin foils situated at various distances behind a target structure.
- A series of X-ray flash photographs of a debris cloud as it expands behind a target structure.

Unfortunately, many impact test programmes do not regularly collect such data, partly for reasons of cost and partly because the ballistic limit of a target (the main purpose of many test programmes) can be derived without it. Nevertheless, it is still possible to construct the basic mathematical form of an empirical cloud model in anticipation of such data becoming available in the future. For this, the empirical secondary ejecta model developed by Rival & Mandeville (1999) could be a useful starting point.

Secondary ejecta are particles that are released following an impact of a primary debris particle with a spacecraft surface. The ejecta, which comprises material from the primary particle and the target surface, is back-scattered from the primary impact crater. Most of the material is contained within a cone, such that the total number of fragments, with size between δ and $\delta+d\delta$ ejected with velocity between v and $v+dv$ in the solid angle $d\Omega=\sin\theta\, d\theta d\varphi$ around the θ (zenith), φ (azimuth) direction, is given by:

$$n_{cone}(\delta,\theta,\varphi,v)d\delta\ d\Omega\ dv \qquad [5\text{-}45]$$

Figure 5-15 illustrates the impact configuration, frame of reference, axis, and coordinates for the ejecta model.

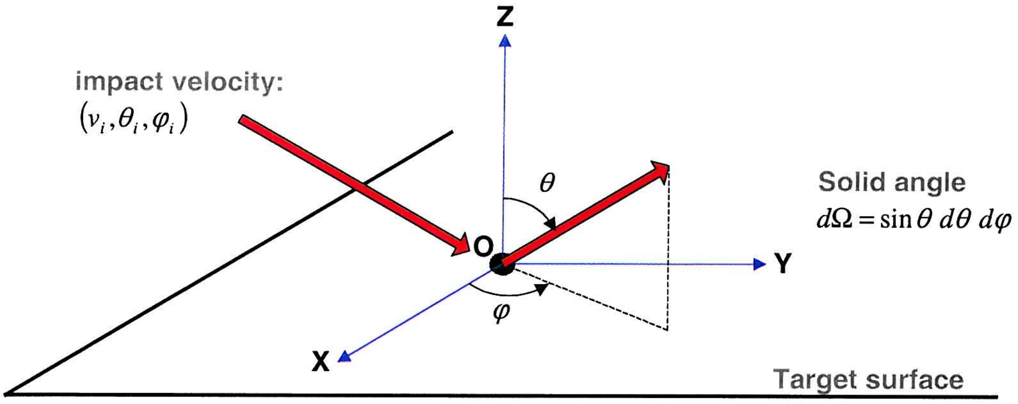

Z
impact velocity:
(v_i,θ_i,φ_i)
θ
Solid angle
$d\Omega = \sin\theta\ d\theta\ d\varphi$
O
Y
φ
X
Target surface

Figure 5-15 Impact configuration, reference frame, axis, and coordinates [Rival & Mandeville (1999)]

Based on the assumption that δ, θ, and φ are independent variables, Rival & Mandeville (1999) define a distribution function for n_{cone} as follows:

$$n_{cone}(\delta,\theta,\varphi,v) = K_{cone} f_{cone}(\delta)\ g_{cone}(\theta)\ h_{cone}(\varphi)\ \Delta(v - v_{cone}(\delta,\theta,\varphi)) \qquad [5\text{-}46]$$

where Δ is the Dirac delta function, and f_{cone}, g_{cone}, and h_{cone} are size, zenith, and azimuth probability density functions, respectively. K_{cone} is a normalisation constant for fitting n_{cone} to the total mass ejected in the cone. And finally, v_{cone} is a velocity function that ensures fragment size and velocity are inversely related.

Given that a cone of ejecta fragments and a cone of penetrating debris fragments have common characteristics, there is no obvious reason why a similar function to that given in Equation 5-46 cannot be applied to the cloud-modelling problem. Once the empirical data has been collected, it should be relatively straightforward to construct the individual probability density functions and implement these in the software code.

It is important to emphasise that the application of this model is limited strictly to the case where a cloud has expanded sufficiently such that the interaction between each of the cloud fragments and the underlying spacecraft equipment can be assessed on an individual basis. In other words, the space between the spacecraft body structure and the internal components is large enough that the cloud fragments do not produce overlapping impact craters or holes on the surfaces of the components. Under these circumstances, a suitable ballistic limit equation can be used on each fragment / component interaction to determine which fragments if any will penetrate the components and potentially cause lethal damage.

For the case where a cloud has not expanded enough, then the cloud must be considered as a single entity with a dispersed mass (as opposed to a concentrated mass such as a particle). Assuming that such a cloud interacts completely with the surface of an underlying component, then it is not necessary to construct a model of the cloud to determine whether the component is penetrated. A suitable multiple wall ballistic limit equation is sufficient to derive the required result. The difficulty arises when there is only partial interaction between the cloud entity and a component, i.e. only some of the cloud hits the component. In this instance, it is not possible to utilise a ballistic limit equation without first knowing what percentage of the cloud mass and energy is impacting the component. Such information can of course be determined by a theoretical treatment of the cloud distribution, as discussed previously. However, to achieve an accurate description of a cloud, it is probably best to derive a model based on empirical data. This would require a comprehensive set of impact test data, possibly supported by hydrocode simulations, which is beyond the scope of the current work.

Setting aside these difficulties, it is perhaps worth looking to see what available evidence there is for the morphology of a post-penetration debris cloud resulting from an impact on a typical unmanned spacecraft structure. This at least may allow a simple cloud model to be incorporated in the software as a first step prior to the implementation of the more comprehensive approaches described above.

Recently, Taylor (1999) has performed a series of hydrocode studies of hypervelocity impacts onto Whipple bumpers and sandwich plates with honeycomb cores. For a 1 cm spaced Whipple bumper, the resulting clouds had velocities ~50% of the initial impact velocity. They also exhibited little radial expansion, and so would not spread out over a large area within a spacecraft, i.e. the damage would be confined within a narrow cone. Additionally, solid fragments of bumper material were present in the clouds, thus increasing the possibility of damage to internal spacecraft equipment. For a 1 cm spaced sandwich plate / honeycomb core configuration, a different morphology was observed, comprising a central hollow cylinder with a debris cloud bubble extending several times the cylinder's radius. This result is shown in Figure 5-16. Taylor observes that the size of the cylinder is most likely driven by the honeycomb cell radius. The large radial expansion rate of the bubble cloud will cause a significantly larger damage area inside a spacecraft than for the Whipple case. Conversely, because the highest density material is concentrated in a small, low velocity hollow cone that does not expand much, this may represent the greatest risk for penetrative damage of internal equipment.

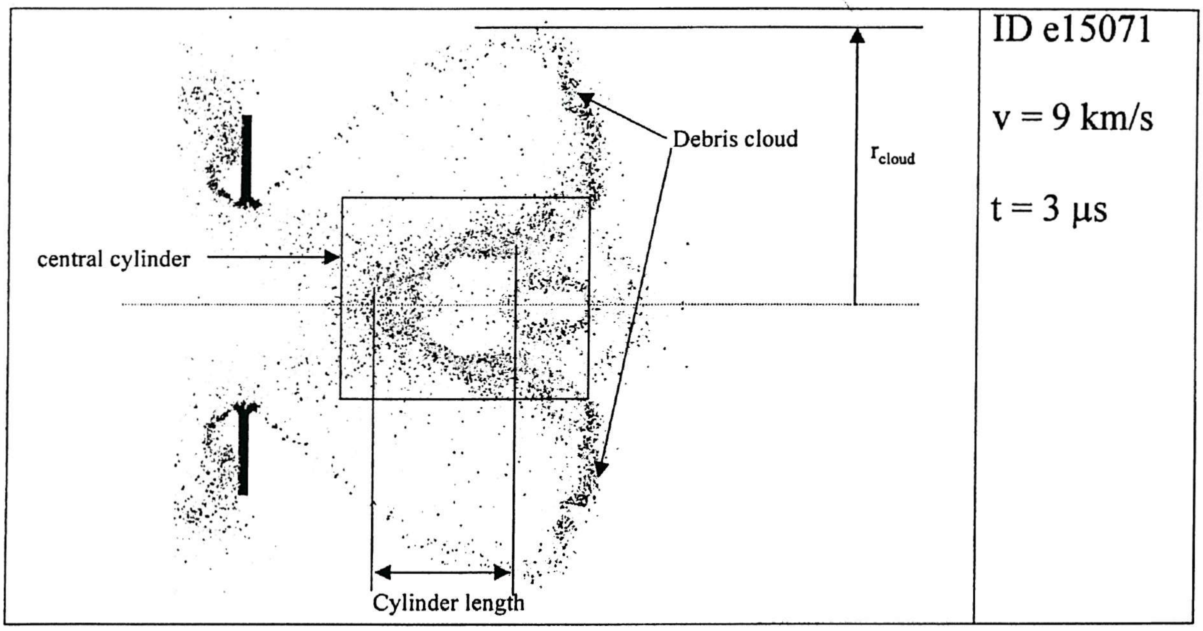


Figure 5-16 Debris cloud produced after penetration of a sandwich panel [Taylor (1999)]

So, these results provide a useful pointer for the construction of a preliminary algorithm to characterise the internal spacecraft damage following penetrations. Specifically, in the first version of the code, it would seem reasonable to start by approximating penetrating debris clouds as concentrated lump masses. This overcomes the immediate problem of the non-availability of the data necessary for developing the desired empirical models. It also has the advantage of being computationally straightforward, since each cloud simulation is simply the interaction of one particle with the internal equipment.

5.6.3 Implemented Methodology for Assessing Internal Damage

Based on the preceding considerations, the following internal damage assessment methodology has been implemented in the first version of SHIELD. Starting with a given satellite design, the software first introduces all of the satellite's internal components, shelves, and walls. The method for this is already described in Section 5.4. Next, the data set of satellite body penetrating particles (calculated in Section 5.6.1) is called up. Using the same vector / plane intersection mathematics summarised in Section 5.5.4, each penetrator is then ray-traced to identify which internal components lie within in its line of sight.

Depending on the internal configuration, it is conceivable that a penetrator's vector could pass through several components. To determine exactly how far inside the satellite the penetrator actually reaches, the model repeatedly calls up one of the multiple wall ballistic limit equations in Chapter 3. Initially, the equation is used to calculate whether the particle penetrates the first component that it hits. In this instance, the body structure is regarded as the bumper element of the equation, and the impacted face of the component is the back-wall in the equation (see Figure 5-17).

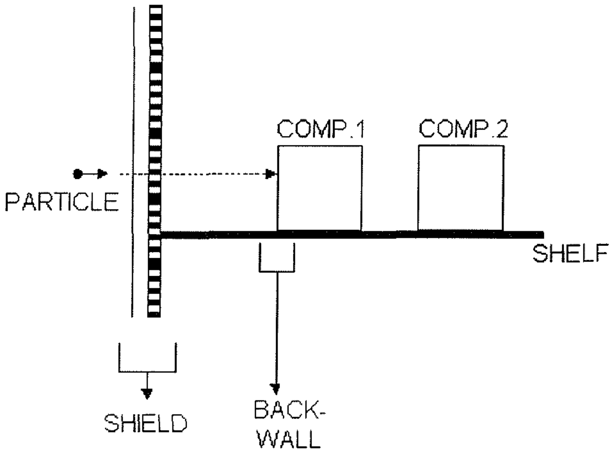


Figure 5-17 Penetration of first component

Because the combination of body structure and component may utilise several different materials, a conversion is necessary before using the multiple wall equation. That is, all materials are converted into an equivalent thickness of aluminium. If the equation predicts a penetration of the first component, then there is a distinct possibility the component will fail. However, it is by no means certain. Depending upon the component's design, function, and operational status (i.e. power on/off), and the impact momentum / energy, a component can be affected in a number of ways, and to varying degrees. Table 3-1 summarised some of the consequences. The key damage criteria to characterise how a component will respond to an impact are:

- ❑ Physical damage, i.e. craters and holes.
- ❑ The generation of plasmas that lead to charging and large transient currents
- ❑ The interaction of shock waves, particularly in equipment with large amounts of stored energy.

Experience from the defence sector reveals that there is a quantifiable probability that a component may survive [Trufitt (1999)]. From extensive impact tests on military hardware, failure probability density functions have been derived for a variety of components. A typical probability profile is shown in Figure 5-18. At the present time, no such data exists for spacecraft hardware, and so similar probability profiles can not yet be included in the SHIELD software. Therefore, an assumption is made that a penetrated component will always fail.

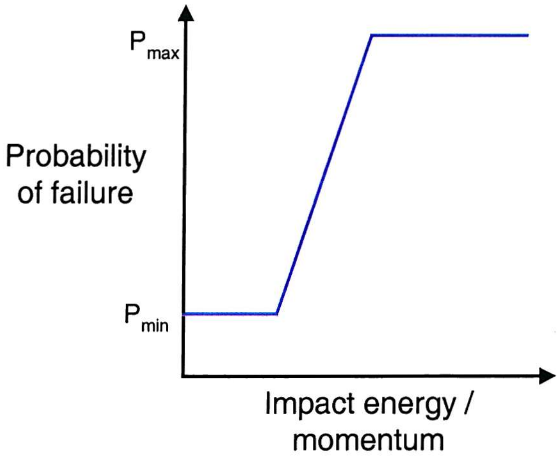


Figure 5-18 Failure probability density function for an impacted component

Assuming a component is predicted to fail, the consequences for the mission must be determined. For example, if the component is not essential to the mission or has a redundant unit, then it is conceivable that the mission may continue largely unaffected. The survival logic inherent in Fault Tree Analysis (FTA) is an invaluable aid in this respect. FTA is a standard technique for analysing the reliability of complex systems. It can predict high-level system responses to one or more low-level component failures, which is exactly what is required for the problem here. However, the development of an FTA module within the software is not a trivial task, and so must be considered beyond the scope of the research. To circumvent this issue, a simpler logic was encoded in which mission success or failure is linked directly to a user-defined component 'criticality' value. For this, the definitions of criticality in Table 4-6 are used. Essentially, if the criticality value is high, then failure of the component will usually lead to a mission failure, whereas failure of a low criticality component will not.

A further factor included in the assessment of mission success or failure, is the extent to which a component is damaged. For example, it is possible that if a penetrator has sufficient energy it could cause complete break-up of the component. This is directly analogous to the break-up of a satellite when it collides with a large enough piece of orbital debris. Although unlikely to occur, if a component experiences such an impact, then the software assumes that the mission always fails. Again, no data for this circumstance exists, so it is based purely on engineering judgement. It is difficult to conceive that numerous, large, high velocity component fragments ejected inside a satellite would not be highly disruptive to the mission. The criterion chosen for determining whether a component will break-up is the ratio of impact energy to target mass. This is the same approach used in satellite break-up predictions. In the absence of any component break-up data, a threshold ratio of 40 J/gram is used, which is a value used in some satellite break-up models [Jonas *et al* (1993)]. So, as an example, a 10 kg component would break-up when impacted by an ~16 gram penetrator with a velocity of 7 km/s. From Equation 5-43, this is equivalent to an ~ 3 cm diameter particle, which is certainly at the higher end of the size range of interest.

Once the first component has been assessed, the model proceeds to consider whether the second component in the line of sight is also vulnerable to penetration. Essentially, the model repeats the above calculations, but now the second component's impacted face is the back-wall in the ballistic limit equation, and the bumper element of the equation comprises both the body structure and the first component (see Figure 5-19).

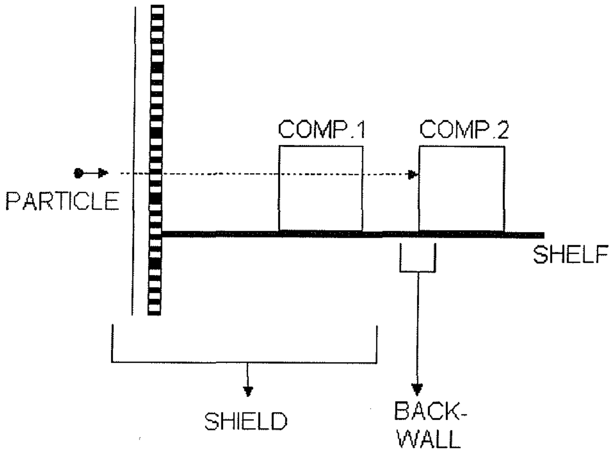


Figure 5-19 Penetration of second component

As before, penetration is calculated, and mission success/failure determined. This iterative process repeats until the penetrator is stopped, or there are no more components to consider.

When all penetrators have been examined in turn to establish their potential for mission damage, the resulting data-set reveals the total number of mission failures out of the 1000+ Monte Carlo lifetime simulations. In other words, it gives the probability of failure, and hence survival, of the satellite in its particular operational environment. This probability is therefore the required measure of survivability of a particular satellite design. A flow diagram summarising the internal damage assessment process is shown in Figure 5-20.

5.7 Derivation of Satellite Survivability Metric

The evaluation of survivability of a satellite, in terms of its probability of failure, is clearly very important particularly when one wishes to quantify and compare possible protection enhancements. However, it is not enough just to assess the probability of failure of a particular design when evaluating its effectiveness. The associated costs in achieving that level of risk must also be considered. In other words, a function is required that encapsulates both risk and cost.

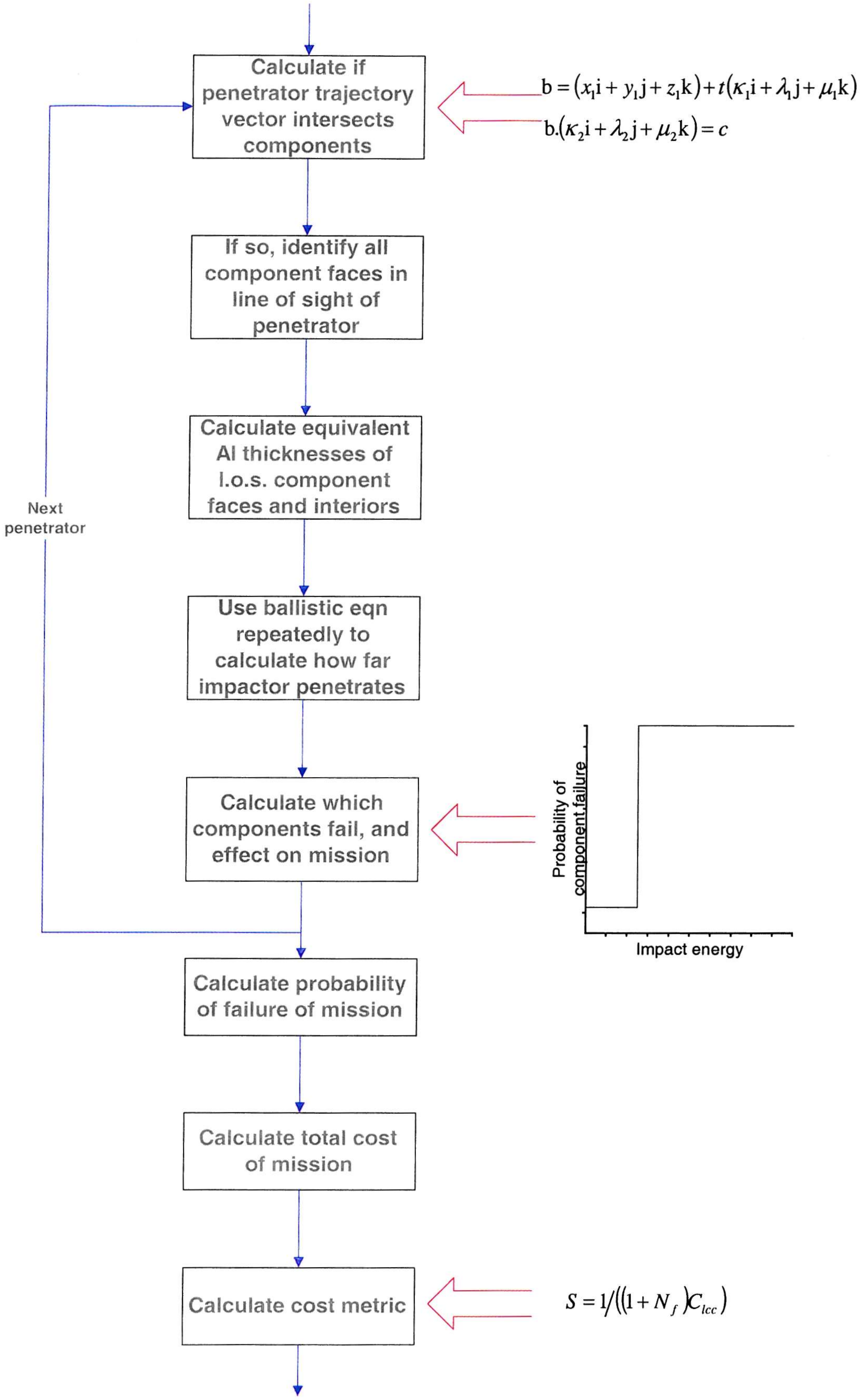

Calculate if penetrator trajectory vector intersects components
$b = (x_1 i + y_1 j + z_1 k) + t(\kappa_1 i + \lambda_1 j + \mu_1 k)$
$b.(\kappa_2 i + \lambda_2 j + \mu_2 k) = c$
If so, identify all component faces in line of sight of penetrator
Calculate equivalent Al thicknesses of l.o.s. component faces and interiors
Next penetrator
Use ballistic eqn repeatedly to calculate how far impactor penetrates
Calculate which components fail, and effect on mission
Probability of component failure
Impact energy
Calculate probability of failure of mission
Calculate total cost of mission
Calculate cost metric
$S = 1/((1 + N_f)C_{lcc})$

Figure 5-20 Flow diagram showing the assessment of internal damage

5.7.1 Cost Model

The standard method for determining the financial cost of a mission is to calculate its Life Cycle Cost (LCC). There are three basic approaches to modelling the LCC of a mission:

1. Detailed bottom-up estimating, where the development and production costs of each low-level element of a system are identified.
2. Analogous estimating, where the cost of a similar item is used, and adjusted for size and complexity.
3. Parametric estimating, where Cost Estimating Relationships (CERs) express the costs as a function of design performance parameters.

Parametric cost models are the most effective method for performing the kinds of trade-off assessments as envisaged for the software. They are relatively quick to implement, provide consistent estimates, and are traceable to the class of systems for which the model is applicable. CERs are readily available in the literature for estimating detailed costs of the space, launch, and ground segments, and the operations and support costs. The LCC is then easily determined by summing these costs. The following simple equations represent the LCC model implemented in SHIELD:

$$C_{sat} = C_{base} + C_{prot} \qquad [5\text{-}47]$$

$$M_{sat} = M_{base} + M_{prot} \qquad [5\text{-}48]$$

$$C_l = \alpha M_{sat}^{\beta} \qquad [5\text{-}49]$$

$$C_{lcc} = C_{sat} + C_l \qquad [5\text{-}50]$$

where,

C_{sat} = total cost to build satellite

C_{base} = baseline cost of satellite, i.e. excluding protection

C_{prot}= cost of implementing protection measures

M_{sat} = total mass of satellite

M_{base} = baseline mass of satellite, i.e. excluding protection

M_{prot} = mass of protection measures

C_l = cost to launch satellite

α, β = factors relating launch cost to mass

C_{lcc} = total cost of a satellite mission, i.e. its life cycle cost

5.7.2 Survivability Metric

To define a simple measure that combines cost and risk, the following example is used. Let us start by assuming that a particular unprotected satellite design has a debris-induced probability of failure, P_f, of 2%. Referring back to the exponential expression in Equation 3-12, the number of satellite failures, N_f, is therefore equal to ~0.02. So, if 100 of these satellites were launched then on average one would expect approximately two satellites to fail. Therefore, two more would have to be launched to replenish the system, i.e. a total of 102 satellites would be required. The total cost of this system of unprotected satellites is then

$$C_{sys,u} = 102C_{lcc,u} \qquad [5\text{-}51]$$

Now, let us suppose the satellite design is changed to include some shielding protection, and as a consequence the probability of failure, P_f, reduces to 1%. Then the cost of the system of protected satellites becomes

$$C_{sys,p} = 101C_{lcc,p} \qquad [5\text{-}52]$$

where $C_{lcc,p} > C_{lcc,u}$. Therefore, a straightforward comparison between $C_{sys,u}$ and $C_{sys,p}$ provides the necessary answer as to which is the most cost-effective solution. Generalising this example, the following 'survivability metric', denoted by S, is proposed

$$S = 1/((1+ N_f)\ C_{lcc}) \qquad [5\text{-}53]$$

This metric is then the required measure to distinguish between competing protection solutions, which is a key element of the software's optimisation function.

5.8 Genetic Algorithm Search Method

Up to this point, the software has been concerned primarily with providing a capability to assess a very specific satellite design and protection strategy. This means that a user wishing to explore variations to the design has to do so on a trial-and-error basis, by manually changing certain parameters and re-running the simulation to quantify the effect. Clearly, this is a reasonable approach if only a limited number of options are to be investigated. However, it quickly becomes problematic when a large 'design space' has to be searched, as is sometimes the case during the early phases of a satellite design. The provision of an automated search technique that can alleviate this burden for the user, whilst converging rapidly on or near an optimum design solution, would be a potentially advantageous feature of the software. Such a technique, i.e. a genetic algorithm, was tested in the prototype model discussed in the previous chapter, and found to be satisfactory for this purpose.

The task here though is more complex, since a number of system design constraints have to be considered simultaneously during the search. For example, if the satellite design has to remain within strict cost and mass budgets, the model must identify how best to employ shielding on the satellite whilst satisfying these prescribed limits. Another constraint concerns arranging internal equipment in such a way that the satellite's overall mass and thermal balance is retained. The internal arrangement may also be constrained because certain items have to be located on particular walls or shelves for various mission reasons. Finally, the need to protect against secondary radiation effects [Dyer *et al* (1996)] can also influence the choice of shielding and location of components.

In this first version of SHIELD, all but the thermal and radiation constraints are incorporated. However, for the longer term, it is the author's intention that these, and possibly other, constraints will also be introduced. At that point, the software would effectively become a concurrent engineering tool.

The remainder of this section describes how the genetic algorithm method has been implemented in SHIELD.

5.8.1 Initial Population

As with the prototype, the genetic algorithm begins by randomly creating a population of satellite design solutions. Each design is unique, and will differ from the other designs according to the values of certain parameters. For example, at the outset the user may have specified the location of one or more components precisely, or alternatively he may allow the software varying degrees of freedom to place it/them inside the satellite. Table 5-6 lists the properties that can currently be set as variable in the optimisation.

Item	Variable
Satellite body:	
structure	thickness or areal density of faces
shelves	position
shields	type thickness or areal density stand-off distance location on body faces
Internal components:	
structure	thickness / areal density of faces
attachment	position on a shelf position on a body face position on an internal wall various permutations of positioning on shelves &/or faces &/or walls orientation (once positioned)

Table 5-6 List of optimisable properties

A user can choose to vary some or all of these as required. Once selected, the user must then preset the extent of variability of the properties, i.e. upper and lower limits must be defined. For example, the user may wish to investigate varying the thickness of a particular component's structural casing between, say, 0.5 mm and 1.5 mm.

The creation, and survivability evaluation, of each satellite design in the initial population follows the sequence set out in Sections 5.4 to 5.7, but with three fundamental differences.

1. A random number generator is called up to fix a value for each variable (within the prescribed upper and lower limits).
2. If the positioning of one or more components inside the satellite body is a variable in the optimisation, then an algorithm is called up to check that there are no spatial clashes, i.e. that a component does not occupy the same space as another component or internal structure. For example, the algorithm checks for component clashes by subdividing the interior of the satellite body into an array of volume cells, or voxels. The voxels are sized to be at least an order of magnitude smaller than a component. When a component is positioned inside the satellite all voxels occupying that space are switched on. Then, if another component is subsequently added that partially or wholly occupies the same space, the presence of voxels that are already switched on causes a clash detection warning to be flagged. The algorithm then proceeds to shuffle the second component away using a 'random walk' process until the clash is no longer flagged up. A flow diagram of the entire process for constructing a satellite in the initial population is shown in Figure 5-21.

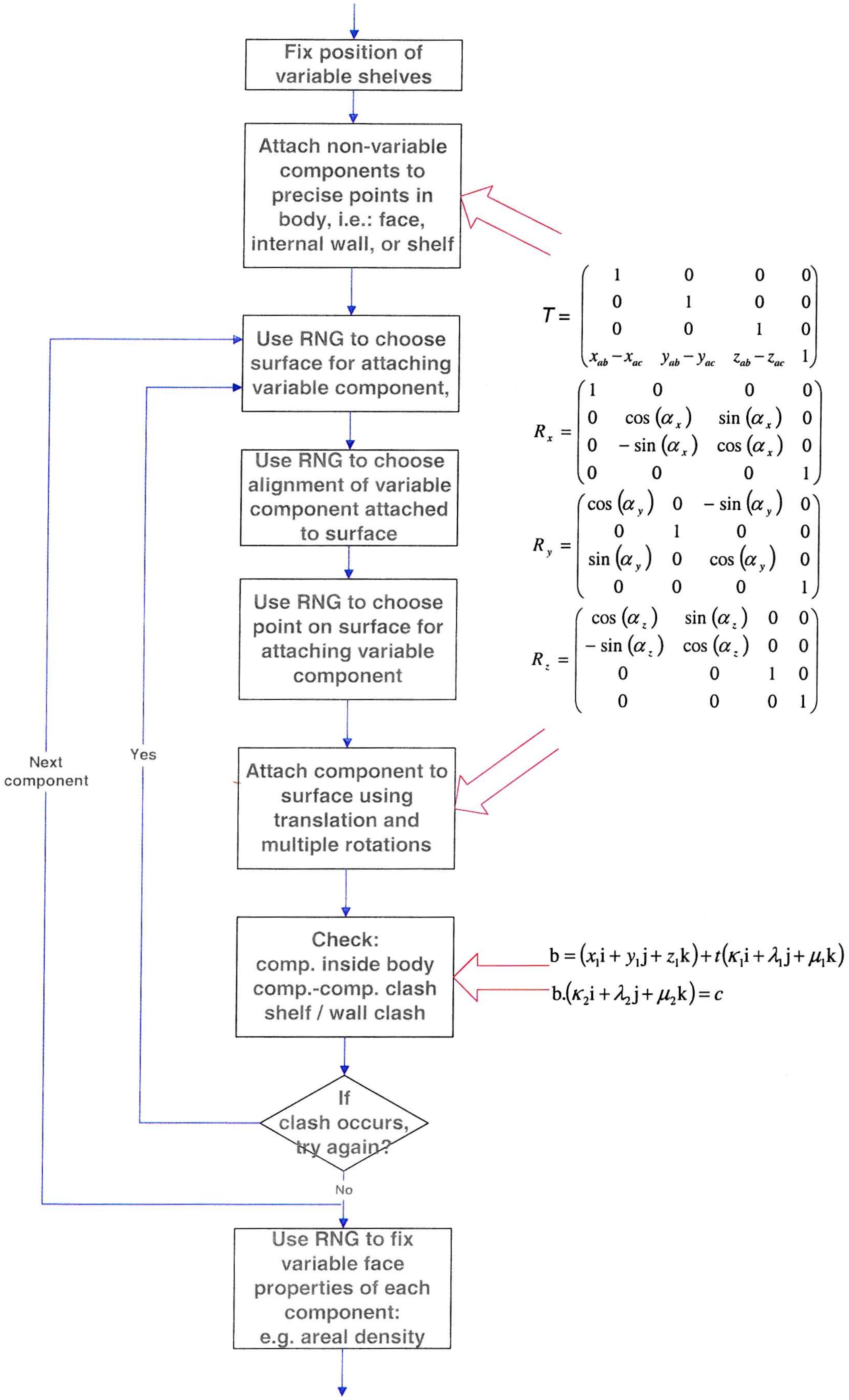

Fix position of variable shelves
Attach non-variable components to precise points in body, i.e.: face, internal wall, or shelf
Use RNG to choose surface for attaching variable component,
Use RNG to choose alignment of variable component attached to surface
Use RNG to choose point on surface for attaching variable component
Attach component to surface using translation and multiple rotations
Check: comp. inside body comp.-comp. clash shelf / wall clash
If clash occurs, try again?
Yes
No
Next component
Use RNG to fix variable face properties of each component: e.g. areal density
$T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ x_{ab}-x_{ac} & y_{ab}-y_{ac} & z_{ab}-z_{ac} & 1 \end{pmatrix}$
$R_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_x) & \sin(\alpha_x) & 0 \\ 0 & -\sin(\alpha_x) & \cos(\alpha_x) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
$R_y = \begin{pmatrix} \cos(\alpha_y) & 0 & -\sin(\alpha_y) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\alpha_y) & 0 & \cos(\alpha_y) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
$R_z = \begin{pmatrix} \cos(\alpha_z) & \sin(\alpha_z) & 0 & 0 \\ -\sin(\alpha_z) & \cos(\alpha_z) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
$\mathrm{b} = (x_1\mathrm{i} + y_1\mathrm{j} + z_1\mathrm{k}) + t(\kappa_1\mathrm{i} + \lambda_1\mathrm{j} + \mu_1\mathrm{k})$
$\mathrm{b}.(\kappa_2\mathrm{i} + \lambda_2\mathrm{j} + \mu_2\mathrm{k}) = c$

Figure 5-21 Flow diagram showing how a spacecraft design solution is created for the optimisation process

3. The survivability metric in Equation 5-53 is modified to include a mass balance constraint, as follows:

$$S = W_0\left[\frac{1}{(1+N_f)C_{lcc}}\right] + W_1\left[\frac{r_{\max} - r_a}{r_{\max}}\right] \qquad [5\text{-}54]$$

where W_0 and W_1 are weighting factors to distinguish between the relative importance of achieving a survivable design and a mass-balanced design. The terms r_a and r_{max} are the actual and maximum possible distance of the centre of mass from the optimum point, respectively. It is important to note that for this new equation to be viable, each of the square bracketed functions must be normalised. For the first bracketed function, normalisation is versus an idealised (or perfect satellite), where the number of satellite failures, N_f, is zero and the life cycle cost, C_{lcc}, includes zero shielding cost.

5.8.2 Genetic Populations

With the evaluation of an initial population of competing satellite designs completed, the software then steps through the process of creating subsequent genetic populations, or generations. The construction of further generations is slightly different to that of the initial population, in that the satellite designs are not derived completely at random. Essentially, the satellites in the second generation are derived from the satellite designs in the initial population. Likewise, third generation satellites are derived entirely from the second generation, and so on until the *N*th generation is reached.

The core mechanism for creating a 'next generation' population of satellites is identical to that described for the prototype model in the previous chapter, namely: selection, crossover and mutation. However, the conversion of design information into genetic code (and back again) is handled differently here because of the extra complexity of the problem. In the prototype, the satellites were very simple and had limited variability. It was therefore quite easy to convert each selected satellite design into a single, long binary chromosome string (prior to the crossover operation), and back again (after the mutation operation). This method though is not so easy to manage computationally when there are many options for introducing variability in a satellite design, as is the case here. A better approach is to convert a satellite design into a set of chromosomes, not just one. Each chromosome in the set is then a concatenation of variables of a given type. For example, the x, y, z, coordinates of the attachment points of all the components are transformed into three chromosome strings; the orientations of the components form another string; the shield types on the satellite body faces form yet another; and so on. The crossover process, where genetic information is swapped between two selected satellites to create two new satellites, is then simply a question of accessing the corresponding chromosomes in each set one at a time.

The remainder of the genetic algorithm process is as described for the prototype model. Each generation of newly created satellites is checked and evaluated in exactly the same way as for the initial population (see Section 5.8.1). After *N* generations have elapsed, the final population will contain the 'best' satellite design, i.e. a low cost design with a high survivability.

5.9 Discussion

The development of the SHIELD model, as described in the preceding sections, represents a significant percentage of the overall effort devoted to the Ph.D. research programme. The model combines several different and diverse disciplines including:

- Software engineering
- 3D vector / planar geometry (i.e. for satellite representation, ray-tracing, clash detection)
- Debris environment fluxes
- Poisson statistics
- Monte Carlo simulation
- Empirical damage relationships (i.e. ballistic limits, secondary debris clouds, and equipment vulnerability)
- Reliability assessment methodologies (i.e. criticality and Fault Tree Analysis)
- Cost modelling
- Genetic algorithm theory
- 3D graphics manipulation
- Windows Graphical User Interface development

Each of these represents a challenge in its own right, and indeed it is possible to devote time exclusively to researching one particular topic. In this first version of SHIELD, some elements of the software are perhaps more developed than others. This is to be expected given the complexity and novelty of the problem, and the time constraints of the research programme. Nevertheless, the software represents a significant step forward. It has now reached a stage where it is ready to be put through a range of simulations that will test each of its individual functions. Of particular concern will be the need to check (a) the validity of the impact simulation process, (b) the effectiveness of the internal damage assessment approach and (c) the efficiency of the automated search methodology.

One interesting early application of the model will be to see whether it can determine a relationship between the size of a debris impactor and the probability of failure of a satellite. Thus, it may be possible once and for all to dispense with the generalisation, adopted by the debris research community, that an impact with a debris particle larger than 1 cm will always result in a satellite failure.

6 Preliminary Validation and Initial Results from SHIELD

6.1 Introduction

There are three main elements of the SHIELD model that must be thoroughly tested and validated. They are the distributions of debris impact flux and penetration flux on a satellite; the evaluation of satellite failure probability compared to penetration probability; and the search method used to improve a satellite's protection strategy.

The first of these is relatively straightforward. Comparison of flux results from other validated models such as ESABASE/DEBRIS and NASA's BUMPER code is one option. Within the framework of the Inter Agency Debris Committee (IADC), test cases have already been defined to compare these models. The second element of SHIELD is much more difficult to validate at this time, since no comparable models are available, and relevant impact test data is at best very limited. That said, it is still possible to show how the assessment of failure probability provides an effective means of comparing radically different protection strategies. Finally, the efficiency of the genetic algorithm search method is a little easier to demonstrate, since one can scrutinise not only the resulting cost-effective protection solution, but also the rate of convergence. The other aspect of the genetic algorithm method worth demonstrating is its ability to handle the presence of a constraint, such as maintaining mass balance, during a search.

In the next few sections the validity of the SHIELD model is demonstrated as far as practicable, before the chapter focuses on applying the software to achieve a very important and unique result. That is, the identification of a relationship between satellite failure probability and debris particle size. Finally, the chapter concludes by demonstrating how the model can be applied to evaluating the survivability of an actual spacecraft design, namely the METOP Service Module. Several of these preliminary results from SHIELD have also been presented by Stokes *et al* (2000).

6.2 Validation of Impact and Penetration Fluxes on a Satellite

6.2.1 Specification of a Test Case Satellite

One of the test case satellites defined by the IADC for comparing impact flux predictions from different models is a simple cube-shaped geometry [IADC (2001)]. Each face of this box satellite has dimensions of 1m × 1m, with one face pointing in the direction of flight and another pointing towards the Earth.

The design of the satellite body itself is defined according to one of two structural options, namely:

1. Monolithic walls that are made from Al6061T6 aluminium. Two thicknesses of aluminium are recommended – semi-infinite and 1 mm – to characterise the numbers of craters and penetrations, respectively.
2. Whipple shields that comprise two Al6061T6 aluminium walls. The outer (or bumper) wall has a thickness of 0.2 cm, and is placed 10 cm in front of the rear (or back-up) wall, which has a thickness of 0.4 cm.

The material properties of Al6061T6 aluminium are summarised in Table 6-1.

Properties	Values
Brinell hardness	95
Density	2.713 g/cm^3
Speed of sound	5.1 km/s
Yield	47 ksi

Table 6-1 Material properties of Al6061T6 aluminium walls [IADC (2001)]

6.2.2 Specification of the Test Case Mission and Environment

The IADC-defined mission parameters for the box satellite are summarised in Table 6-2.

Parameters	Values
Altitude	400 km (circular)
Inclination	51.6 degrees
Year	2002
Mission life	1 year

Table 6-2 Mission parameters of the IADC box satellite [IADC (2001)]

Various simulated environments are recommended for the target satellite to fly through. For debris, either the ORDEM96 or NASA90 (the previous model to ORDEM96) environments are suggested, and for meteoroids, the SSP30425 or Grün environments [IADC (2001)]. Any debris particles impacting the satellite are assumed to have a density of 2.8 g/cm^3, whereas the density of any impacting meteoroids is fixed at 1 g/cm^3.

6.2.3 Specification of the required Flux Analyses

The IADC specifies four sets of results that should be calculated for each face of the box satellite, as follows:

1. The probability of no impacts (PNI), the number of impacts from particles ≥ 0.1 mm in diameter, and the number of impacts from particles ≥ 1 cm in diameter.
2. The probability of no penetration (PNP) and the number of craters with penetration depth ≥ 0.1 mm, assuming the semi-infinite monolithic walls are used.
3. The PNP and the number of penetrations, assuming 1-mm thick monolithic walls are used.
4. The PNP and the number of penetrations, assuming Whipple shields are used.

The calculation of PNP and number of penetrations is of course dependent on the choice of ballistic limit equations for the monolithic and Whipple shield walls. For consistency, the IADC specifies the equations provided by Christiansen (1993). These are also listed, in a different format, in Chapter 3.

6.2.4 Flux Results from ESABASE/DEBRIS and BUMPER

Comparisons between ESABASE/DEBRIS and BUMPER have already been conducted for the box satellite. These are summarised in Tables 6-3 and 6-4, which show the numbers of impacts and penetrations distributed on the box.

On the whole, the results reveal that there is reasonable agreement between the two models. Notable discrepancies are attributable to differences in the use of the damage equation, and to limitations in the treatment of the directionality of impacting particles. Specifically, for the ORDEM96 model, the ESABASE implementation included the elevation dependence of the fluxes, while this was not the case for BUMPER. Also, for the meteoroid fluxes, BUMPER used a different built-in value for density.

The IADC recommends that these results should serve as a benchmark against which other tools can be compared.

6.2.5 Flux Results from SHIELD

The equivalent SHIELD results for the numbers of impacts and penetrations on the box satellite are given in Table 6-5. Before discussing the results it is worth reiterating that SHIELD is currently designed only to accept debris environment flux data from the IDES model. The possibility of utilising NASA90, ORDEM96, and Grün fluxes within SHIELD is a matter that will be explored in the long term. Nevertheless, this does not prevent SHIELD from being validated, provided one understands how the IDES fluxes differ from those generated by other models.

One important data point in Table 6-5 to compare with Tables 6-3 and 6-4 is the total number of impacts greater than 0.1 mm. Using the IDES generated environment fluxes, SHIELD predicts a value of ~0.84 impacts. This is approximately an order of magnitude less than the values predicted by ESABASE/DEBRIS and BUMPER (when using the ORDEM96 fluxes). The cause of the discrepancy is explained by differences in the IDES and ORDEM96 flux results. This is seen quite clearly when one compares the orbit-integrated fluxes from each of the models. IDES predicts a total flux of 0.67 particles/m^2/year, whereas ORDEM96 gives a value of 6.92 particles/m^2/year, i.e. a factor of 10.3 higher. The reason for the difference is that the deterministic approach, as used by models such as IDES and MASTER, does not yet characterise completely and accurately all of the small-size debris sources, whereas the ORDEM96 model is derived entirely from a variety of measurement data. If one rather crudely applies the factor 10.3 to the above SHIELD result, then the number of impacts on the box satellite becomes 8.65, which is comparable with the ESABASE/DEBRIS and BUMPER results.

Another important set of data to compare in the tables is the distribution of impacts on the box satellite. For all three models the largest number of impacts occurs on the front face of the box, followed by the port and starboard side faces. This is exactly as one would expect given the directionality of the fluxes generated by IDES and ORDEM96, as previously illustrated in Figure 3-9. The predicted ratio of front to side face impacts is 2.0:1 for ESABASE/DEBRIS, 1.9:1 for BUMPER, and 1.8:1 for SHIELD. Thus, SHIELD generates slightly more impacts on the side faces, which is attributable to differences in flux directionality between IDES and ORDEM96.

In the final part of the validation, it is necessary to compare the numbers of penetrations predicted by the three models. SHIELD predicts there will be ~0.079 penetrators on the box satellite, which is a factor of 5.2 less than the ESABASE/DEBRIS result, and a factor of 5.8 less than the BUMPER result. Again, this discrepancy is explained by the flux differences between IDES and ORDEM96. Most of the particles penetrating the 1 mm thick aluminium faces are at least 0.215 mm in diameter, as illustrated by the SHIELD derived data in Table 6-6. The IDES and ORDEM96 orbit-integrated fluxes for particles $\geq$ 0.215 mm are 0.13 particles/m^2/year and 0.76 particles/m^2/year, respectively. That is, the IDES flux is a factor of 5.8 less, which is comparable to the SHIELD under-prediction factor for the number of penetrators on the box.

In summary, these validation results confirm that the numbers of impacts and penetrations derived using SHIELD are consistent with the predictions from ESABASE/DEBRIS and BUMPER.

Simulation	Front	Starboard	Port	Back	Space	Earth	Total
NASA90 Debris							
# impacts > 0.1 mm	2.37E+00	1.10E+00	1.07E+00	0.00E+00	0.00E+00	0.00E+00	4.56E+00
# penetrations of 1 mm Al	2.50E-01	5.50E-02	5.50E-02	0.00E+00	0.00E+00	0.00E+00	3.60E-01
ORDEM96 Debris							
# impacts > 0.1 mm	4.70E+00	2.32E+00	2.37E+00	6.00E-02	6.60E-02	6.10E-02	9.58E+00
# penetrations of 1 mm Al	2.60E-01	7.20E-02	7.40E-02	1.40E-04	1.90E-04	1.40E-04	4.10E-01
Grün Meteoroids							
# impacts > 0.1 mm	7.37E+00	3.48E+00	3.57E+00	1.10E+00	5.22E+00	4.40E-01	2.12E+01
# penetrations of 1 mm Al	3.20E-01	1.50E-02	8.00E-02	1.50E-02	1.10E-01	5.40E-04	6.00E-01

Table 6-3 ESABASE/DEBRIS-derived numbers of impacts and penetrations on a box satellite [IADC (2001)]

Simulation	Front	Starboard	Port	Back	Space	Earth	Total
NASA90 Debris							
# impacts > 0.1 mm	2.32E+00	1.07E+00	1.07E+00	0.00E+00	0.00E+00	0.00E+00	4.46E+00
# penetrations of 1 mm Al	2.22E-01	5.42E-02	5.42E-02	0.00E+00	0.00E+00	0.00E+00	3.31E-01
ORDEM96 Debris							
# impacts > 0.1 mm	4.60E+00	2.38E+00	2.38E+00	6.61E-02	0.00E+00	0.00E+00	9.42E+00
# penetrations of 1 mm Al	2.86E-01	8.55E-02	8.55E-02	6.68E-04	0.00E+00	0.00E+00	4.57E-01
Grün Meteoroids							
# impacts > 0.1 mm	4.15E+00	1.84E+00	1.84E+00	4.77E-01	2.61E+00	2.65E-01	1.12E+01
# penetrations of 1 mm Al	3.44E-01	8.58E-02	8.58E-02	1.38E-02	1.12E-01	7.13E-03	6.49E-01

Table 6-4 BUMPER-derived numbers of impacts and penetrations on a box satellite [IADC 2001)]

Simulation	Front	Starboard	Port	Back	Space	Earth	Total
IDES Debris							
# impacts > 0.1 mm	3.91E-01	2.17E-01	2.14E-01	2.00E-03	1.06E-02	9.00E-03	8.43E-01
# penetrations of 1 mm Al	5.38E-02	1.05E-02	1.43E-02	< 1.00E-04	< 1.00E-04	< 1.00E-04	7.86E-02

Table 6-5 SHIELD-derived numbers of impacts and penetrations on a box satellite

Impactor size, d (mm)	# penetrations of 1 mm Al
d ≥ 0.100	7.86E-02
d ≥ 0.215	7.79E-02
d ≥ 0.464	1.54E-02
d ≥ 1.000	2.00E-03

Table 6-6 SHIELD-derived numbers of penetrations on a box satellite

6.3 Demonstration of Satellite Survivability Evaluation

The evaluation of satellite survivability in terms of failure probability, as opposed to penetration probability, is a crucial element of the SHIELD model. It allows us to quantify and compare radically different protection solutions, and make informed judgements on the most effective strategy. One of the best ways to illustrate this potential benefit, is to examine how the solutions compare when applied to simple idealised satellites. Although this is not a validation of the failure probability computation, it can nevertheless give a first impression of whether the results are consistent with what one might reasonably expect.

6.3.1 Comparison of Competing Protection Strategies

In Chapter 5, a simple question was posed regarding whether it was better to either add shielding to the body of a satellite, or to internal critical components, or to rearrange the position of the internal components. The answer to this question though is difficult, and requires much investigation. As a first step, attention is focused on the simplest of satellite designs. For this, a box-shaped satellite comprising two internal equipment units is entered into SHIELD, where one of the units is critical to mission success, and the other is not. The baseline design parameters for this satellite are shown in Table 6-7.

A geometrical representation of the baseline satellite design, showing the internal arrangement of units, shelves, and structure walls, is given in Figure 6-1. The red-coloured unit is the mission-critical item; i.e. it has a criticality of 1.0 (see Table 4-6 for criticality definitions). In this particular configuration it is located close to the face pointing in the flight direction. The blue coloured unit has a criticality of 0 assigned to it, and therefore is non-critical.

Design parameters	Values
Mission:	
lifetime	10 years
orbit	800 km, circular, 98 degree polar
attitude control	3-axis stabilised, (-x: flight direction, -z: Earth)
Environment:	
debris	IDES generated fluxes
meteoroids	None
Satellite geometry:	
dimensions	2m x 2m x 2m
body side faces	0.2cm thick Al plate
body base face	0.2cm thick Al plate
body top face	0.2cm thick Al plate
internal wall 1	0.05cm thick Al plate
shelf 1	0.7m above body base, 0.05cm thick Al plate
shelf 2	1.4m above body base, 0.05cm thick Al plate
Internal units:	
unit #1	Mission-critical (criticality = 1.0) 0.4m x 0.4m x 0.4m, 0.1 cm thick Al plate faces, attached to shelf 1
unit #2	Non-critical (criticality = 0) 0.4m x 0.4m x 0.4m, 0.1 cm thick Al plate faces, attached to shelf 1

Table 6-7 List of design parameters for a box satellite with two units

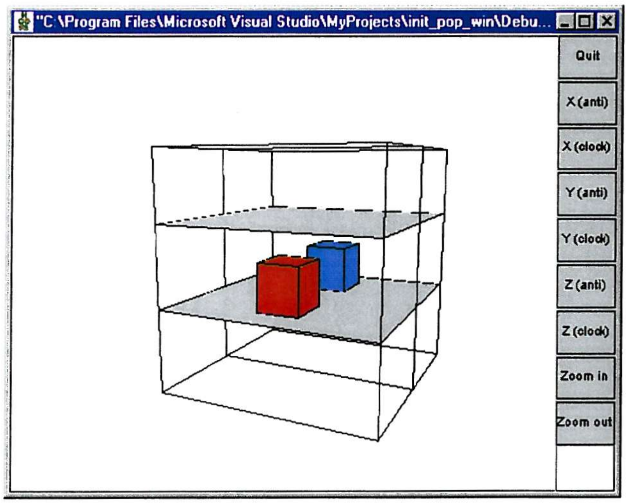

"C:\Program Files\Microsoft Visual Studio\MyProjects\init_pop_win\Debu...
Quit
X (anti)
X (clock)
Y (anti)
Y (clock)
Z (anti)
Z (clock)
Zoom in
Zoom out

Figure 6-1 3-D geometry of baseline design of satellite

With this input data defined, SHIELD is used to compute the following results:

- ❑ the number of impacts and penetrations on each body face
- ❑ the number of penetrators that are lethal
- ❑ the overall probability of failure

A total of 1000 Monte Carlo lifetime simulations are performed to generate a sufficiently large distribution of impacting particles on the baseline satellite. 16,325 particles with diameter ≥ 0.2 mm are observed. That is, the satellite can expect on average ~16 impacts from particles larger than 0.2 mm during the course of its mission life. In other words, from Equation 3-10, the probability of impact is 0.9999999. Similarly, the average number of penetrations during the mission is computed to be ~3.3, which equates to a probability of penetration of 0.964. The expected number of impacts and penetrations distributed on the baseline satellite is given in Table 6-8. Not surprisingly the front (flight direction) face receives the greatest number of impacts and penetrations, and therefore one assumes should be the focus of any protection strategy.

The size distribution of all the impactors and penetrators on the baseline satellite is shown in Figure 6-2. As expected, the number of impacts increases as the size of the particles becomes smaller. However, the smaller particles do not necessarily penetrate. For example, only ~13% of impactors in the size range 0.215 mm to 0.464 mm penetrate the satellite body. It is only when particles larger than ~1 mm impact the satellite that the likelihood of a penetration becomes virtually certain.

The occurrence of a penetration through the satellite body structure though does not necessarily result in failure of the satellite. In this particular example, the ~3.3 penetrators cause only 0.151 satellite failures. Therefore, from Equation 3-12, the through-life probability of failure of the satellite, due to debris impact, is 0.140. Again, the size distribution of the lethal penetrators is also shown in Figure 6-2.

Simulation	Front	Starboard	Port	Back	Space	Earth	Total
# impacts > 0.2 mm	10.470	2.696	2.754	0.039	0.181	0.185	16.325
# penetrations on baseline	2.903	0.225	0.202	<0.001	0.001	0.001	3.332
# penetrations on baseline with front face 1mm thicker	1.060	0.225	0.202	<0.001	0.001	0.001	1.489

Table 6-8 Expected numbers of impacts and penetrations on a box satellite with and without enhanced body protection

Strategy #	Design	Number of penetrators	Probability of penetration	Number of failures	Probability of failure	Relative cost	Survivability metric
	Ideal			0.000	0.000	1.000	1.000
1	Baseline	3.332	0.964	0.151	0.140	1.000	0.869
2	Units' position swapped	3.332	0.964	0.053	0.052	1.000	0.950
3	Body front face thicker	1.489	0.774	0.077	0.074	1.009	0.929
4	Critical unit walls thicker	3.332	0.964	0.145	0.135	1.001	0.873

Table 6-9 Comparison of survivability of box satellite with different protection strategies

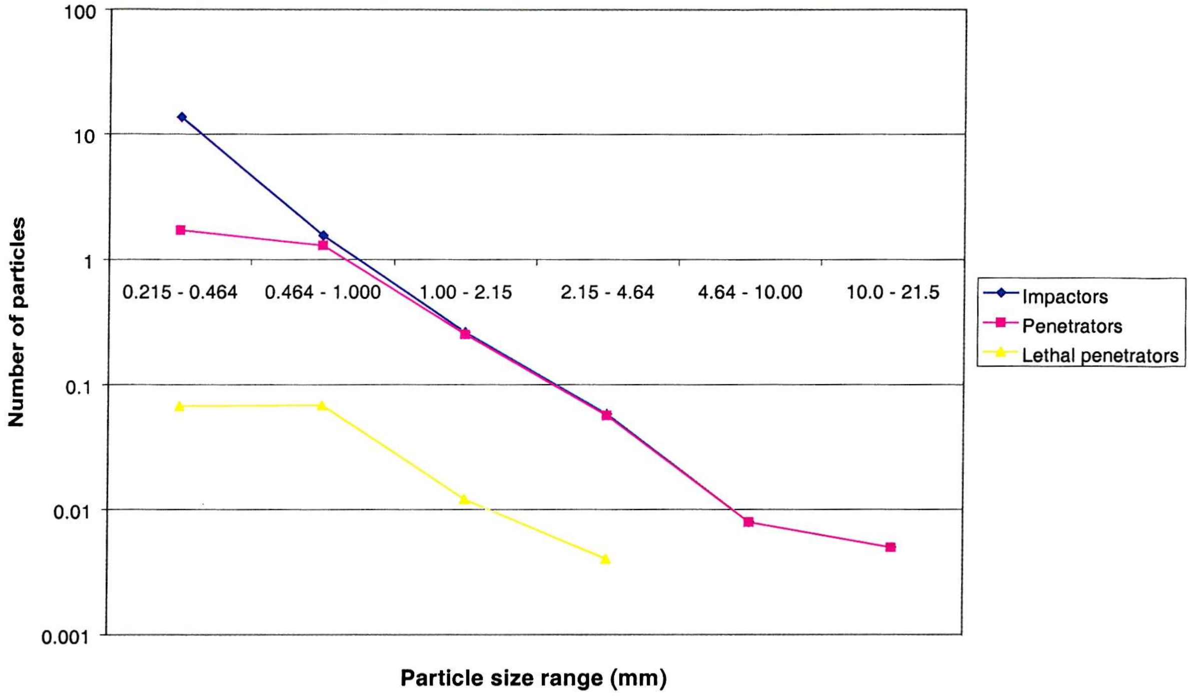


Figure 6-2 Size distribution of all impactors, all penetrators, and all lethal penetrators on the spacecraft body

From the above computations it is clear that there is a substantial difference between the probability of penetration (~96%) and probability of failure (14%) of this particular unmanned spacecraft. The difference is quite marked in this example because there is only one mission-critical component inside the satellite, which occupies just 0.8% of the total internal volume of the satellite. A more representative satellite with several critical units should have a probability of failure that is closer to the probability of penetration.

In order to investigate the benefits of different protection solutions, the baseline satellite design is then modified in the three distinct ways mentioned earlier. First, the position of the two units is swapped, so that the mission-critical unit is further away from the front face and is protected by the non-critical item (see Figure 6-3). Second, the benefit of increasing the areal density of the front face is examined by increasing its thickness by 1 mm. And finally, the areal density of the walls on the mission-critical unit is increased. Again, this is done by increasing the casing thickness by 1 mm.

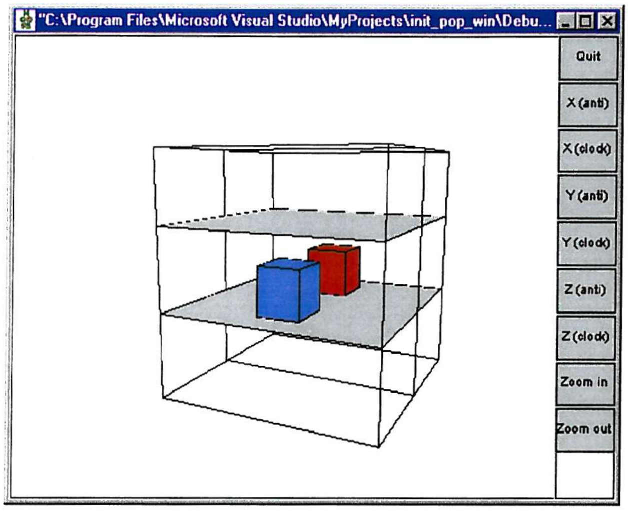

"C:\Program Files\Microsoft Visual Studio\MyProjects\init_pop_win\Debu...
Quit
X (anti)
X (clock)
Y (anti)
Y (clock)
Z (anti)
Z (clock)
Zoom in
Zoom out

Figure 6-3 3-D geometry of satellite with position of units swapped

It is far from obvious, even for this very simple idealised satellite, which of these three protection enhancement options will be the best. However, SHIELD's ability to assess probability of failure allows one to make direct comparisons between these completely different protection strategies. The results from these simulations are summarised in Table 6-9 and compared with the results from the baseline design.

Examining Table 6-9 the first result of note is that swapping the position of the units, so that the critical unit is further away from the vulnerable front face, does indeed produce a significant drop in failure probability as expected (i.e. from 14% to ~5%). This is primarily attributed to the additional protection that the critical unit receives from the intervening non-critical unit. Because the costs are unchanged, it is clear that this protection strategy is an improvement over the baseline design. The measure of this improvement is quantified in the 'survivability metric' column of the table. As discussed in Chapter 5, the values of *S* in the table are normalised against a notional 'ideal' design, where the cost of protection is zero and there is zero probability of failure. The closer *S* tends to 1, the nearer the design is to a perfect solution. For the case where the units are swapped, there is an ~9% increase in the survivability metric, which is significant enough to merit consideration.

It is interesting to compare this result with that from the recent ESA-funded study [Turner *et al* (1999)], which was briefly discussed at the beginning of Chapter 4. In that study, the author employed a simple, but entirely different metric approach, to assess the potential benefit of unit relocation in the METOP satellite Service Module as a means of improving protection cost-effectively. In the investigation, one of the critical units (i.e. a reaction wheel) on the vulnerable front face of this module (refer to Figure 4-1) was relocated to the back face to see how much enhancement there would be in the satellite's survivability. It was revealed that the survivability of the new configuration was almost 50% higher than the original one. To observe such a dramatic improvement from a relatively simple design change was a surprising result. However, in the light of SHIELD's 'unit swap' result above, where failure probability is substantially reduced in a very similar manner, the importance of this protection strategy is further underlined.

The second strategy of increasing the areal density of the satellite body front face, as expected, leads to a decrease in the probability of penetration of the satellite. Consequently, the probability of failure of the satellite also reduces. However, the risk reduction is not quite as much as when the position of the units was swapped. This fact, combined with the additional mass and cost of the thickened front face, results in a survivability metric that is lower than the metric for the 'swapped units' strategy. Nevertheless, the strategy is still a noticeable improvement over the baseline design.

The final strategy of thickening the walls of the critical unit provides only a marginally better failure probability and survivability metric compared to the baseline design. On the face of it this result is perhaps surprising, since one might have expected the extra casing thickness on the critical unit to produce a more beneficial risk reduction. However, the reason is almost certainly a consequence of the two ballistic limit equations used in the assessment of the satellite's body penetration and then the internal damage. A single wall equation (see Section 3.4.6) is used first to compute the set of impactors that penetrate the 2-mm thick body structure. This is then followed by the ESA triple wall equation (see Section 3.4.7) to determine which of these penetrate the combination of body structure and unit casing. Examining the equation, one can see that there is no shield thickness term present for hypervelocity particle impacts (> 7 km/s). This means that the shielding effect of the body structure is ignored, and penetration is dominated by the back-up wall thickness term in the equation. In other words, the unit casing thickness defines whether unit penetration occurs. Since it has already been computed that the set of impactors will penetrate a 2-mm thick wall, then an increase in unit casing thickness from 1 to 2 mm has little effect. The peculiarity of this result therefore is a direct consequence of a limitation in the multiple wall equation.

From these results one can deduce two things. First, that simply adding shielding mass does not necessarily imply that a satellite's survivability is being enhanced in the most cost-effective fashion. And second, that consideration of the vulnerability and arrangement of equipment in a satellite is potentially a cost-effective protection strategy. That does not mean improved structural protection should always be ruled out for unmanned spacecraft. A 'real' spacecraft is different to this simple test case in several respects. For example, in a 'real' spacecraft there are many critical units, and so one would expect the probability of failure figures to be higher. The need for an enhanced structure in that instance may justify the associated costs. In practice, it is likely that an approach combining both protection strategies will provide the optimum survivability.

6.3.2 Potential Benefit of Rearranging Internal Components

To gain a slightly clearer understanding of the potential benefit of rearranging internal components it is worth looking at another slightly more complicated satellite example. The simple box-shape is retained, but instead the satellite now comprises ten internal equipment units. Five of the units are critical to mission success, the other five are not. Furthermore, in this example, the software is given complete freedom to position the ten units inside the satellite. That is, it can locate the units at random so that they sit on any of three shelves. All other design parameters are fixed as shown in Table 6-10.

In this mode of operation, the software can be used to randomly generate and evaluate not just one spacecraft design, but an entire population of designs such that each one has a unique internal configuration. By evaluating the failure probability and survivability metric of each design, one can see how much variation is possible and gain some insight into the difference between 'good' and 'bad' designs. The results of 100 such assessments are plotted in Figures 6-4 and 6-5.

Design parameters	Values
Mission:	
lifetime	10 years
orbit	800 km, circular, 98 degree polar
attitude control	3-axis stabilised, (-x: flight direction, -z: Earth)
Environment:	
debris	IDES generated fluxes
meteoroids	None
Satellite geometry:	
dimensions	2m x 2m x 2m
body side faces	0.2cm thick Al plate
body base face	0.2cm thick Al plate
body top face	0.2cm thick Al plate
internal wall 1	0.05cm thick Al plate
shelf 1	0.5m above body base, 0.05cm thick Al plate
shelf 2	1.0m above body base, 0.05cm thick Al plate
shelf 3	1.5m above body base, 0.05cm thick Al plate
Internal units:	
units #1 - 5	Mission-critical (criticality = 1.0) 0.4m x 0.4m x 0.4m, 0.1 cm thick Al plate faces, attached to any shelf
units #6 - 10	Non-critical (criticality = 0) 0.4m x 0.4m x 0.4m, 0.1 cm thick Al plate faces, attached to any shelf

Table 6-10 List of design parameters for a box satellite with ten units

Both plots reveal that there is a significant variation across the 100 designs. The most survivable configuration (#19) has a probability of failure of 0.245, as compared to the 'worst' configuration (#17), which has a failure probability of 0.506. Such an improvement is very difficult to ignore. This is particularly so when one considers that system designers usually strive to maximise the reliability of spacecraft by incorporating redundancy and expensive space-qualified components. Therefore, the benefit that can be gained by carefully considering the vulnerability and internal arrangement of equipment in a satellite should not be underestimated.

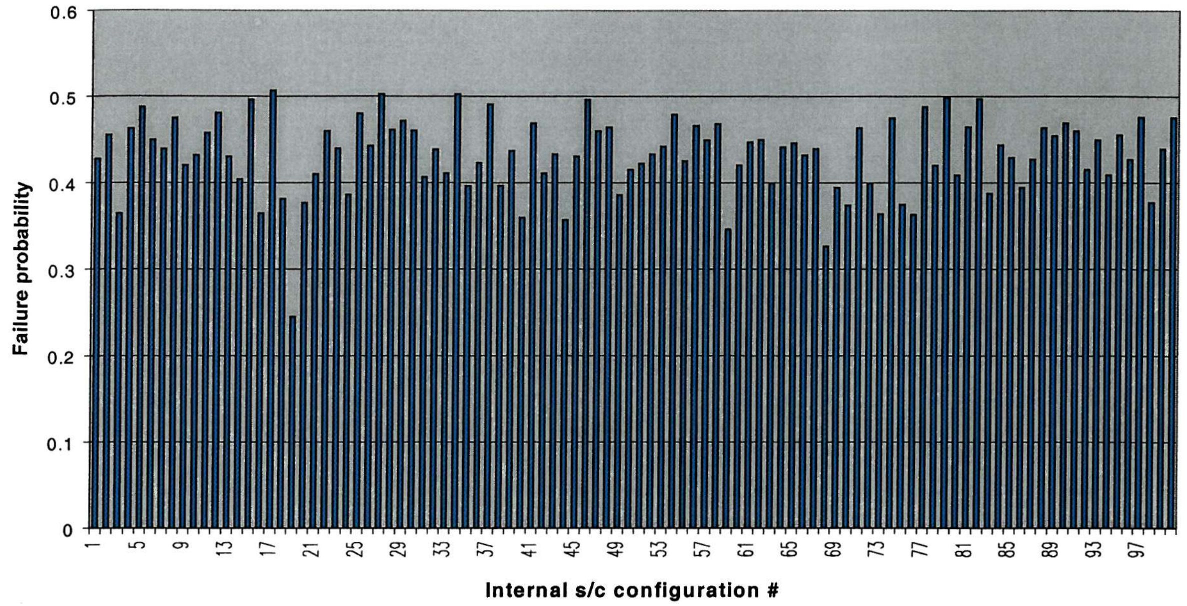


Figure 6-4 *Failure probabilities of 100 randomly generated internal configurations*

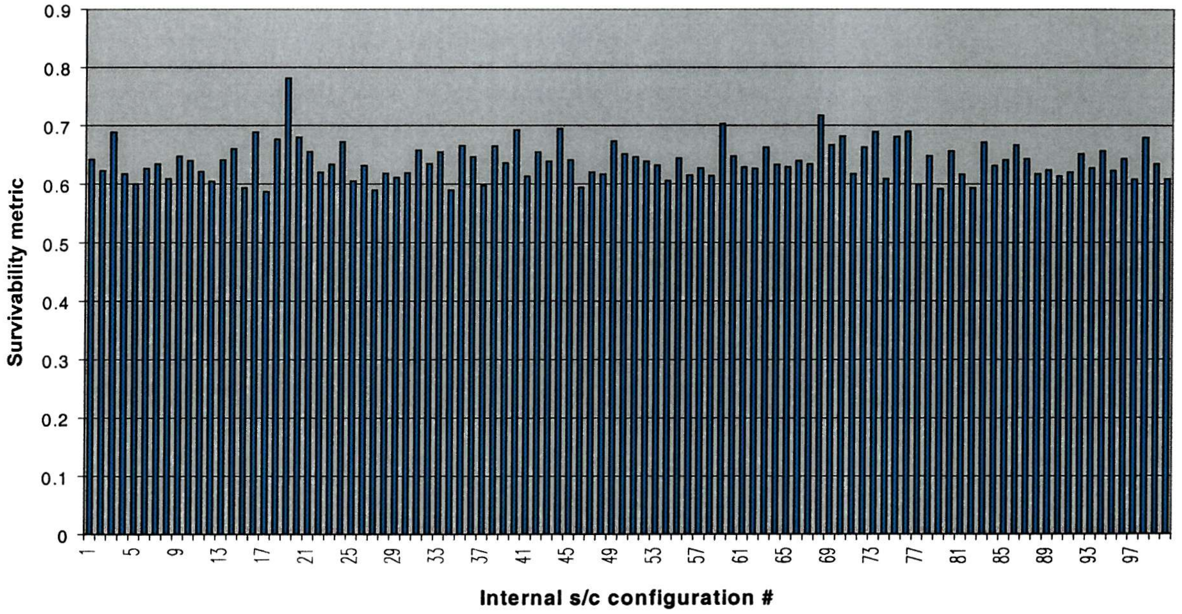


Figure 6-5 *Survivability metrics of 100 randomly generated internal configurations*

To understand why design #17 is the 'worst' design it is necessary to take a closer look at its internal configuration. Figures 6-6 and 6-7 provide two different views of the satellite; one through the front face, and the other through the port face. It is clear from these that all of the critical units are in exposed positions, (a) because of their close proximity to the vulnerable front face, and (b) because there are no intervening units to protect them. Therefore, impactors penetrating the front face will be more likely to cause catastrophic failure of one of the critical units and terminate the mission.

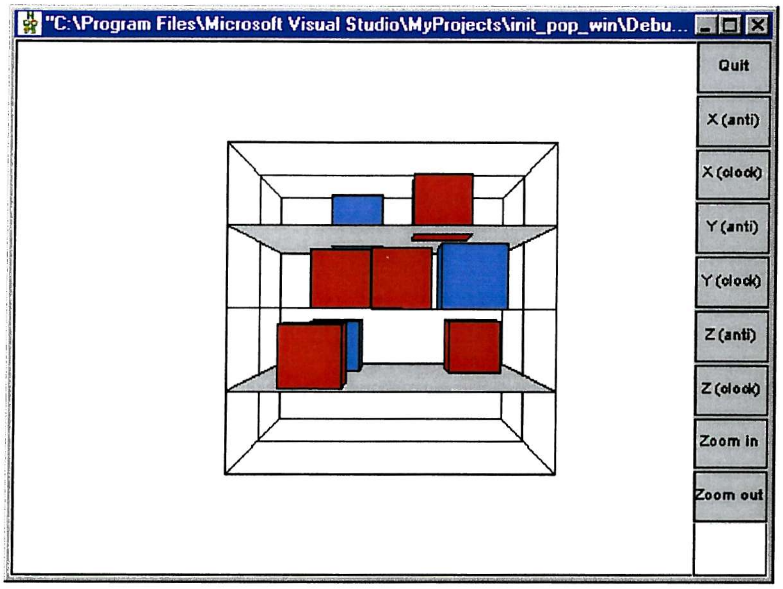

"C:\Program Files\Microsoft Visual Studio\MyProjects\init_pop_win\Debu...
Quit
X (anti)
X (clock)
Y (anti)
Y (clock)
Z (anti)
Z (clock)
Zoom in
Zoom out

Figure 6-6 Lowest survivability configuration (#17) – view of front face

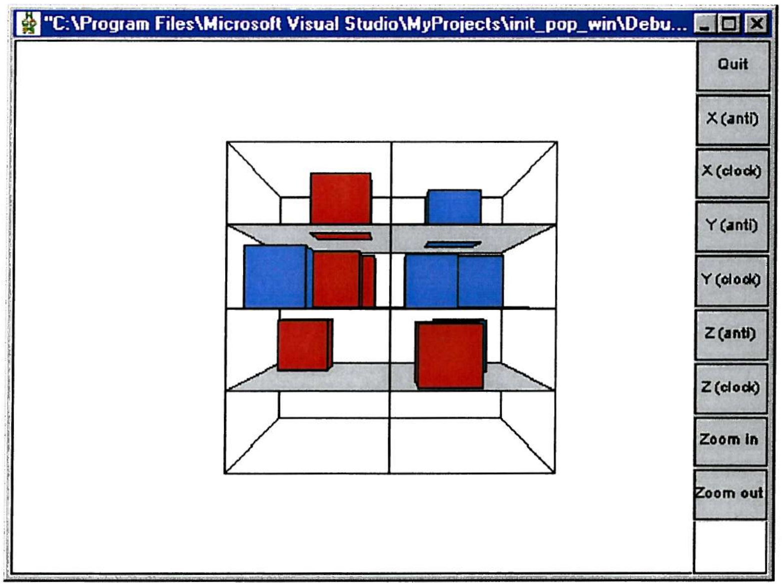

"C:\Program Files\Microsoft Visual Studio\MyProjects\init_pop_win\Debu...
Quit
X (anti)
X (clock)
Y (anti)
Y (clock)
Z (anti)
Z (clock)
Zoom in
Zoom out

Figure 6-7 Lowest survivability configuration (#17) – view of port face

In a similar fashion, it is possible to examine the internal configuration of design #19 to understand why it is better than the other designs. Figures 6-8 and 6-9 provide two views of the satellite; one through the front face, and the other through the front, port and space faces.

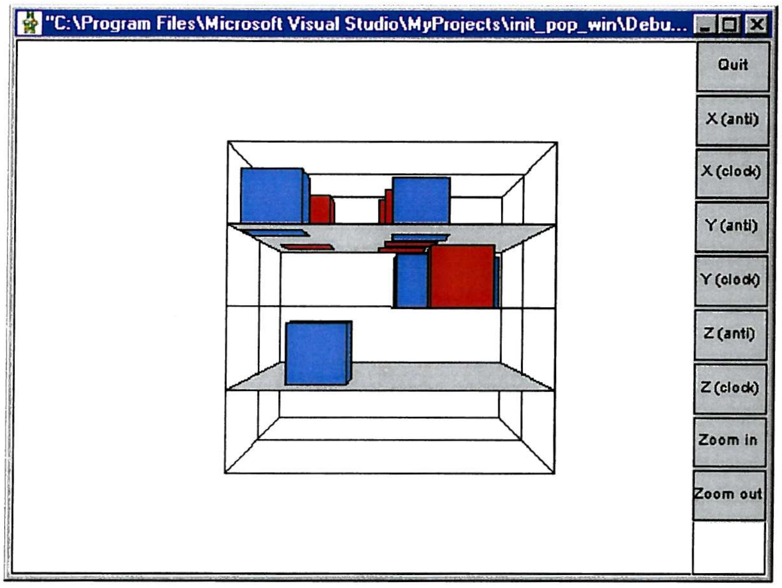

"C:\Program Files\Microsoft Visual Studio\MyProjects\init_pop_win\Debu...
Quit
X (anti)
X (clock)
Y (anti)
Y (clock)
Z (anti)
Z (clock)
Zoom in
Zoom out

Figure 6-8 *Highest survivability configuration (#19) – view of front face*

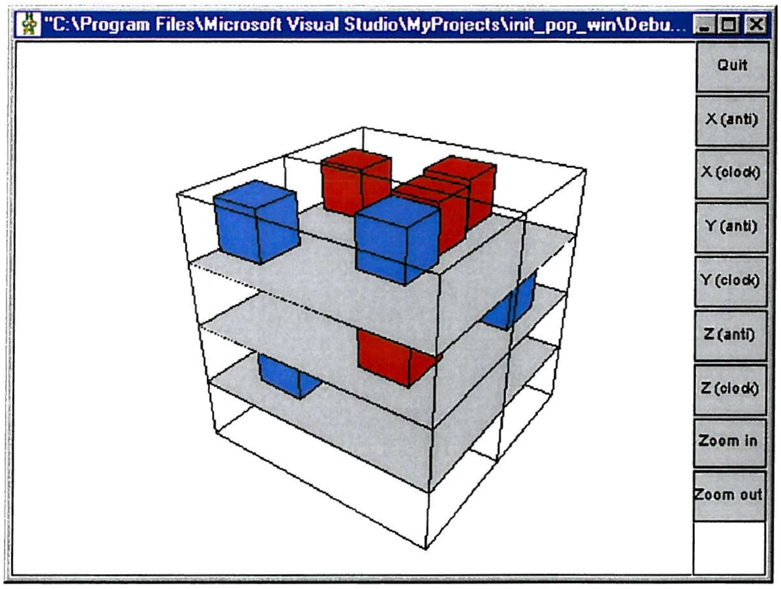

"C:\Program Files\Microsoft Visual Studio\MyProjects\init_pop_win\Debu...
Quit
X (anti)
X (clock)
Y (anti)
Y (clock)
Z (anti)
Z (clock)
Zoom in
Zoom out

Figure 6-9 *Highest survivability configuration (#19) – view of front, port, and 'space' faces*

In Figure 6-8, one can see that, with the exception of one critical unit, all of the critical units are well protected from the vulnerable front face by intervening non-critical units. Figure 6-9 shows that three of the critical units are located near the back face and space face, both of which experience a far lower probability of penetration. The remaining critical unit is hidden from view, and is tucked behind three other units, near the back face on the middle shelf.

Although one might consider this configuration to be a satisfactory protection solution, it is quite apparent from examining the internal arrangement that the design has a rather unbalanced mass distribution, i.e. the satellite is 'top heavy'. Thus, in reality, the design may not be a very practical solution. In the previous chapter, a modification was introduced to the survivability metric, which allowed the mass balance constraint to be incorporated as part of the evaluation of a design solution. If this is now switched on in the SHIELD software, then it is possible to re-evaluate the above 100 random designs, and see whether another configuration emerges as the 'best'.

By assuming that the desire for mass balance carries an equal weighting to the desire for protection, i.e. $W_0 = W_1 = 0.5$ in Equation 5-54, then the 100 designs have the survivability metrics shown in Figure 6-10.

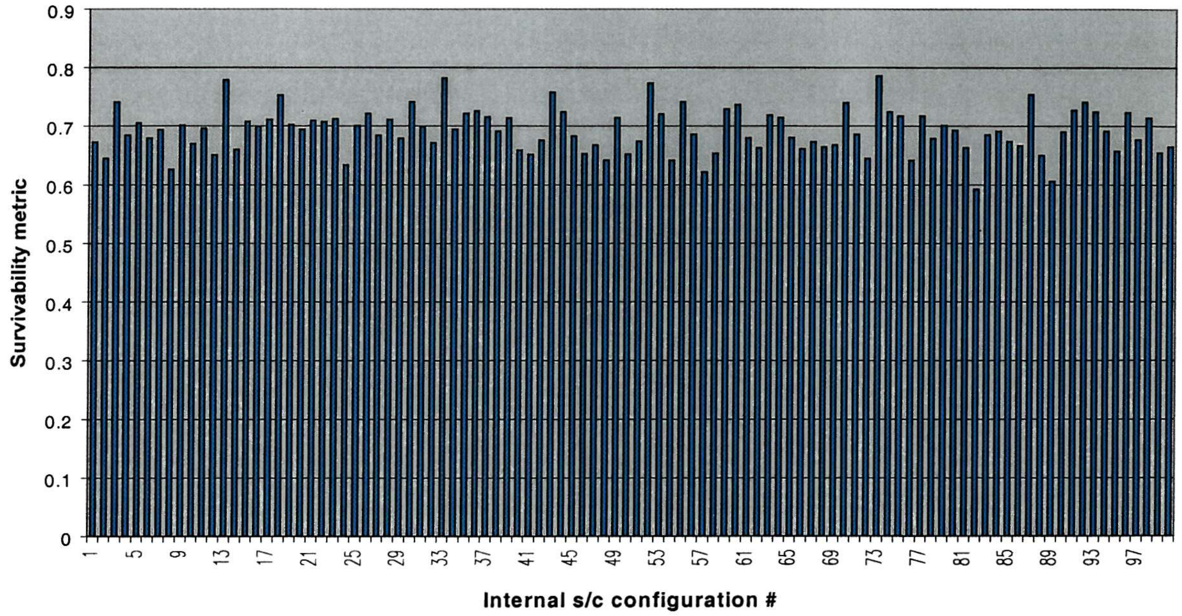


Figure 6-10 Survivability metrics of 100 randomly generated internal component configurations (with mass balance constraint switched on)

These are completely different to the metrics derived in Figure 6-5. Now, design #19 is no longer considered the 'best', but is relegated to that of a fairly average solution. Instead, the accolade goes to design #73. Essentially, this is the best compromise solution. It is the one that has achieved a reasonably low failure probability (= 0.364), while at the same time distributing the equipment fairly evenly inside the spacecraft. Figure 6-11 illustrates both of these points.

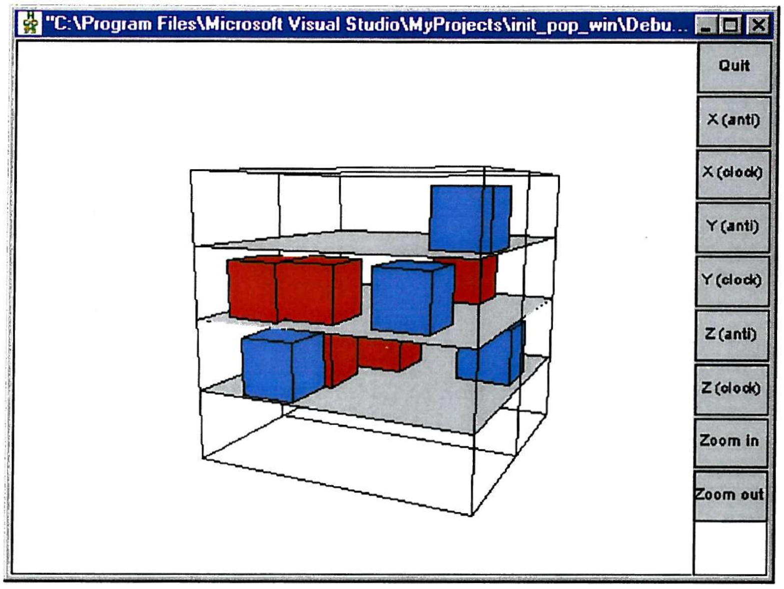

"C:\Program Files\Microsoft Visual Studio\MyProjects\init_pop_win\Debu...
Quit
X (anti)
X (clock)
Y (anti)
Y (clock)
Z (anti)
Z (clock)
Zoom in
Zoom out

Figure 6-11 Highest survivability configuration when mass balance constraint is introduced (#73) – view of front face

One can see that two critical units are in exposed positions close to the front face, while the other three are reasonably well protected. This explains why the design does not have a failure probability quite as low as that observed for design #19. However, unlike design #19, the internal configuration is not 'top heavy', which is sufficient to qualify it as the more practical solution.

It is not too difficult to see that much better internal arrangements should be possible, which satisfy the desire for high survivability while ensuring good mass balance. The 100 randomly generated designs here offer a tantalising glimpse of what might be possible if one were to consider and evaluate the vulnerability and arrangement of equipment on a satellite as part of a standard protection strategy.

6.4 Investigation of Genetic Algorithm for Identifying Effective Protection

The preceding analyses illustrate how a typical spacecraft can have a relatively poor survivability if no consideration is given to its internal architecture. Indeed, this is very likely. If one plots the failure probabilities in Figure 6-4 as a frequency distribution, this fact becomes very apparent (see Figure 6-12).

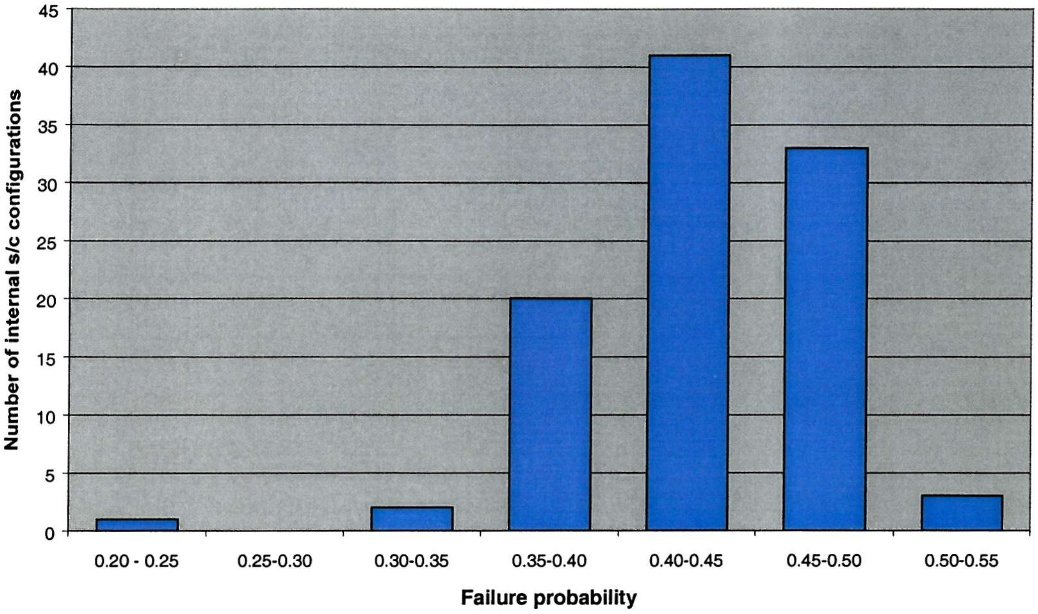


Figure 6-12 Frequency distribution of the 100 failure probabilities in Figure 6-4

The majority of failure probabilities occur between 0.350 and 0.500, with a mean of 0.434. Compared to the lowest probability of 0.245, these are substantially higher. Furthermore, the distribution appears to be skewed to the left, i.e. it has a long tail on the left hand side, which would seem to indicate that even lower failure probabilities should be possible. Certainly, if every conceivable design was generated, evaluated, and its survivability result plotted on the distribution, then one may see the appearance of a long tail, at the end of which lies the design solution with the lowest failure probability. However, such an approach is not a practical proposition because of the large number of permutations for arranging the internal equipment. This is where the genetic algorithm technique can be a valuable tool, as it provides a reasonably efficient means of searching for designs near the tip of the distribution tail.

6.4.1 Unconstrained Search - Box Satellite with Ten Components

In the event that a designer should wish to find an improved spacecraft protection strategy, the SHIELD software can be run in an automated search mode using its in-built genetic algorithm. To illustrate the efficiency and potential benefit of this capability, it is worth applying it to the 10-component spacecraft specified in the preceding Section (6.3.2). In the case where there was no mass balance constraint, the most survivable design from the 100 generated had a metric value of 0.7806 (and as noted above a failure probability of 0.245). Whilst this design was considered to be quite reasonable, it was evident from the arrangement of the equipment that further improvements might be possible. With this goal in mind, the genetic algorithm was set up according to the parameters in Table 6-11, and a search performed. All spacecraft design parameters are as listed in Table 6-10.

Parameter	Value
Number of populations	50
Number of designs per population	100
Probability of crossover	0.70
Probability of mutation	0.05

Table 6-11 List of parameters used in genetic algorithm

The results of the search are illustrated in Figures 6-13 to 6-17.

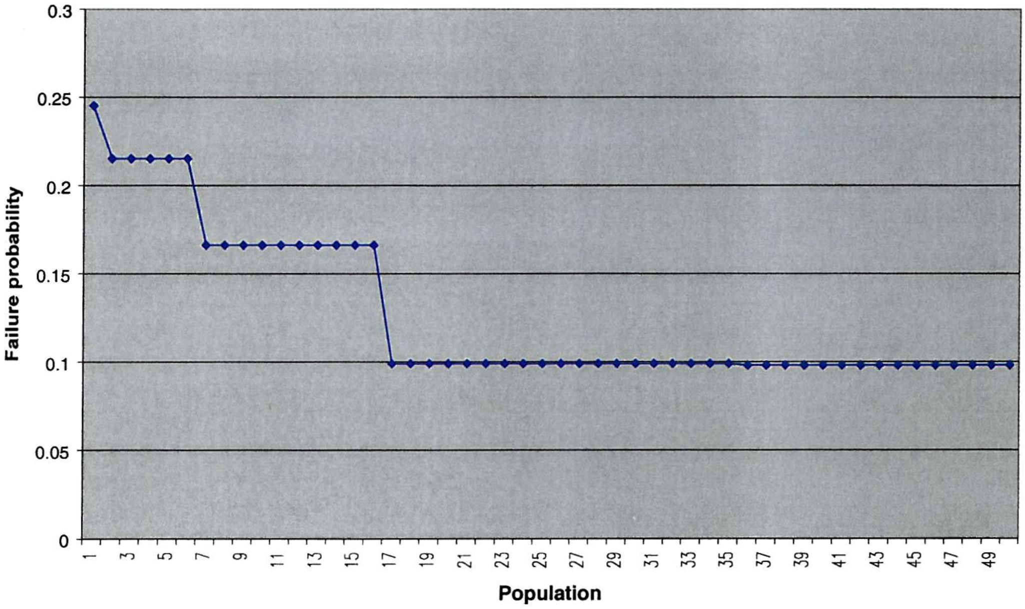


Figure 6-13 Plot of failure probability of most survivable design in each genetic population

Figure 6-13 shows the probability of failure of the most survivable design generated in each of the 50 populations. As one would hope, the search has found designs that represent an improvement over the 'best' one identified in the initial population. In the final population created by the genetic algorithm, the most survivable design has a failure probability of 0.098. This is considerably lower than the value of 0.245 observed in the initial population, thus confirming the earlier observation that there may be scope to further improve the internal arrangement. When compared to the average value of a completely random design (0.434), i.e.

one where no consideration has been given to optimising the arrangement, the difference is even more dramatic. This result really does emphasise that designers should be fully aware of the potential protection benefits achievable by carefully considering the spacecraft architecture design.

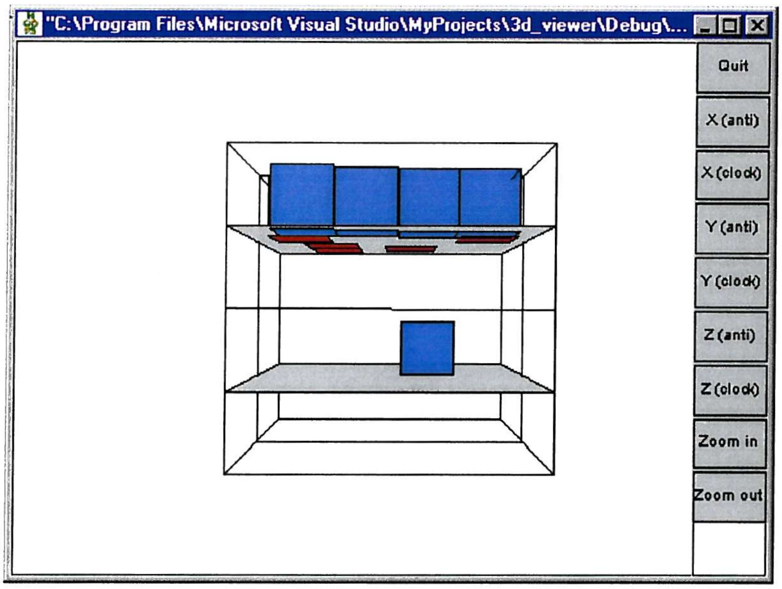

"C:\Program Files\Microsoft Visual Studio\MyProjects\3d_viewer\Debug\...
Quit
X (anti)
X (clock)
Y (anti)
Y (clock)
Z (anti)
Z (clock)
Zoom in
Zoom out

Figure 6-14 Highest survivability configuration in final genetic population – view of front face

Having identified the design with the best survivability in the final population, it is useful to examine its configuration to try to understand why it is the 'best'. Figure 6-14 shows the front face of the satellite, i.e. the face with the highest impact and penetration flux. It is very apparent that all five of the critical units are now well protected from this face by four intervening non-critical units, which effectively act as a shield. This is consistent with expectation.

The position of the critical units is seen more clearly in Figure 6-15. Interestingly, the genetic algorithm search has identified the top shelf as the most survivable location for critical equipment, rather than the central shelf. This result was also observed for the best design in the initial population, as reported in the discussion of Figures 6-8 and 6-9. The reason is that the space face experiences only a very small penetration flux compared to the front face (as observed in Table 6-8). By contrast, if the critical units were placed on the central shelf, they would be exposed to a much larger number of penetrators through the front face of the satellite. Even with the presence of the thin intervening shelves providing some additional shielding, the critical units are still more vulnerable.

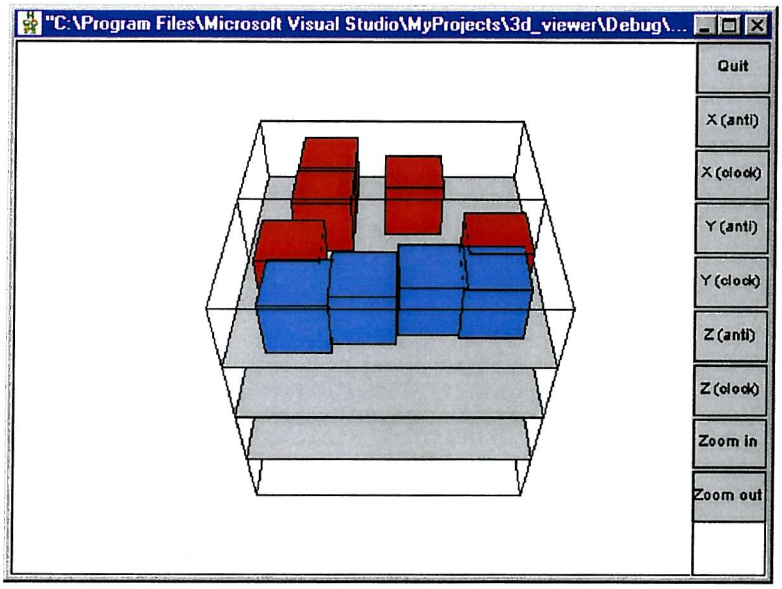

"C:\Program Files\Microsoft Visual Studio\MyProjects\3d_viewer\Debug\...
Quit
X (anti)
X (clock)
Y (anti)
Y (clock)
Z (anti)
Z (clock)
Zoom in
Zoom out

Figure 6-15 Highest survivability configuration in final genetic population – view of front and 'space' faces

In Figure 6-16, the survivability metric of the 'best' design in each population is plotted. Again, one can see a significant growth in the value between the initial and final populations, i.e. from 0.7806 to 0.9066. Perhaps the most interesting result to note from the graph is the rapid increase in the value of the metric during the early populations followed by a long, fairly flat plateau where improved designs are much more difficult to find. This is indicative firstly of the genetic algorithm's efficiency in converging quite quickly on a 'good' solution, and secondly the difficulty that genetic algorithms tend to have in focusing on the global optimum solution. As discussed in Section 4.2.2, genetic algorithms are sometimes hybridised to overcome this problem. However, for the application here, i.e. survivability enhancement, it is not yet clear whether such a modification is necessary.

Figure 6-17 shows a plot of the mean survivability metric of the 100 designs in each population. Again, one can see a growth trend between the initial and final populations. The result serves to highlight another feature of the genetic algorithm method, namely that the entire population of designs gradually improves during the evolutionary process, as one might expect. If a purely random process had generated the 50 populations, rather than the genetic algorithm, then one would of course expect to see no such growth.

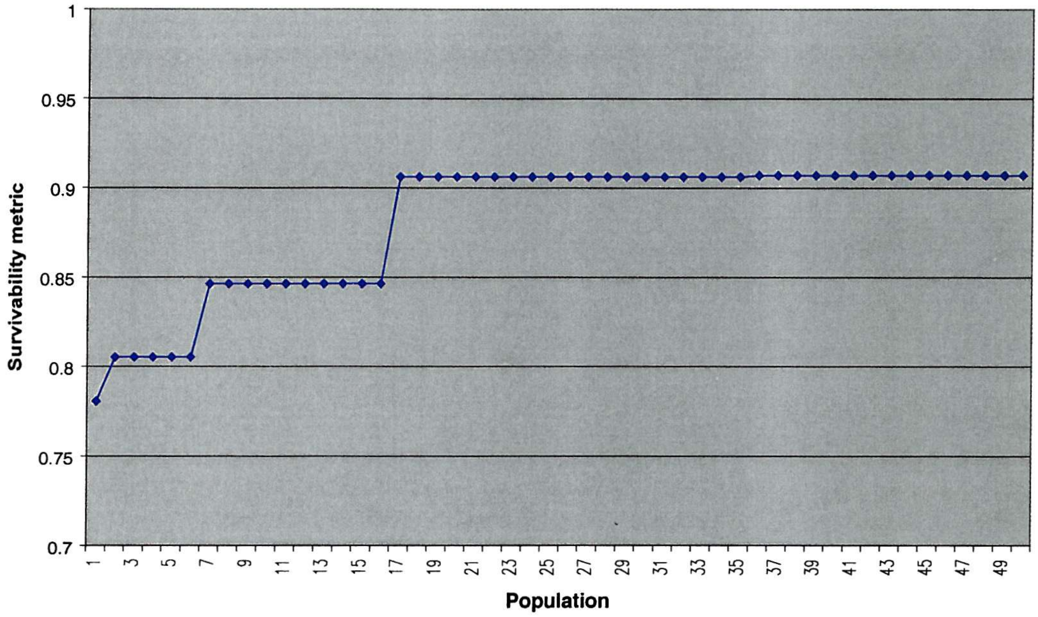


Figure 6-16 Plot of metric of most survivable design in each genetic population

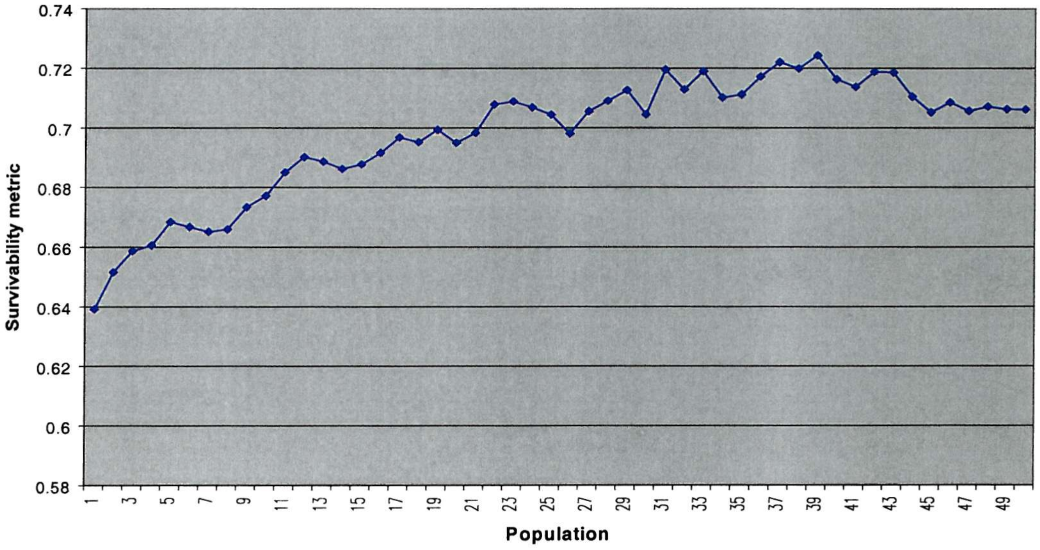


Figure 6-17 Plot of mean survivability metric of designs in each genetic population

Another interesting feature of the graph is the slight drop in metric value over the last 10 populations. This indicates that the populations have lost some 'good' schemata during the genetic algorithm's selection, crossover, and mutation process (see Section 4.2.2). The result suggests that there may be scope to refine the genetic algorithm, for example by modifying parameters such as the probability of crossover or mutation. Such an investigation is a necessary future activity, but falls outside the immediate objectives of the Ph.D. research programme.

To confirm that the genetic algorithm method is more efficient than a random search approach, the SHIELD model was used to produce 50 populations of randomly generated designs. The survivability metric of the 'best' design in each population is plotted in Figure 6-18, together with a plot of the overall 'best' design as the simulation proceeded. It is fairly evident when comparing this plot with Figure 6-16 that the genetic algorithm has a far superior performance. The 'best' design identified using the random process has a survivability metric of 0.7949, which is somewhat lower than the metric of the 'best' genetically derived design (i.e. 0.9066). Furthermore, the random process did not exhibit the rapid increase in metric value during the early populations, as was the case with the genetic algorithm.

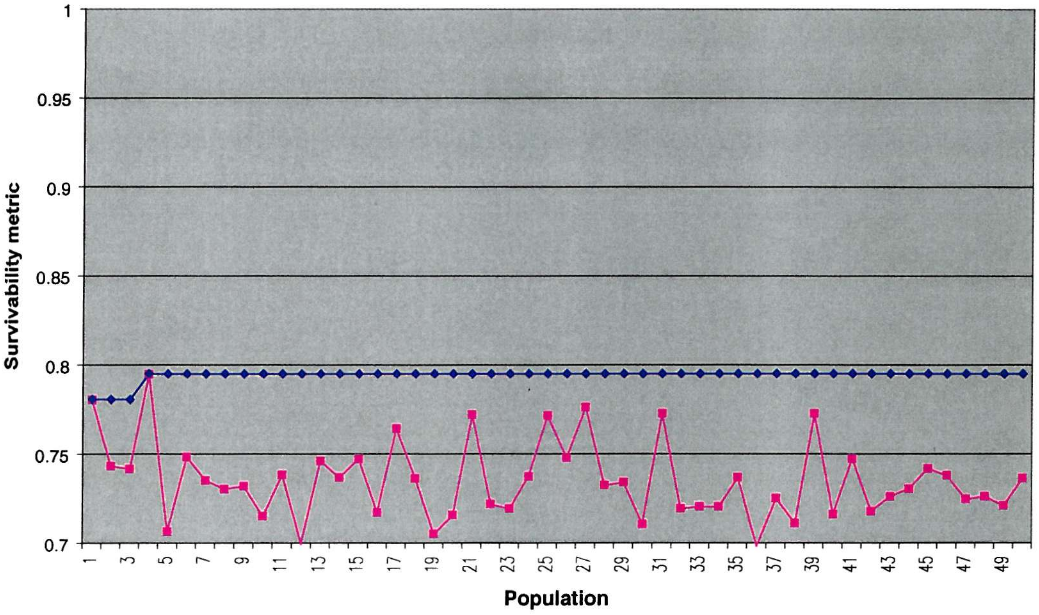


Figure 6-18 Plot of metric of most survivable design in each random population

6.4.2 Constrained Search - Box Satellite with Ten Components

In the previous section, the genetic algorithm in SHIELD was used to search for a cost-effective and survivable internal arrangement for the 10-component box satellite. With the mass balance constraint switched off, the resulting 'best' solution displayed a very 'top heavy' arrangement, which undoubtedly would prove to be problematic to implement in practice. The inclusion of a mass balance constraint has already been investigated for a randomly generated population of 100 designs in Section 6.3.2. The next logical step therefore is to see how the mass balance constraint influences the genetic algorithm search for the 'best solution'. Thus the simulation parameters of Tables 6-10 and 6-11 are retained here. Equal weighting factors,

W_0 and W_1, are also assumed in Equation 5-54, as before. The results of the search are illustrated in Figures 6-19 to 6-22.

Figure 6-19 shows how the survivability metric gradually increases from a value of 0.7849 in the initial population to 0.9003 in the final population. The general features observed for the unconstrained search in Figure 6-16 are also seen here. However, the plot now shows more structure, which is indicative of the additional complexity introduced by the mass balance constraint.

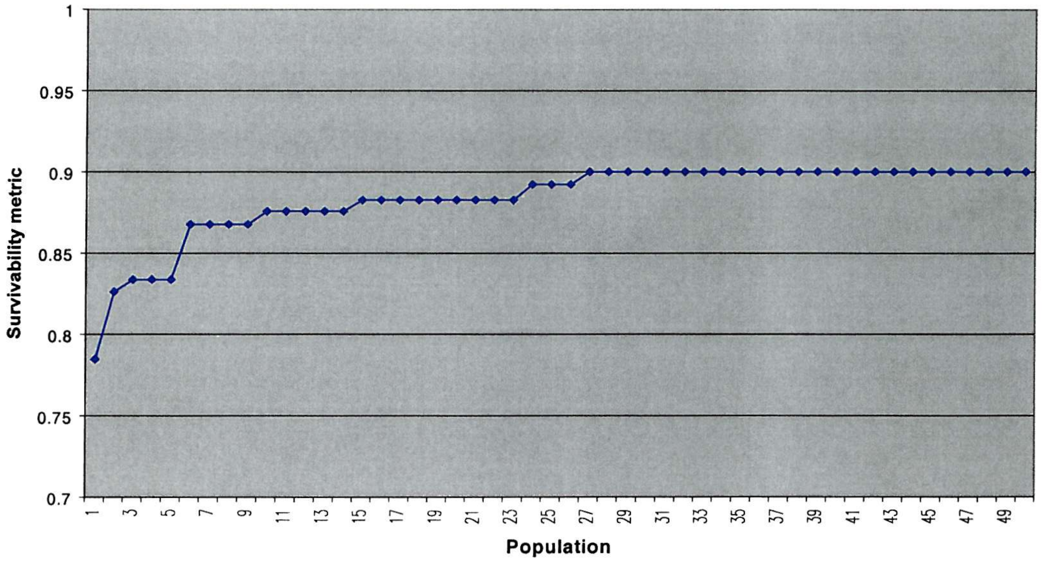


Figure 6-19 Plot of metric of most survivable design in each genetic population (when mass balance constraint is introduced)

Figure 6-20 illustrates how the failure probability of the highest metric design evolves during the search. As expected, one can see a gradual reduction in failure probability, until a value of 0.206 is reached. Interestingly, the lowest failure probability on the plot has a value of 0.137. However, this is achieved at the expense of a poorer mass balance, and so the corresponding metric is not quite as high (see Figure 6-19). This need for a compromise between mass balance and survivability also explains why the lowest failure probability found here is much higher than that obtained for the unconstrained case in Figure 6-13.

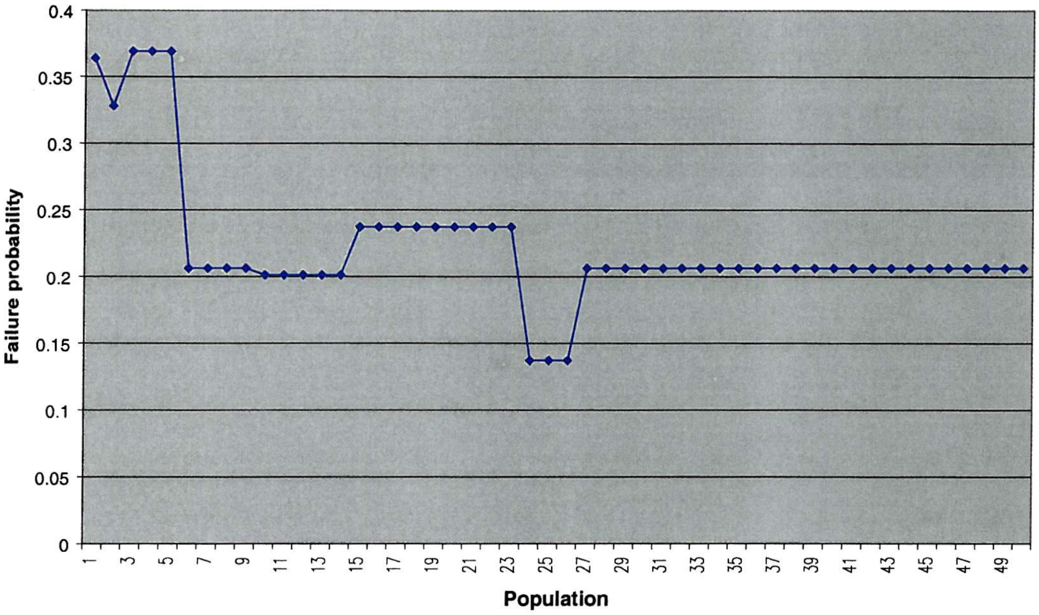


Figure 6-20 Plot of failure probability of most survivable design in each genetic population (when mass balance constraint is introduced)

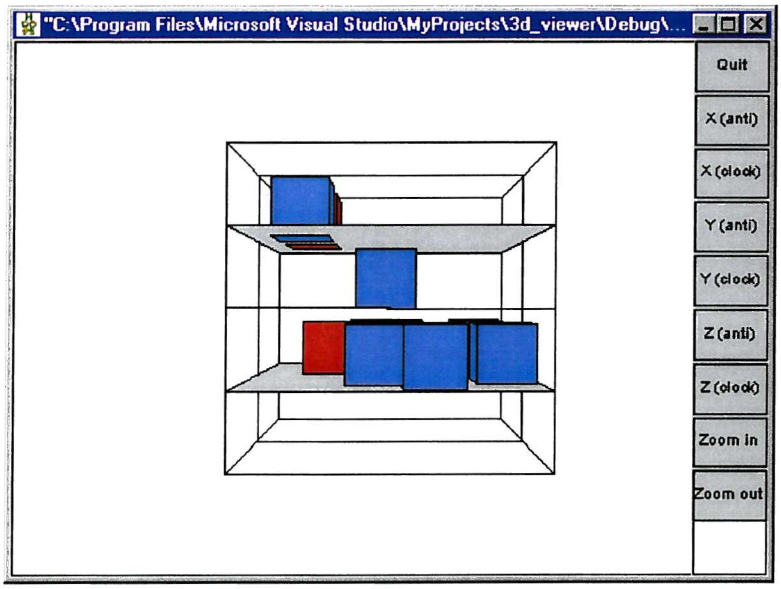

"C:\Program Files\Microsoft Visual Studio\MyProjects\3d_viewer\Debug\...
Quit
X (anti)
X (clock)
Y (anti)
Y (clock)
Z (anti)
Z (clock)
Zoom in
Zoom out

Figure 6-21 Highest survivability configuration in final genetic population (when mass balance constraint is introduced) – view of front face

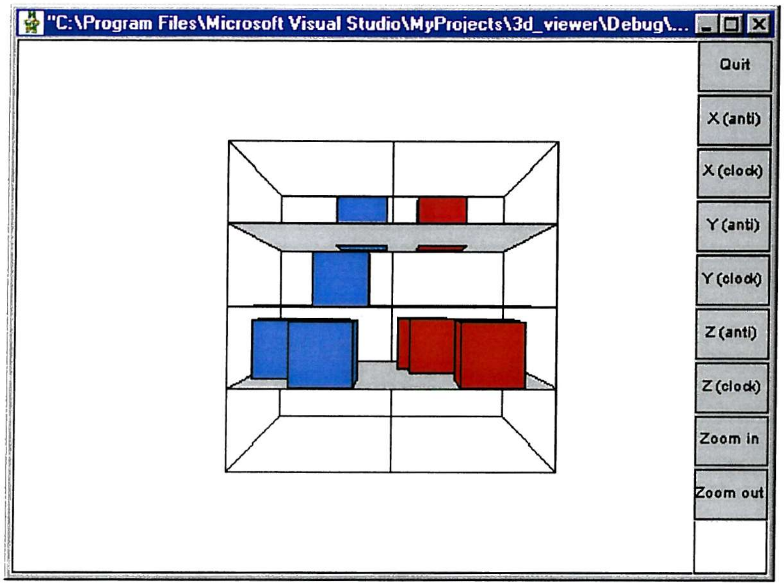

"C:\Program Files\Microsoft Visual Studio\MyProjects\3d_viewer\Debug\...
Quit
X (anti)
X (clock)
Y (anti)
Y (clock)
Z (anti)
Z (clock)
Zoom in
Zoom out

Figure 6-22 Highest survivability configuration in final genetic population (when mass balance constraint is introduced) – view of port face

To appreciate the effect of this compromise, Figures 6-21 and 6-22 show front and port side views of the internal arrangement of the design with the highest survivability metric. One can see the familiar placement of non-critical units between the vulnerable front face and the critical units. However, unlike the unconstrained case shown in Figures 6-14 and 6-15, the units are now distributed with a much better overall mass balance.

6.4.3 Constrained Search – Shielded Box Satellite with Ten Components

The genetic algorithm searches in the previous two sections have focused exclusively on the problem of finding the most survivable, cost-effective configuration for a 10-component box satellite. The possibility of increasing the thickness of the satellite body faces, either as an alternative or a supplementary strategy, has not yet been considered. This level of variability, where the satellite's internal arrangement is simultaneously considered with its structure design, offers a much greater challenge to the genetic algorithm (and indeed a spacecraft designer). So, its demonstration here is one of the most important results.

The set-up parameters for the satellite and the genetic algorithm are as described for the constrained optimisation in the previous section. However, the difference now is that the thickness of each of the satellite body faces is allowed to vary between 0.2 and 0.4 cm in discrete intervals. That is, the thickness can take the following values: 0.2, 0.267, 0.333, and

0.4 cm. Obviously, the presence of these different thicknesses on the satellite faces will result in different numbers of penetrations, as shown in Table 6-12.

Simulation	Front	Starboard	Port	Back	Space	Earth
# impacts (> 0.2 mm)	10.470	2.696	2.754	0.039	0.181	0.185
# penetrations (2 mm face)	2.903	0.225	0.202	<0.001	0.001	0.001
# penetrations (2.67 mm face)	1.300	0.131	0.113	<0.001	0.001	0.001
# penetrations (3.33 mm face)	0.860	0.065	0.067	<0.001	0.001	0.001
# penetrations (4 mm face)	0.554	0.040	0.054	<0.001	<0.001	0.001

Table 6-12 Expected numbers of impacts and penetrations on a box satellite with varying body face thicknesses

The results of the search are illustrated in Figures 6-23 to 6-26. Because of the increased complexity of this search, the genetic algorithm is now run through 100 populations, instead of 50. Figure 6-23 reveals the customary rapid increase in survivability metric during the early populations, followed by a longer phase where improvements are much more difficult to find. The highest survivability design has a metric value of 0.9573, which is somewhat higher than the value of 0.9003 observed for the previous search. So, the addition of shielding mass to the satellite body faces is beneficial in this case, i.e. the extra risk reduction outweighs the costs. Therefore, this satellite benefits from a protection strategy where physical shielding and the internal arrangement of equipment are considered concurrently.

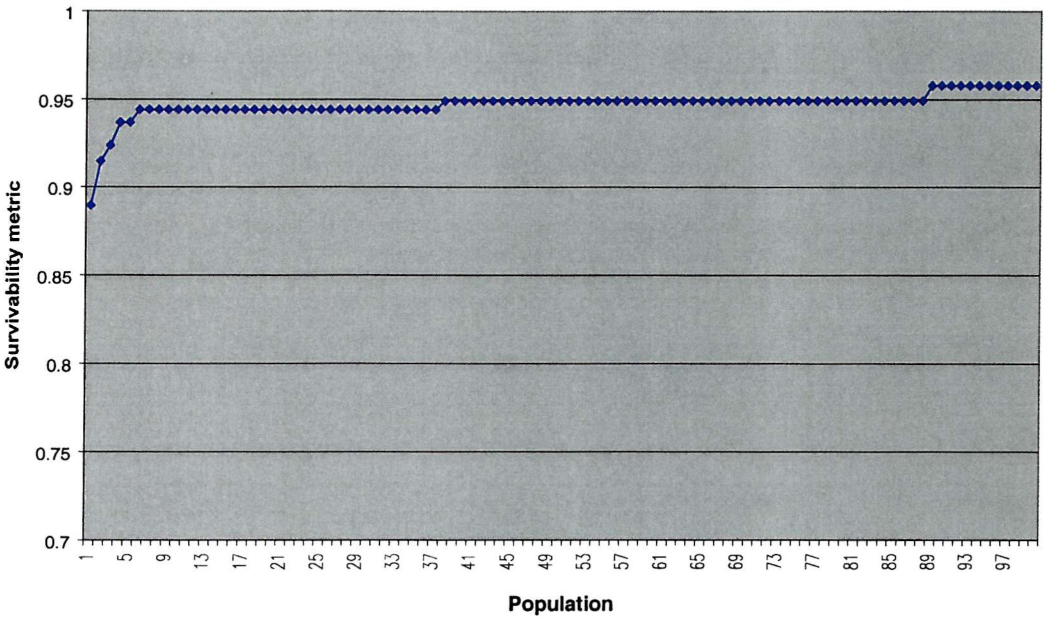


Figure 6-23 Plot of metric of most survivable design in each genetic population (when the mass balance constraint and the variable satellite body shielding are introduced)

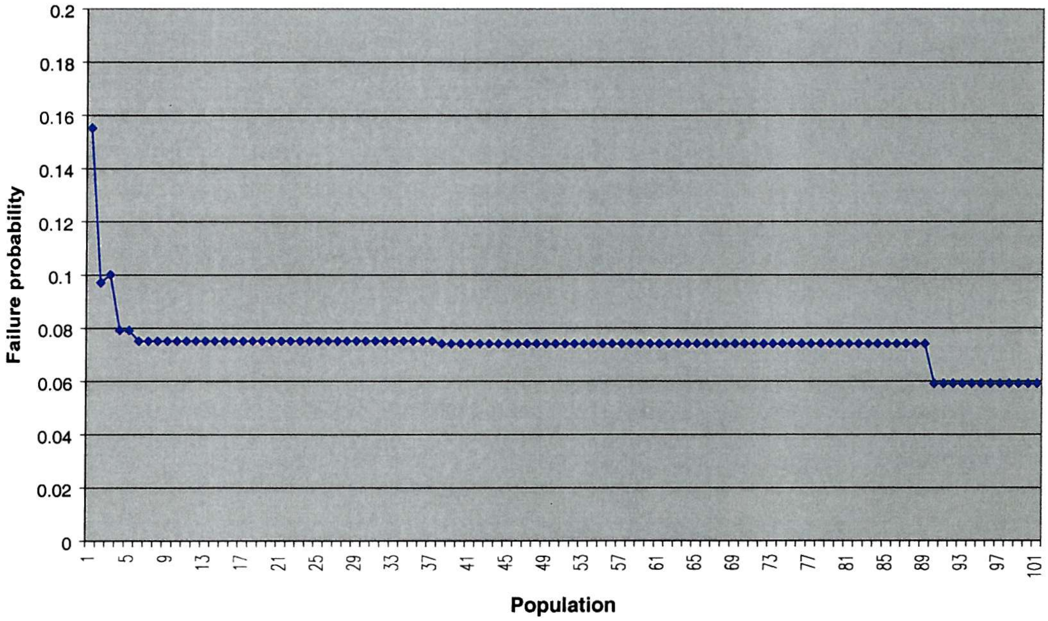
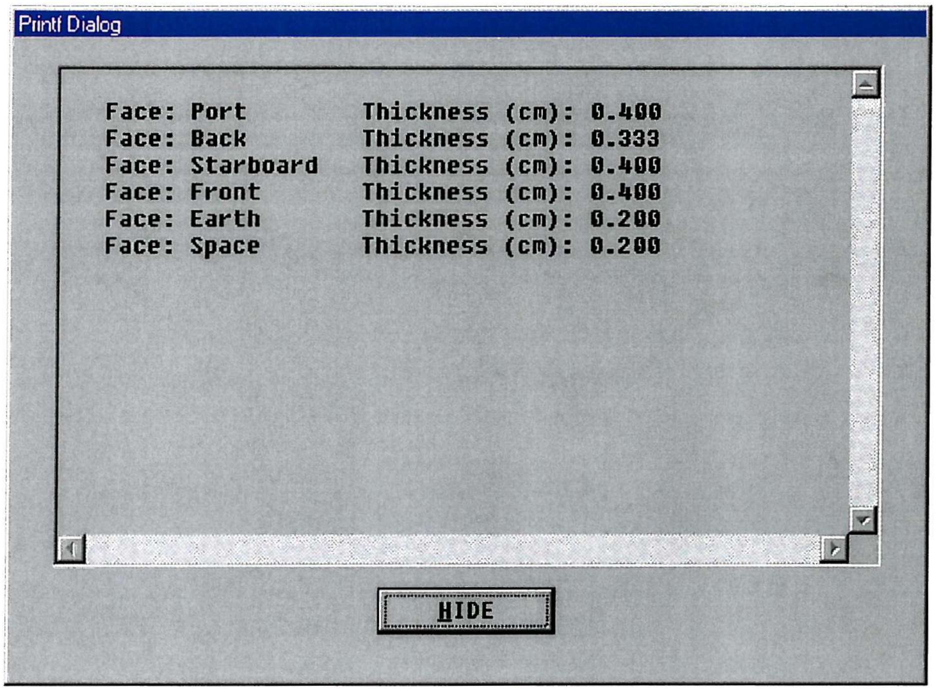


Figure 6-24 Plot of failure probability of most survivable design in each genetic population (when the mass balance constraint and the variable satellite body shielding are introduced)

The reduction in failure probability throughout the search is evident in Figure 6-24. One can see that the 'best' design has a value of 0.059, which is lower than the failure probabilities observed for either of the two previous searches. This is because of the additional shielding mass on the satellite body faces, which reduces the probability of penetration and thus the failure probability. Confirmation of this is given in Figure 6-25, which shows a printout of the thickness of each body face on the satellite



```
Face: Port        Thickness (cm): 0.400
Face: Back        Thickness (cm): 0.333
Face: Starboard   Thickness (cm): 0.400
Face: Front       Thickness (cm): 0.400
Face: Earth       Thickness (cm): 0.200
Face: Space       Thickness (cm): 0.200
```

Figure 6-25   Printout of body face thicknesses of most survivable design

Reassuringly, the genetic algorithm has identified that the most vulnerable faces, i.e. on the front and sides, will benefit from additional shielding, which should not be surprising given the penetration numbers in Table 6-12. Similarly, it has found that no further shielding is required for the space and Earth faces, since these are much less vulnerable to penetration. There is however one face where the correct amount of shielding has not been correctly identified during the simulation. The back face contains some additional shielding, whereas Table 6-12 reveals that none is needed. This is a clear indication that, although it has found a very good design, the genetic algorithm has not succeeded in reaching the global optimum solution within the 100-population search. A longer search may have produced an improved solution, however this would be at the expense of unacceptably long run-times. As mentioned previously, hybridisation of the genetic algorithm is an option that may need to be considered in the future to overcome this issue.

Having looked at the distribution of shielding on the satellite body, it is interesting now to see how the internal equipment has been arranged (see Figure 6-26).

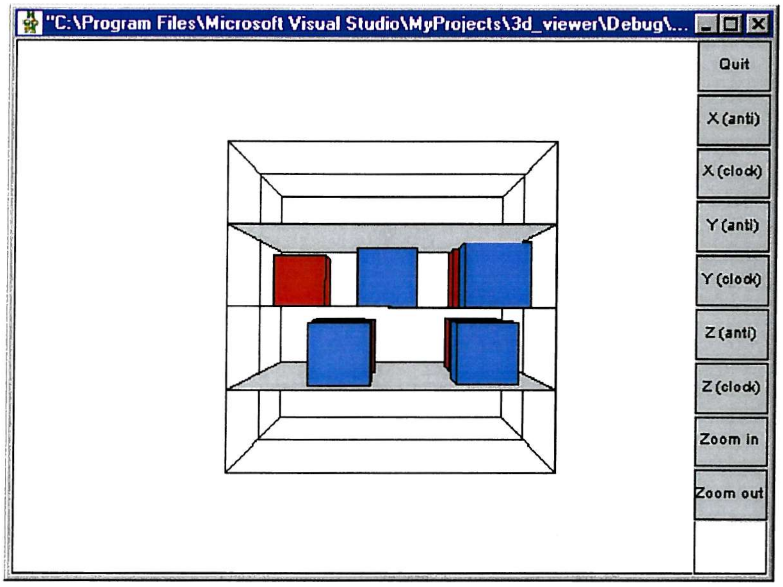

"C:\Program Files\Microsoft Visual Studio\MyProjects\3d_viewer\Debug\...
Quit
X (anti)
X (clock)
Y (anti)
Y (clock)
Z (anti)
Z (clock)
Zoom in
Zoom out

Figure 6-26 Highest survivability configuration in final genetic population (when the mass balance constraint and the variable satellite body shielding are introduced) – view of front face

Once again, the equipment has been distributed evenly, in accordance with the requirements for ensuring mass balance. Critical equipment is also reasonably well protected from the front face by non-critical units. Even with additional shielding, the front face is still the most vulnerable part of the satellite, and therefore dominates the internal arrangement. So, the result is entirely consistent with what one might have expected. However, one critical unit is still fairly exposed to the front face, which suggests that the optimum configuration has not been found. Again, this is symptomatic of the genetic algorithm's difficulty in converging on the global optimum solution.

6.5 Identification of Satellite Failure Probability versus Debris Size

The next result from SHIELD concerns the identification of a relationship between the size of a debris particle impacting a satellite and the probability that the satellite will fail. Such a relationship is a key element of long-term environment models, e.g. IDES and EVOLVE, which are often used to predict the risk and replenishment strategy for satellite constellation systems. As mentioned in previous discussions, this is an important result, which has never been derived before.

It is likely that the relationship will be highly dependent on the design of the satellite, and so one should expect significant variation. To examine a wide range of spacecraft designs, from nano/microsatellites up to the largest satellites, is a sizeable undertaking. Therefore, as an illustration, the assessment is limited to just one basic satellite design. In keeping with the previous analyses, the 10-component box satellite defined in Section 6.3.2 is used.

Despite this restriction, it is still possible to look at a wide variety of internal architectures for the satellite, by utilising SHIELD's in-built facility for creating these at random. Thus, it will be possible to derive a general relationship for this particular satellite.

Essentially, the SHIELD simulation here is almost identical to that conducted in Section 6.3.2. That is, 100 different configurations of the satellite are generated at random, and each is assessed to determine its probability of impact, penetration and failure. From these, the mean probabilities of impact, penetration and failure of the 100 designs are calculated, respectively. The main difference with the assessment in Section 6.3.2 is that instead of exposing the satellite to the entire range of debris particle sizes, i.e. greater than 0.2 mm, the simulation here is restricted to specific logarithmic size ranges. The results are summarised in the first seven columns of Table 6-13. It is evident that because the numbers of sub-millimetre particles are very much higher, then they are much more likely to penetrate the satellite and (despite their small size) cause the satellite to fail. Particles larger than ~2 mm would seem to be less of a concern to the satellite due to their much smaller fluxes.

The most interesting results are listed in the final two columns of Table 6-13. These are the required conditional probabilities. That is, the probability that the satellite will be penetrated given an impactor of a certain size, and the probability that it will fail given the same impactor. For clarity, the results are also displayed graphically in Figure 6-27. As expected, the larger an impacting particle, the more likely it is to penetrate, such that if its diameter exceeds ~1 mm it is almost certain to penetrate. Similarly, the larger an impacting particle is, the more likely it is to cause the satellite to fail. There is an ~2% chance that any given impactor, whose diameter is between 0.215 and 0.464 mm, will terminate the mission. This rises to 20% for an impactor whose diameter is between 4.64 mm and 21.5 mm.

Particle size range (mm)	No. of impacts ($N_{impacts}$)	Probability of impact (P_{impact})	Mean no. of penetrators (N_{pen})	Mean probability of penetration (P_{pen})	Mean no. of satellite failures ($N_{failures}$)	Mean probability of satellite failure ($P_{failure}$)	Mean prob. of penetration given an impact ($P_{pen \mid impact}$)	Mean prob. of failure given an impact ($P_{failure \mid impact}$)
0.215 – 0.464	13.7965	0.999999	1.7830	0.8319	0.2995	0.2588	0.1292	0.0217
0.464 – 1.000	1.5925	0.7966	1.3336	0.7365	0.2318	0.2069	0.8374	0.1456
1.000 – 2.150	0.2704	0.2369	0.2623	0.2307	0.0470	0.0459	0.9700	0.1738
2.150 – 4.640	0.0446	0.0436	0.0442	0.0433	0.0080	0.0080	0.9910	0.1794
4.640 – 10.000	0.0090	0.0089	0.0090	0.0089	0.0018	0.0018	1	0.2000
10.000 – 21.500	0.0030	0.0030	0.0030	0.0030	0.0006	0.0006	1	0.2000

Table 6-13 Relationship between mean satellite failure probability and particle size (for 100 randomly generated internal configurations of the design specified in Table 6-10)

On the face of it this last result is a little surprising, since one would expect a much higher failure probability from such a large impactor. However, the result is consistent with what should be expected from the current internal damage assessment methodology. The inclusion of a debris cloud distribution function, as discussed in Chapter 5, would undoubtedly produce a different result, and perhaps give higher failure probabilities.

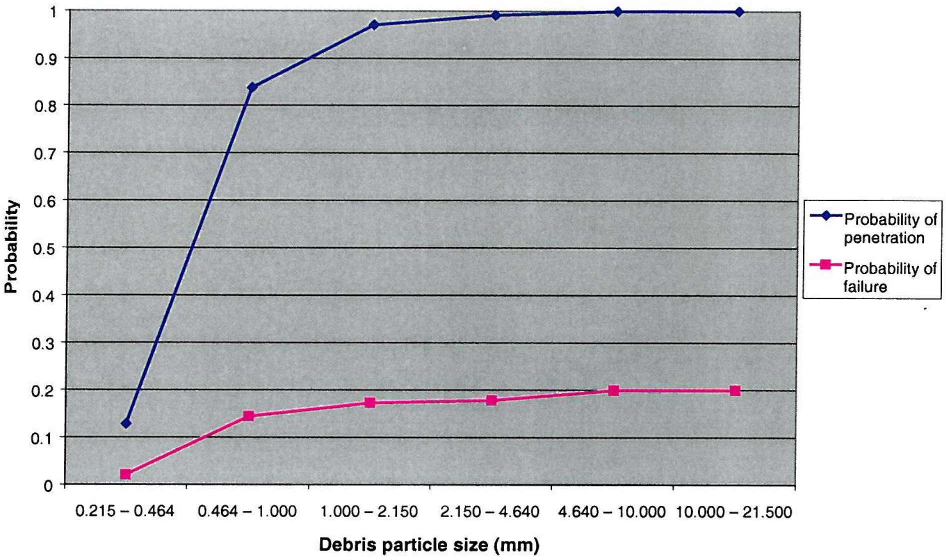


Figure 6-27 Mean probability of penetration and failure of a specific spacecraft design versus debris impactor size

It is worth emphasising that the above relationship between mean failure probability and impactor size is specific to one particular spacecraft design. Although not yet verified, it is expected that different relationships would be produced by varying factors such as:

- ❑ The design of the body spacecraft structure panels
- ❑ The total number of components inside the spacecraft
- ❑ The values of criticality assigned to individual components
- ❑ The 'packing density' of equipment inside the spacecraft

This latter point is particularly worthy of future investigation. Yakovlev *et al* (2001) have determined that there is a non-linear relationship between packing density and the size of a spacecraft. As the spacecraft dimensions change from ~200 cm to ~10 cm the density is observed to grow by an order of magnitude from ~100 kgm^{-3} to ~1000 kgm^{-3}. Thus, in a large spacecraft the equipment will be more spread out, and therefore the interaction of penetrating particles inside the spacecraft is likely to be different to that of a small spacecraft.

6.6 Survivability Evaluation of the METOP Satellite's Service Module

In this final result from the SHIELD model, the survivability of the Service Module on the METOP satellite is evaluated. The result is provided as an illustration of how SHIELD might be applied to the assessment of an actual satellite, rather than the idealistic designs considered so far. METOP was chosen, as this satellite was the subject of an ESA funded study into cost-effective debris shielding to which the author contributed [Turner *et al* (1999)]. Therefore, it is instructive to compare the results derived here with those from the study.

6.6.1 METOP Design

As discussed previously, METOP is a meteorological satellite that will be launched into an 800 km circular, polar orbit for a period of four years from 2005. It comprises a Service Module to house the main subsystem equipment, a Payload Module that contains the payload instruments, and a solar array. The satellite is three-axis stabilised, with one side always facing the velocity direction, and another pointing towards the Earth. The overall configuration of the satellite is shown in Figure 6-28.

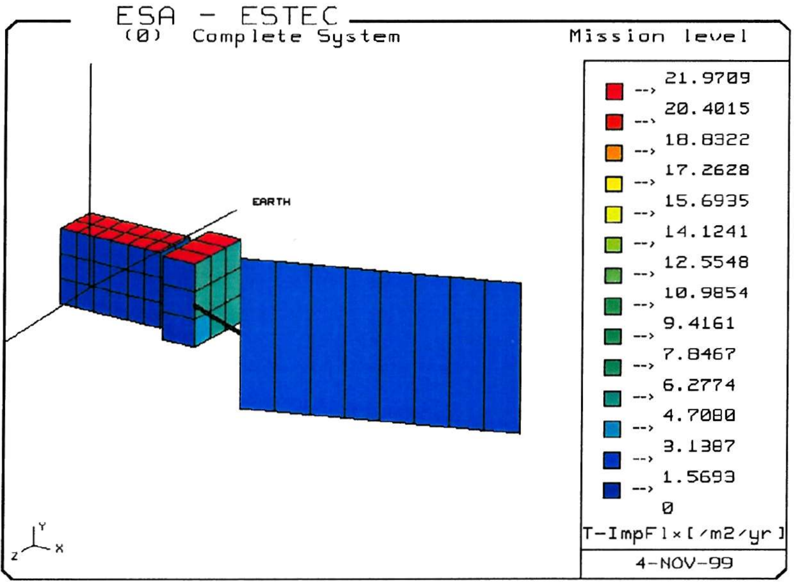


Figure 6-28 ESABASE generated representation of the METOP satellite [Wilkinson & Stokes (1999)]

The design and layout of equipment on the Service Module has already been discussed briefly in Section 4.1. As a precursor to the SHIELD simulation it was necessary to construct the Service Module geometry and identify the precise size, location and orientation of the individual items of equipment. These were obtained from a set of engineering drawings provided during the ESA study. In the absence of any details on the construction of the equipment, an assumption is made that all units are protected by 2mm thick aluminium walls, which is typical for radiation shielding purposes. The structure of the Service Module itself is assumed to comprise 0.4 mm thick aluminium panels sandwiching an aluminium honeycomb core. The overall thickness of the panels is assumed to be 4 cm. Figures 6-29a to 6-29d illustrate four views of the Service Module, as constructed using the SHIELD model. Figure 6-29a can be compared to the schematic diagram in Figure 4-1. The central cylinder contains batteries and propellant tanks that, although not specifically shown, are considered mission-critical.

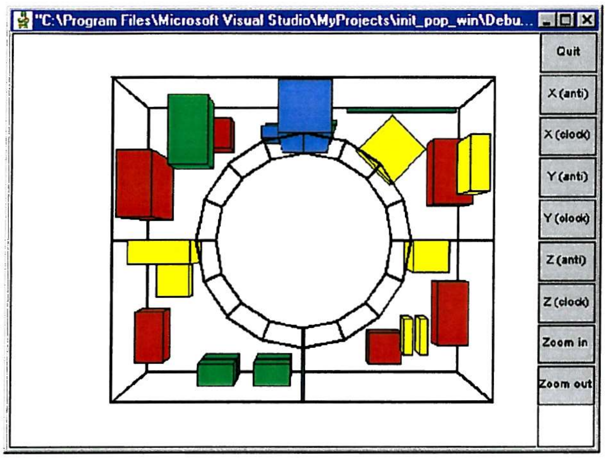

"C:\Program Files\Microsoft Visual Studio\MyProjects\init_pop_win\Debu...
Quit
X (anti)
X (clock)
Y (anti)
Y (clock)
Z (anti)
Z (clock)
Zoom in
Zoom out

Figure 6-29a METOP Service Module - view of starboard face

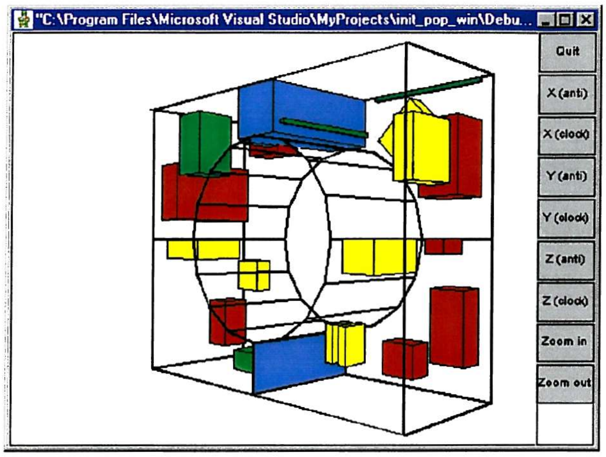

"C:\Program Files\Microsoft Visual Studio\MyProjects\init_pop_win\Debu...
Quit
X (anti)
X (clock)
Y (anti)
Y (clock)
Z (anti)
Z (clock)
Zoom in
Zoom out

Figure 6-29b METOP Service Module - view of starboard and front faces

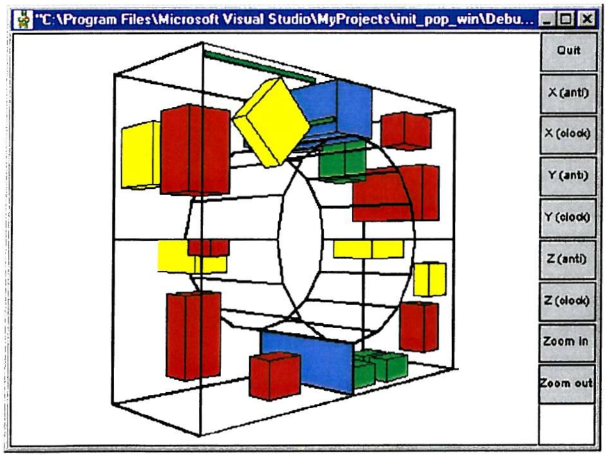

"C:\Program Files\Microsoft Visual Studio\MyProjects\init_pop_win\Debu...
Quit
X (anti)
X (clock)
Y (anti)
Y (clock)
Z (anti)
Z (clock)
Zoom in
Zoom out

Figure 6-29c METOP Service Module - view of port and front faces

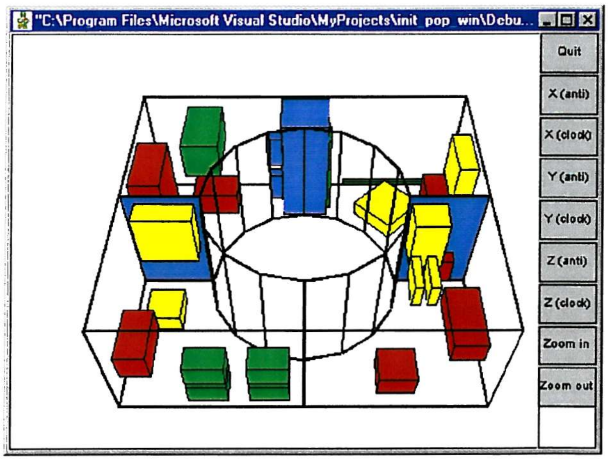

"C:\Program Files\Microsoft Visual Studio\MyProjects\init_pop_win\Debu...
Quit
X (anti)
X (clock)
Y (anti)
Y (clock)
Z (anti)
Z (clock)
Zoom in
Zoom out

Figure 6-29d METOP Service Module - view of starboard and Earth faces

Comp	Redundancy	Criticality	Comments on criticality
CCU	All CCU functions are internally redundant. Active redundant mass memory with marginal redundancy. Watchdog allows the EDR to automatically reconfigure the CCU if the signal is not refreshed.	0.9	**Critical**, since CCU contains the back-up software for data interface control during Safe Mode.
BSP	Not specified in available documentation, therefore assume none.	0.7	Critical at start of mission, but less important as the mission progresses.
OBA	Not specified in available documentation, therefore assume none.	0.9	No information available, therefore assume critical.
EDR	Internal redundancy of all functions, but not distributed.	0.9	**Critical**, due to high level of authority and control of Safe Mode.
TRSP	2 completely redundant units but with only very small separation.	0.7	**Critical**, since insufficient separation of the redundant units could allow both to be lost from a single penetration.
EAIM	Internal redundancy provides 4 combined drive chains allowing the capability for 1 nominal and 1 redundant for each of the Magnetorquers and 1 spare channel for the control of any of the 3 Reaction Wheels.	0.9	**Critical**, since loss of this unit will result in a complete loss of the attitude pointing control.
RW Y	Electrical part is redundant. Mechanical bearings are not redundant but they are not used during routine operation (since electro-mag suspension is used).	0.7	Loss of wheel implies a loss of pointing control. Magnetorquers will provide degree of redundancy.
RW X	Electrical part is redundant. Mechanical bearings are not redundant but they are not used during routine operation (since electro-mag suspension is used).	0.7	Loss of wheel implies a loss of pointing control. Magnetorquers will provide degree of redundancy.
MAC Y	Redundant windings on the torque rod are provided.	0.4	Loss of magnetorquer implies a loss of pointing control. Reaction wheels will provide degree of redundancy.
MAC X	Redundant windings on the torque rod are provided.	0.4	Loss of magnetorquer implies a loss of pointing control. Reaction wheels will provide degree of redundancy.
EPRM	Not specified in available documentation, therefore assume none.	0.9	**Critical**
EIU	Fuses and current limiting are used inside the unit to provide protection to the satellite against catastrophic electrical faults.	0.4	Critical at the beginning of the mission, but not in nominal operations.
RSJD	Low level active redundant units with marginal redundancies and majority voting; very small physical separation of nominal and redundant units.	0.9	**Critical**, since this unit controls other units which are critical to satellite survivability.
MEGS-E	There are two separate MEGS units mounted on the SADM bracket shear wall.	0.2	Not critical, since there is deemed sufficient separation between the redundant units.
RW Z	Electrical part is redundant. Mechanical bearings are not redundant but they are not used during routine operation (since electro-mag suspension is used).	0.7	Loss of wheel implies a loss of pointing control. Magnetorquers will provide degree of redundancy.
BMG	2 are used for nominal operations, the other 2 are preselected to support the Safe Mode function.	0.7	**Critical**, since there is insufficient separation of the redundant gyros.
T4S	Not specified in available documentation, therefore assume none.	0.9	**Critical**, since it handles all attitude data for all modes of operation.
BEG	Each BEG provides electrical support to 1 gyro, but only 2 are used during nominal operation.	0.4	Not critical, since the orientation of the 4 gyros provides adequate separation of the redundant units.
Cylinder	Contains five batteries and four propellant tanks. Implies limited redundancy.	0.9	**Critical**, due to large amount of stored energy.

Table 6-14 Criticality values of METOP Service Module equipment (see Table 4-1 for component descriptions) [Turner et al (1999)]

The identification of the criticality of each equipment unit for the mission is an important input for the simulation. Following the four categories of criticality listed in Table 4-6, and applying engineering judgement, the equipment was assigned criticality values as shown in Table 6-14. In Figures 6-29a to 6-29d, the most critical units are identified in red (i.e. criticality > 0.75), followed by yellow, green, and finally blue (i.e. criticality < 0.25).

6.6.2 Survivability Evaluation

As with the previous studies in this chapter, the debris environment directional fluxes generated by IDES are input into SHIELD to simulate the distribution of impacting particles on the body of the Service Module. However, in this case, the fluxes are scaled up to avoid the flux under-prediction problem (discussed in Section 6.2.5). Table 6-15 summarises the distribution of impacts and penetrations on the Service Module derived using SHIELD. It should be noted that the simulation only considers the Service Module by itself, i.e. any shadowing effects or secondary ejecta effects from the Payload Module or solar array are ignored (for reasons of simplicity). For example, it is likely that the Payload Module will in reality provide some protection against debris impacting the starboard face of the Service Module. To address these issues, a more comprehensive SHIELD assessment would be required, in which the integrated spacecraft is analysed.

Comparing the total number of penetrations (i.e. 1.06) with the result from the ESA study, which gave a value of 1.21, one can see that there is reasonable agreement. Not surprisingly, the front face experiences the greatest number of impacts and penetrations, followed by the starboard and port faces.

Face	No. of impacts (> 0.2 mm)	No. of penetrations	Probability of penetration
Front	2.49	0.82	0.56
Starboard	1.39	0.12	0.11
Port	1.39	0.12	0.11
Rear	0.01	<0.0002	<0.0002
Earth	0.05	0.0002	0.0002
Space	0.04	<0.0002	<0.0002
TOTAL	5.37	1.06	0.65

Table 6-15 Distribution of impacts and penetrations on METOP Service Module

The most interesting result from SHIELD is of course the determination of the probability of failure of the Service Module, and the identification of the equipment items that are the greatest contributors to mission failure. These results are summarised in Table 6-16

Comp.	Criticality	No. of mission failures caused by component	% contribution to mission failure
CCU	0.9	0.0290	11.5
BSP	0.7	0.0394	15.6
OBA	0.9	0.0104	4.1
EDR	0.9	0.0212	8.4
TRSP	0.7	0.0188	7.5
EAIM	0.9	0.0384	15.2
RW Y	0.7	0.0208	8.3
RW X	0.7	0.0310	12.3
MAC Y	0.4	0.0006	0.25
MAC X	0.4	0.0072	2.9
EPRM	0.9	0.0016	0.6
EIU	0.4	0.0030	1.2
RSJD	0.9	0.0016	0.6
MEGS-E	0.2	0.0000	0.0
RW Z	0.7	0.0014	0.6
BMG	0.7	0.0006	0.25
T4S	0.9	0.0008	0.3
BEG	0.4	0.0040	1.6
Cylinder	0.9	0.0222	8.8
TOTAL	-	0.2520	100

Table 6-16 Contribution of METOP Service Module equipment to mission failure (see Table 4-1 for component descriptions)

The total number of mission failures is 0.252, which (from Equation 3-12) is equivalent to a failure probability of 0.22 (i.e. 22%). These values are considerably lower than the total number of penetrations and probability of penetration in Table 6-15, thus underlining the notion that penetration does not necessarily equal failure for an unmanned spacecraft. However, it is worth pointing out that the debris-induced failure probability is still significant relative to the typical failure probabilities observed in standard satellite reliability assessments, which are derived primarily by considering random and wear-out failures of components. This suggests two conclusions, which are also in broad agreement with those from the ESA funded study of METOP:

1. The assessment of debris-induced failure probability should be considered as part of the normal risk assessment process for typical unmanned spacecraft.
2. Consideration should be given to enhancing the Service Module's survivability by incorporating extra protection.

The equipment items that are the greatest contributors to mission failure are contained in the two quadrants nearest the vulnerable front face of the Service Module. These are the first eight items listed in Table 6-16. All of the items have high criticality values (i.e. 0.7 or higher), which explains why they dominate the mission failure probability. Another conclusion from the assessment, therefore, is that consideration should be given to redesigning the arrangement of equipment inside the Service Module. The results show that the obvious place for critical items is in the two quadrants near the rear face.

Finally, it is worth commenting that the propellant tanks and batteries in the central cylinder also have a notable contribution to mission failure, despite being well protected by the cylindrical structure. This is because both items are large, have high criticality values assigned to them due to their large amounts of stored energy, and are relatively exposed to the debris fluxes on the Service Module side faces. Again, some additional shielding might be beneficial here.

6.7 Discussion

The results presented in this chapter illustrate the main capabilities of the SHIELD software model. They also provide confidence that the software is operating in accordance with the designed functions. Specifically, it is worth highlighting the following:

1. Based on a comparison with ESABASE / DEBRIS and BUMPER, SHIELD has demonstrated its ability to accurately predict the numbers of impacts and penetrations on a satellite.
2. Using a methodology for assessing the damage inside an unmanned satellite (when penetrated by a debris particle), the software can determine the failure probability of the satellite.
3. Failure probability is shown to be a more meaningful measure of impact risk than penetration probability for an unmanned satellite.

4. Radically different protection strategies can be evaluated and compared using a survivability metric that combines the costs of protection with the associated failure risks.
5. Consideration of the internal arrangement of equipment is an extremely important means of improving the survivability of a satellite in a cost-effective manner.
6. The location of critical internal equipment away from vulnerable satellite faces, such as the front face, is highly advantageous particularly if non-critical units can be placed in-between.
7. The addition of physical shielding, by increasing the thickness of a satellite's body structure, can also be a cost-effective protection strategy when the probability of penetration is sufficiently high.
8. Physical shielding offers greatest benefits when applied to the most vulnerable satellite faces, typically the front and side faces. The addition of shielding to the back, space, and Earth faces is less likely to enhance a satellite's protection cost-effectively.
9. The choice of physical shielding should not be considered independently of the arrangement and vulnerability of the internal equipment.
10. The genetic algorithm method appears to be a satisfactory means of searching through many competing protection strategies for an unmanned satellite, to produce a cost-effective design solution.
11. The genetic algorithm implemented in SHIELD appears to be efficient in converging rapidly near the optimum solution. However, it has difficulty in focusing on the precise global optimum solution. This is a common trait exhibited by genetic algorithms, which can be overcome by hybridisation.
12. The SHIELD genetic algorithm has also demonstrated that it is possible to introduce design constraints such as mass balance into the search for an optimum protection solution.
13. One of the more interesting features of the software is its ability to identify the probability that a satellite will fail when impacted by a debris particle of a given size.
14. Although highly dependent on the internal damage assessment methodology built into SHIELD, there are indications that even a quite large impactor (>> 1mm) will not necessarily cause a satellite to fail.

The core element underpinning many of the above statements is the approach to satellite survivability evaluation, particularly in terms of characterising the internal satellite damage caused by penetrating impactors. As noted at the beginning of the chapter, the validation of this is not possible at the current time. However, it may eventually be possible to compare results with those from lethality models that have been developed by the defence sector. Typically, lethality models examine the effects of a single missile impact on a vehicle. At present, they are limited to low velocity impacts (< 3 km/s), but work is ongoing to upgrade to higher velocities. One such model is QinetiQ's INTAVAL [Trufitt (1999)]. This is a suite of programs, written in FORTRAN, for the assessment of the vulnerability of air-targets to gun-fired projectiles (shot and shell) and fragmenting warheads. Separate modules calculate the effects of penetration, blast, fragment shatter, fire and explosion, fuel loss, and structural kill. These modules are based on empirical algorithms derived from extensive trials data over the past 20 years. It is possible some time in the future that simple test cases could be defined to enable comparisons between SHIELD and INTAVAL.

Finally, the research has demonstrated the benefit of incorporating a genetic algorithm into the software to assist in the search for cost-effective protection solutions. In a project support context, this might be beneficial during the early phases of a satellite design, where there is scope for considering a range of options. Its most immediate benefit though is as a research aid. The ability to explore the parameter design space of a given problem is essential if general conclusions are to be reached. For example, a possible future study might look at the question of whether distributing redundancy will provide significant improvements to survivability. Without the genetic algorithm to assist, it would be very difficult to arrive at a meaningful answer. Recently, other researchers have also begun to recognise the benefits of a genetic algorithm. Reimerdes & Wohlers (2001) have incorporated one into a code called MDPANTO. The basic software has very similar functionality to ESABASE / DEBRIS (see Section 3.5.3), but with the addition of the genetic algorithm it can now assist with the identification of optimal shielding on a spacecraft structure. However, the code has not yet been extended to optimise the layout and vulnerability of equipment on a spacecraft. For this capability, SHIELD is the only code known to exist.

7 Conclusions and Recommendations

7.1 Overview

Projections of the future debris population indicate that it is likely to almost double in the next 50 years if mitigation measures are not widely adopted. Popular orbits, such as 800 – 1000 km altitude polar orbit, and 1400 – 1500 km altitude polar orbit, will be particularly affected. Already, the spatial density of debris objects in these regions is sufficiently high that the impact risk to unmanned spacecraft can no longer be ignored. For example, a recent study of the collision risk to the METOP spacecraft, which will operate in an 800 km altitude polar orbit, revealed that it could expect to receive at least one penetrative impact during its mission life. Consequently, designers are now beginning to pay closer attention to this issue.

Typically, they perform a risk assessment using established damage prediction codes such as ESA's ESABASE / DEBRIS, coupled with environment models such as ORDEM96, to predict the distribution of debris impactors on a spacecraft during its life. The codes are also used to determine the distribution of penetrators by utilising empirically derived ballistic limit equations. This is usually sufficient information to ascertain the most vulnerable parts of the spacecraft, and to reach a judgement over whether some form of additional protection is required. For unmanned spacecraft, consideration is generally given to increasing the shielding of existing materials and structures. This may be through the addition of layers of high-strength materials such as Nextel and Kevlar to MLI-covered surfaces, or by modifying the design of honeycomb panels. The benefit of this type of protection does of course come at a price, namely increased mass and cost.

Another approach, which is less well developed, is to configure a spacecraft so that mission-critical equipment is protected from the most vulnerable regions, possibly by less critical neighbouring items. This approach provides protection enhancement without incurring significant mass penalty. However, there are two problems associated with it. First, for it to be effective a designer must be able to consider a variety of equipment arrangements in the earliest stages of a spacecraft design. Second, until now no tool has been available to quantify and compare the survivability of such different designs. The solution to the second problem requires a methodology that can compute the probability of failure of a satellite given a penetration distribution. This can then be incorporated into an automated search technique, which would provide a means to address the first problem. Both of these are very challenging and worthwhile research topics, and neither had been attempted at the outset of this Ph.D. programme.

7.2 Model Description

The high level of novelty suggested by these two topics required the development of a new software model, which was subsequently called SHIELD. In keeping with good engineering practice, SHIELD was designed and coded according to ESA's software engineering standards.

Inevitably, several functions in the model share common features with existing software tools such as ESABASE / DEBRIS. For example, both models enable a user to construct a geometrical representation of a satellite, including the material properties of surfaces, which can then be viewed on-screen in 3-D. The approach in SHIELD for achieving this is to build up all surfaces of the satellite (i.e. satellite body faces, internal walls, shelves, and equipment) using the mathematics of vector equations of planes.

Another common feature is the ability to simulate the debris impacts experienced by a satellite during its mission life. The principle input for this computation in SHIELD is a target-centred debris collision flux data set generated by the IDES model. From this, distributions of flux-azimuth-elevation and velocity-azimuth-elevation are constructed, and Poisson statistics utilised to generate a set of test particles. Each particle has a unique mass, speed, and direction of impact relative to a cross-sectional disk that intersects and bounds the satellite. To determine if and where a particle hits the satellite, a standard ray-trace approach is used. That is, using vector mathematics, the intersection of the particle's trajectory vector is calculated for each satellite surface plane until the precise intersection point is derived.

With a distribution of impactors computed in this fashion, one has the basis for a survivability assessment. Traditionally, this involves determining the satellite's probability of penetration. Knowing the properties of the particles and satellite surfaces, ballistic limit equations are used to calculate which particles penetrate, thus giving the required probability. In SHIELD, the calculation goes a step further by calculating the probability of failure of the satellite. This is one of the two most innovative aspects of the research, and sets SHIELD apart from other models.

Essentially SHIELD calculates the probability of failure by determining the internal damage caused by each of the penetrating particles distributed on the satellite. Any equipment that lies within the line of sight of a given particle's trajectory is considered vulnerable. Multi-wall ballistic limit equations are called up to ascertain the extent of penetration inside the satellite, and therefore identify exactly which units are damaged. Knowing the criticality of these units, the consequences for the mission are then easily derived. When all penetrators are considered in this fashion, the result is the desired satellite failure probability.

To derive a true measure of a satellite's survivability, SHIELD is able to combine this failure probability, or risk, with information on the various costs associated with the mission. A life cycle cost model has therefore been incorporated. Thus, it is possible to determine and compare the cost-effectiveness of radically different protection strategies in a completely objective manner.

This 'survivability metric' is also a key element of an automated protection search methodology, which represents the second innovative aspect of the research. A genetic algorithm was implemented for this purpose, since the principles of natural selection were considered well suited to search and optimisation problems of this type. In the case where there is significant scope for varying a satellite's protection strategy, usually in the early design stages, then the genetic algorithm can help designers focus on the most cost-effective solutions. Different levels of design variability may be examined, as pre-defined by the user. For example, one or more of the following can be investigated: structure and shield designs for the satellite body and internal units, and positioning of the internal shelves and units. Additionally, the search can be set to take account of various system design constraints that may be present, such as retaining mass balance.

Perhaps the most challenging aspect in the construction of the genetic algorithm approach was the conversion of 'real' satellite design information into genetic code, and then back again into a meaningful design. However, these concerns were alleviated to a large extent during the development of a 2-D prototype model, which was a precursor to SHIELD. The prototype also provided some additional assurance that a genetic algorithm would be sufficiently effective in identifying cost-effective protection solutions.

7.3 Model Accuracy

The development of any new model is only complete when it has passed through a series of validation runs, and SHIELD is no exception in this regard. Of course, demonstrating the accuracy of a model is not without its difficulties, particularly when the model is attempting to break new ground in several respects, as is the case with SHIELD. The challenge therefore is twofold. First, to validate those elements which can readily be compared with other models or data. Second, for the innovative aspects, to derive results that are at the very least credible and explainable.

For SHIELD, the most straightforward part of the model to validate was the impact and penetration distribution simulation. The existence of other established models such as ESA's ESABASE / DEBRIS and NASA's BUMPER provided an ideal opportunity for direct comparisons to be made. Furthermore, the problem was alleviated considerably by the existence of a set of benchmark simulations prescribed by the Inter Agency Space Debris Committee's Protection Working Group. Specifically, one of these benchmark cases defines a simple box-shaped spacecraft design, orbital and mission parameters, and the debris environment through which the spacecraft is to be flown. When input into SHIELD, the resulting impact and penetration distributions on the box spacecraft were found to be up to an order of magnitude lower than those derived from ESABASE / DEBRIS and BUMPER. On close investigation these discrepancies were shown to be a consequence of under-prediction in the small debris particle fluxes generated by the IDES debris environment model, as compared to the ORDEM96 environment model fluxes used in ESABASE / DEBRIS and BUMPER. Thus, the impact and penetration distributions from SHIELD were in fact consistent with those from the other models. To overcome the under-prediction problem in the short term, it will be necessary to apply flux scaling factors to the fluxes output from IDES. This relatively straightforward approach has been adopted by other deterministic models such as MASTER. In the longer term, it is anticipated that continued improvements to the modelling of small-size debris source terms in IDES will close the gap.

For the first innovative element of SHIELD, i.e. the evaluation of survivability, the possibility of comparisons with other models and data was not an option. Instead, it was necessary to gain confidence by applying the model to a series of example satellites with varying degrees of complexity and different protection strategies. The first such simulation illustrated how the satellite survivability methodology could be applied meaningfully to three radically different protection strategies, and enable a judgement as to the best solution. The second simulation focused on a more complex satellite representation to demonstrate the potential variation in satellite survivability simply by reconfiguring the satellite's internal architecture. Both simulations provided results that seem to be within the bounds of credibility.

To validate the second innovative element of SHIELD, i.e. the identification of cost-effective protection using a genetic algorithm, this too could only be approached by performing a set of confidence-building demonstrations. Three cases of a particular satellite representation were investigated: (i) an unconstrained satellite employing one protection strategy, (ii) a constrained satellite employing one protection strategy, and (iii) a constrained satellite employing two protection strategies. In all three cases, the genetic algorithm search was successful in identifying cost-effective protection solutions that were explainable and consistent with what one might have expected. Additionally, when compared with a random search approach, the genetic algorithm demonstrated much greater efficiency.

7.4 Implications for Evaluating Survivability of Satellites to Debris Impact

One of the main contentions of the research was that the survivability of an unmanned spacecraft should not be characterised exclusively by the assessment of probability of penetration, an approach which is usually more appropriate to the assessment of manned spacecraft. This led to the methodology for determining probability of failure instead, the idea being that there was a quantifiable probability that a satellite might survive a penetrative impact. For the satellite representations investigated, the SHIELD model clearly showed that substantial differences could exist between failure probability and penetration probability. Not surprisingly, failure probability was also shown to be highly dependent upon the number of critical equipment units, their design, and their relative arrangement in the satellite. For example, significant variations in failure probability were observed simply by rearranging the internal equipment. In another fundamental result, the model established that a correlation existed between the probability of failure of a satellite and the size of a debris impactor.

These results have profound implications for the design of an unmanned satellite. If the probability of failure of a satellite, due to debris impact, is calculated to be even a few percent then this represents a very sizeable proportion of a satellite's through-life reliability specification. It means that satellite designers will be under greater pressure to accept the incorporation of debris risk calculations as part of the normal satellite reliability budget. With the development of models such as SHIELD, this is now a credible proposition. Obviously, in such a circumstance, the reduction of debris risk would become a more dominant satellite design driver than it has been in the past, and so the desire for well considered and cost-effective protection will increase.

7.5 Implications for Achieving Cost Effective Debris Protection in Satellites

The construction of an objective function, which can trade off the costs of applying protection with the benefits of reducing risk, is key to finding a cost-effective protection strategy. For example, the survivability metric in SHIELD has demonstrated that radically different protection strategies can be evaluated and compared. In particular, the model provided very clear evidence for the importance of defining a 'good' internal arrangement of the satellite equipment. The location of critical equipment away from vulnerable satellite faces was found to be highly advantageous, especially when less essential equipment was placed in-between. Generally, this protection strategy has been poorly considered by designers, partly through a lack of awareness of its potential, which is only now becoming apparent with the results of research such as this. If designers are to embrace such a strategy in future then, for maximum effectiveness, it must be implemented in the earliest stages of a satellite design.

The other main protection strategy, of increasing the physical shielding to the satellite body panels, was also shown to be potentially cost-effective. This was particularly true for high penetration probability regions, such as the front and side faces, but not for the less vulnerable faces. The results from SHIELD indicated that, in practice, this protection strategy is likely to be complementary to the one above, i.e. they should not be considered independently of one another. That of course adds further complexity to the problem when trying to identify the most cost-effective solution. Therefore, the usefulness of an automated search technique, such as the genetic algorithm in SHIELD, should not be underestimated.

7.6 Further Work

The work presented in this Ph.D. represents a promising start to a relatively new field of debris research, namely satellite survivability and protection modelling. There is of course much still to do. Various models and algorithms require extension and improvement; a more comprehensive analysis programme has to be performed; and a rigorous set of protection guidelines needs to be derived.

Examining the first of these, perhaps the most important and difficult aspect for SHIELD to address is the satellite internal damage assessment. The key to this is a realistic model for an expanding penetrative debris cloud and its interaction with various types of internal equipment. An empirical distribution function has been proposed, based on an existing secondary ejecta model, that could characterise the interaction between the particles of a dispersed cloud and one or more of a satellite's internal units. The development of such a function is however dependent on an impact test programme. For a non-dispersed cloud interacting with one or more units, then a different approach is suggested, which again requires supporting impact test data.

The response of specific items of equipment to a particle or cloud impact is another important unknown at this time. In SHIELD, a simple fail / survive criterion is applied depending on whether the unit is penetrated or not. Experience from the defence sector suggests that this is not necessarily the case in practice. There is in fact a quantifiable probability of survival, which again can be determined through impact tests.

At the time of writing, ESA are proposing to issue a contract to investigate the vulnerability of internal satellite equipment to debris impacts. If the resulting test programme can address either or both of the above issues as part of the investigation, it will be a significant contribution to the ongoing development of models such as SHIELD.

Another element of the survivability evaluation process in SHIELD that would benefit from further effort is the development of a module with similar functionality to Fault Tree Analysis

(FTA). This would replace the current simpler approach where mission failure is determined according to whether a particular failed unit is 'critical' or 'non-critical' to the mission. The benefit of an FTA-type approach is that it can cater for failures of redundant equipment in a more meaningful fashion.

The modelling of costs in the survivability metric is another area that deserves further investigation. A more detailed set of Cost Estimating Relationships (CERs) could be envisaged for the cost model. This would have to be supported by accurate data on the costs of manufacturing and integrating shielding on actual spacecraft, which may be difficult due to the commercial sensitivity of such information. The cost model could also be extended to include a revenue term. For example, if a particular satellite design has some shielding mass added, then to offset this in the design it may be necessary to reduce the payload or on-board propellant mass. This of course translates into a reduction in revenue and therefore cost-effectiveness of the mission.

Other improvements to the SHIELD software include the incorporation of meteoroid fluxes; the use of a more sophisticated Graphical User Interface (GUI) and 3-D graphics library such as OpenGL; and the addition of wiring and pipes as a special type of equipment in the satellite design. With regard to this last suggestion, the geometric representation of wiring and pipes is not straightforward, particularly in the genetic algorithm search, where designs are generated automatically by the software. So, considerable thought and care will be needed to incorporate these correctly.

In terms of expanding upon the analyses conducted in this Ph.D. programme, there are many issues that need to be examined more closely. Some of the results presented so far, although very promising, can not yet be considered conclusive. Much rests on the above-mentioned model improvements, and the ability to further validate the results. If possible, a comparison study between SHIELD and another model such as INTAVAL or MSCSurv would be a major step forward in the validation process.

In the short term, future studies could focus on one or more of the following important issues:

- To examine a much broader range of idealised satellite representations and missions than have been covered so far in this research. This would confirm either the generality of the results, or establish whether there are exceptions.
- To gain a better understanding of the role that equipment redundancy plays in enhancing protection. In particular, it would be useful to clarify whether distributed redundancy is an improvement or not.
- To look at the importance of wiring and piping as critical risk items compared to other items. It is conceivable that mission failure probability may be dominated by impacts on these items rather than other vulnerable equipment in a satellite.
- To derive a generalised relationship between mission failure probability and the size of a debris impactor. This requires a much more extensive investigation compared to the specialised cases already studied.
- To test whether the genetic algorithm search efficiency can be improved by fine-tuning parameters such as the probabilities of cross-over and mutation. It may also be beneficial to check if a hybrid version of the genetic algorithm will be effective in converging on global optimum solutions.
- To evaluate a variety of realistic spacecraft designs, such as METOP. This would illustrate SHIELD's suitability not just in a research capacity, but also as a project support tool.

In the longer term, the results of these investigations should provide the basis for developing a considered and rigorous set of protection guidelines for unmanned spacecraft. These would be compared with the established guidelines summarised at the outset of this research.

Finally, a very ambitious long-term aim would be to extend SHIELD's optimisation capability to include non-debris environmental terms and further system design constraints. For example, radiation and charging are two other environmental factors that have a strong influence on satellite survivability, shielding, and the location of equipment. Meanwhile, constraints such as thermal control and electromagnetic compatibility of equipment similarly have a strong influence. Essentially, SHIELD would become a multidisciplinary design optimisation (MDO) tool.

7.7 Summary

The thrust of this research has been the development of a model comprising new techniques to quantify and compare radically different debris protection strategies for unmanned spacecraft. This was driven by a belief that simply adding shielding mass to a satellite is not necessarily the best approach. Results appear to confirm this, and show that a complementary, and possibly alternative, strategy is to consider the vulnerability and arrangement of equipment inside the spacecraft. Such a strategy requires designers to accept a shift in philosophy from 'design for no penetration' to 'design for no failure'. This is a subtle but important difference, since it implies accepting the possibility of a higher penetration risk. However, the strategy is essential to avoid over-designing the protection, and therefore to achieve a cost-effective solution.

In conclusion, the implementation of cost effective debris protection in unmanned spacecraft is a challenging problem that requires original thinking and innovative modelling techniques. This Ph.D. research programme represents a first step towards that objective.

Bibliography

Alby, F., Lansard, E., Michal, T., *Collision of Cerise with Space Debris*, Proceedings of the Second European Conference on Space Debris, ESA SP-393, pp. 589-596, 1997.

Anz-Meador, P.D., Matney, M.J., Liou, J.-C., Johnson, N.L., *Updating the NASA Engineering Model: A Review of Source Data and Analytical Techniques*, 33rd COSPAR Plenary Meeting, Warsaw, July, 2000.

Anz-Meador, P., *A Decade of Growth*, Orbital Debris Quarterly News, Orbital Debris Program Office, NASA Johnson Space Center, Vol. 5, Issue 4, October, 2000.

Ashley, H., *On Making Things the Best - Aeronautical Uses of Optimisation,* Journal of Aircraft, Vol. 19, pp. 5 - 28, January 1982.

Banhart, J., *Foam Metal: The Recipe*, Europhysics News, Vol. 30, No. 1, Jan/Feb 1999.

Bendisch, J., Rex, D., *The Long-Term Evolution of Orbital Debris - New Findings Concerning Collisional Cascading*, IAA-95-IAA.6.4.08, 46th International Astronautical Congress, Oslo, Norway, October 1995.

Berthoud, L. Mandeville, J.C., *Empirical Impact Equations and Marginal Perforation*, Proceedings of the First European Conference on Space Debris, Darmstadt, ESA SD-01, pp. 459-464, 1993.

Beruto, E., Destefanis, R., Faraud, M., Buchwald, P., *Debris Shielding Development for the COLUMBUS Orbital Facility*, Proceedings of the Second European Conference on Space Debris, Darmstadt, ESA SP-393, pp. 509-518, May 1997.

Birnbaum, N.K., Cowler, M., Hayhurst, C., *Numerical Simulation of Impact using AUTODYN*, Proceedings of the 2nd International Symposium on Impact Engineering, Beijing, September 1996.

Bol, J., Fucke, W., *Shaped Charge Technique for Hypervelocity Impact Tests at 11 km/s on Space Debris Protection Shields*, Proceedings of the Second European Conference on Space Debris, Darmstadt, ESA SP-393, pp. 391-396, May 1997.

Bourne, N., Vignjevic, R., Millett, J., Campbell, J., *Capabilities of the Centre for Materials Dynamics at Cranfield University*, Presentation at ESA Network of Centres technical meeting on Hypervelocity Impacts and Protection, ESTEC, Noordwijk, February 2002.

Burchell, M.J., Cole, M.J., McDonnell, J.A.M., Zarnecki, J.C., *Hypervelocity impact studies using the 2MV Van de Graaff accelerator and two-stage light gas gun of the University of Kent at Canterbury*, Measurement Science Technology, Vol. 10, pp. 41-50, 1999.

Carter, M.T., Vadali, S.R., *Parameter Optimization using Adaptive Bound Genetic Algorithms*, From: AAS/AIAA Spaceflight Mechanics Meeting, Albuquerque, New Mexico, AAS-95-140, 13-16 February 1995.

Caswell, D., *Olympus and the 1993 Perseids: Lessons for the Leonids*, Leonids Threat Conference, Manhattan Beach, April 26/271998.

Christiansen, E.L., Hyde, J.L., Snell, G., *Spacecraft Survivability in the Meteoroid and Debris Environment*, AIAA 92-1409, AIAA Space Programs and Technologies Conference, Huntsville, Alabama, March 1992.

Christiansen, E.L., *Performance Equations for Advanced Orbital Debris Shields*, AIAA 92-1462, AIAA Space Programs and Technologies Conference, Huntsville, Alabama, March 1992.

Christiansen, E.L., *Design and Performance Equations for Advanced Meteoroid and Debris Shields*, International Journal of Impact Engineering, Vol. 14, pp. 145-156, 1993.

Christiansen, E.L., Cykowski, E., Ortega, J., *Highly Oblique Impacts into Thick and Thin Targets*, International Journal of Impact Engineering, Vol. 14, pp. 157-168, 1993.

Christiansen, E.L., Kerr, J.H., *Mesh Double Bumper Shield: A low-weight alternative for spacecraft meteoroid and orbital debris protection*, International Journal of Impact Engineering, Vol. 14, pp. 169-180, 1993.

Christiansen, E.L., Hyde, J.L., *Validating BUMPER Code Perdictions Using Observed Meteoroid and Orbital Debris Impact Damage*, AIAA 94-4483, September 1994.

Christiansen, E.L., Crews, J.L., Williamsen, J.E., Robinson, J.H., Noten, A.M., *Enhanced Meteoroid and Orbital Debris Shielding*, International Journal of Impact Engineering, Vol. 17, pp. 217-228, 1995.

Christiansen, E.L., *NASA progress report*, presented at the Protection Working Group meeting, 16th IADC, Toulouse, France, November 1998.

Cour-Palais, B.G., *Meteoroid Environment Model 1969*, NASA SP-8013, 1969.

Cour-Palais, B.G., Crews, J.L., *A Multi-Shock Concept for Spacecraft Shielding*, International Journal of Impact Engineering, Vol. 10, pp. 135-146, 1990.

Coverstone-Carroll, V., *Near-Optimal Low-Thrust Trajectories via Micro-Genetic Algorithms*, J. Guidance, Vol. 20, No. 1: Engineering Notes, 1996.

Davis, L. (ed.), *Handbook of Genetic Algorithms*, International Thomson Computer Press, 1996.

De Cecco, M., Pavarin, D., *Computer Based Control and Diagnosis of a Two Stage Light Gas Gun by means of Pressure Recordings*, IMEKO TC-4 Symposium on Development in Digital Measuring Instrumentation, Naples, Italy, September 1998.

De Jong, K. A., *An Analysis of the Behaviour of Genetic Adaptive Systems*, Dissertation Abstracts International, Vol. 41, No. 9, Paper 3503B, 1975.

Destefanis, R., Callea, M., *Meteoroid and Debris Impacts on Orbiting Structures: A Methodology*, Proceedings of the First European Conference on Space Debris, Darmstadt, ESA SD-01, pp. 523-533, 1993.

Destefanis, R., Faraud, M., *Testing of Advanced Materials for High Resistance Debris Shielding*, International Journal of Impact Engineering, Vol. 20, pp. 209-222, 1997.

Divine, N., Grün, E., Staubach, P., *Modelling the Meteoroid Distributions in Interplanetary Space and Near Earth*, Proceedings of the First European Conference on Space Debris, ESA SD-01, pp. 245-250, April 1993.

Drohlshagen, G., McDonnell, J.A.M., Stevenson, T., Aceti, R., Gerlach, L., *Post Flight Measurements of Meteoroid/Debris Impact Features on EURECA and the Hubble Solar Array*, Adv. Space Res., Vol. 16, No. 11, pp. 85-89, 1995.

Dyer, C.S., Truscott, P.R., Evans, H., Sims, A.J., Hammond, N., Comber, C., *Secondary Radiation Environments in Heavy Space Vehicles and Instruments*, Advances in Space Research, Vol. 17, No. 2, pp. 53-58, 1996.

Eichler, P., Rex, D., *Debris Chain Reactions*, AIAA 90-1365, AIAA/NASA/DOD Orbital Debris Conference: Technical Issues & Future Directions, Baltimore MD, April 1990.

ESA, *European Space Debris Safety and Mitigation Standard*, Issue 1, Revision 0, September 2000.

Ewing, M. S., Downs, K., *Optimization of a Rectangular Cross-Section Wingbox using Genetic Search Algorithms*, AIAA Paper 96-1536-CP, 1996.

Fortson, B.H., *Trade-offs in Mass and Effectiveness in Satellite Shielding: A Design Approach*, Final Report, PL-TR-92-1020, Phillips Laboratory, Kirtland AFB, New Mexico, August 1992.

Friichtenicht, J.F., Slattery, J.C., *Ionization Associated with Hypervelocity Impact*, NASA TN D-2091, 1963.

Furuya, H., Haftka, R. T., *Combining Genetic and Deterministic Algorithms for Locating Actuators on Space Structures*, AIAA Paper 95-1400-CP, 1995.

Gage, P. J., *New Approaches to Optimisation in Aerospace Conceptual Design,* NASA Contractor Report 196695, March 1995.

Grady, D.E., *Fragmentation of Rapidly Expanding Jets and Sheets*, International Journal of Impact Engineering, Vol. 5, pp. 285-292, 1987.

Grün, E., Zook, H.A., Fechtig, H., Giese, R.H., *Collisional Balance of the Meteoric Complex*, ICARUS 62, pp. 244-277, 1985.

Gurule, A.P., Yates, K.W., Evans, R.M., *Impact of Space Debris on Solar Photovoltaic Array Performance*, IAF paper 92-0335, presented at the 43rd International Astronautical Congress, Washington, USA, September 1992.

Hajela, P., Yoo, J., *Constraint Handling in Genetic Search - A Comparative Study*, AIAA Paper 95-1143-CP, 1995.

Hayhurst, C., Hiermaier, S.J., Clegg, R.A., Riedel, W., Lambert, M., *Development of Material Models for Nextel and Kevlar-Epoxy for High Pressures and Strain Rates*, Hypervelocity Impact Symposium, Huntsville, Alabama, November 1998.

Hayhurst, C., Clegg, R.A., Hiermaier, S.J., Riedel, W., Wentzel, C.M., Lambert, M., *Advanced Material Models for Hypervelocity Impact Simulations*, presented at the Protection Working Group meeting, 17th IADC, ESOC, Darmstadt, Germany, October 1999.

Herbert, M.K., Taylor, E.A., *Hypervelocity Impact Response of Honeycomb: Shielding Performance and Spacecraft Subsystem Design Issues*, IAT/NASA Hypervelocity Shielding Workshop, Galveston, Texas, March 1998.

Hicks, D. L., Porter, B., *Evaluation of Optimisation Procedures in the Design of Digital Model-Following Systems*, AIAA Paper 95-3260-CP, 1995.

Huang, M-W., Arora, J. S., *Engineering Optimisation with Discrete Variables*, AIAA Paper 95-1333-CP, 1995.

IADC, *Protection Manual*, IADC-WD-00-03, version 2.0, 1 May 2001.

Jenkin, A. B., *The Importance of Continued Study of the Hazard Posed by Orbital Debris: Orbital Debris and Satellite Constellations*, AIAA News, February 1998.

Jenniskens, P., *Meteor Stream Activity I, The annual streams*, Journal of Astronomy & Astrophysics, 287, pp.990-1013, 1994.

Johnson, N. L., *History of On Orbit Fragmentations*, Teledyne Brown Engineering, Colorado Springs, CS92-TR-JSC-007, Sixth Edition, 1992.

Johnson, N.L., *Monitoring and Controlling Debris in Space*, Scientific American, pp. 42-47, August 1998.

Jonas, F., Yates, K., Evans, R., *Comparisons of Debris Environment Model Breakup Models*, AIAA 93-0166, 31st Aerospace Sciences Meeting & Exhibit, January 1993.

Keane, A. J., *The OPTIONS Design Exploration System - Reference Manual and User Guide*, Version B0.2, University of Southampton, September 1993.

Keane, A. J., *Experiences with Optimisers in Structural Design*, Proceedings of ACEDC, 1994.

Keane, A. J., *Passive Vibration Control via Unusual Geometries: The Application of Genetic Algorithm Optimization to Structural Design*, Journal of Sound and Vibration, 185(3), p. 441 - 453, 1995.

Keane, A. J., *A Brief Comparison of Some Evolutionary Optimization Methods*, In: Modern Heuristic Search Methods, John Wiley & Sons Ltd., 1996.

Kessler, D. J., *Collisional Cascading: The Limits of Population Growth in Low Earth Orbit*, Adv. Space Res. Vol. 11, No. 12, pp. 63-66, 1991.

Kessler, D. J., *Orbital Debris Environment in Low Earth Orbit: An Update*, Adv. Space Res. Vol. 13, No. 8, pp. 139-148, 1993.

Kessler, D. J., *The Current and Future Environment: An Overall Assessment*, The Preservation of Near-Earth Space for Future Generations, Cambridge University Press, 1994.

Kessler, D.J., Zhang, J., Matney, M.J., Eichler, P., Reynolds, R.C., Anz-Meador, P.D., Stansbery, E.G., *A Computer-based Orbital Debris Environment Model for Spacecraft Design and Observations in Low-Earth Orbit*, NASA TM 104825, 1996.

Klinkrad, H., *Collision Risk Analysis for Low Earth Orbits*, Adv. Space Res., Vol. 13, No. 8, pp. 177-186, 1993.

Klinkrad, H., Bendisch, J., Bunte, K.D., Krag, H., Sdunnus, H., Wegener, P., *The MASTER-99 Space Debris and Meteoroid Environment Model*, 33rd COSPAR Plenary Meeting, Warsaw, July 2000.

KrishnaKumar, K., *Genetic Algorithms - A Robust Optimization Tool*, AIAA Paper 93-0315, 31st Aerospace Sciences Meeting & Exhibit, January 1993.

Krishnakumar, K., Swaminathan, R., Garg, S., Narayanaswamy, S., *Solving Large Parameter Optimisation Problems Using Genetic Algorithms*, AIAA Paper 95-3223, 1995.

Krisko, P.H., Johnson, N.L., Opiela, J.N., *Evolve 4.0 Orbital Debris Mitigation Studies*, 33rd COSPAR Plenary Meeting, Warsaw, July 2000.

Lambert, M., *Shielding against Orbital Debris – A Challenging Problem*, In: Proc. ESA Symp.: 'Space Applications of Advanced Structural Materials', ESTEC, Noordwijk, ESA SP-303, June 1990.

Lambert, M., Schneider, E., *Shielding against Space Debris. A Comparison between Different Shields: The Effect of Materials on their Performance*, International Journal of Impact Engineering, Vol. 17, pp. 477-486, 1995.

Lawrence, R.J., *A Simple Model for the Optimization of Stand-off Hypervelocity Particle Shields*, International Journal of Impact Engineering, Vol. 5, pp. 451-461, 1987.

Lawrence, R.J., *A Simple Approach for the Design and Optimization of Stand-off Hypervelocity Particle Shields*, AIAA 92-1465, AIAA Space Programs and Technologies Conference, Huntsville, Alabama, March 1992.

Leler, W., Merry, J., *3D with HOOPS*, Publ. Addison Wesley Longman Inc., 1996.

Lemcke, C., et al, *Enhanced Debris / Micrometeoroid Environment Models and 3D Software Tools*, ESABASE/DEBRIS Release 2, Final Report, ESA-CR-P-4214, August 1998a.

Lemcke, C., Scheifele, G., Maag, M.C., *Enhanced Debris / Micrometeoroid Environment Models and 3D Software Tools*, ESABASE/DEBRIS Release 2 Final Presentation, June 1998b.

Lu, C-Y., Nahra, H.K., *Assessment of environmental effects on Space Station Freedom Electrical Power System*, In: Proceedings of the 26th Intersociety Energy Conversion Engineering Conference, Boston, MA, Vol. 1, pp. 374-379, August 1991.

Lu, C-Y., Aronoff, I., *Spacecraft electrical power system environmental control and protection - A Low Earth Orbit Space Station example*, In: Proceedings of the 29th Intersociety Energy Conversion Engineering Conference, Monterey, CA, Vol. 1, pp. 326-331, August 1994.

McDonnell, J.A.M., McBride, N., Gardner, D.J., *The Leonid Meteoroid Stream: Spacecraft Interactions and Effects*, Proceedings of the Second European Conference on Space Debris, Darmstadt, ESA SP-393, pp. 391-396, May 1997.

McDonnell, J.A.M., *HVI Phenomena: Applications to Space Missions*, International Journal of Impact Engineering, 1999.

McGill, P.B., Mount, A.R., *Effectiveness of Metal Matrix and Ceramic Composites as Orbital Debris Shield Materials*, AIAA 92-1461, AIAA Space Programs and Technologies Conference, Huntsville, Alabama, March 1992.

McKnight, D., Justice, D., Riley, W., Shukry, I., Shukry, A., *Correlation of Spacecraft Anomalies to the Debris Environment: An Update*, Acta Astronautica, Vol. 41, No. 11, pp. 751-756, 1997.

Maclay, T.D., Fudge, M.L., Harris, C.M., *Validity of the Poisson Distribution for Orbital Debris Impact Problems*, In: Proceedings of the SPIE Conference on Characteristics and Consequences of Orbital Debris and Natural Space Impactors, Denver, Colorado, Vol. 2813, pp. 116-126, August 1996.

Maiden, C.L., *Experimental and Theoretical Results Concerning the Protective Ability of a Thin Shield against Hypervelocity Projectile*, 6th Symposium on Hypervelocity Impact, Vol. III, pp. 70 – 156, 1963.

Matney, M., Goldstein, R., Kessler, D., Stansbery, E., *Recent Results from the Goldstone Orbital Debris Radar*, Adv. Space Res., Vol. 23, No. 1, pp. 5-12, 1999.

Mazza, C., Fairclough, J., Melton, B., De Pablo, D., Scheffer, A., Stevens, R., *Software Engineering Standards*, ESA PSS-05-0, 1994.

Noton, M., *Orbital Strategies Around a Comet by Means of a Genetic Algorithm*, J. Guidance, Vol. 18, No. 5: Engineering Notes, 1995.

Office of Science and Technology Policy, *Interagency Report on Orbital Debris*, 1995.

Orbital Debris Program Office, NASA Johnson Space Center, *Orbital Debris Quarterly News*, Vol. 4, Issue 1, January 1999.

Passman, S.L., *Models for Penetration of Thin Plates*, In: Shock Waves in Condensed Matter (Ed.: Schmidt, S.C. & Holmes, N.C.), pp. 733-736, 1987.

Piekutowski, A.J., *A Simple Dynamical Model for the Formation of Debris Clouds*, International Journal of Impact Engineering, Vol. 10, pp. 453-471, 1990.

Pinon III, E., Fowler, W.T., *Use of a Genetic Algorithm to Generate Earth to Moon Trajectories*, From: AAS/AIAA Spaceflight Mechanics Meeting, Albuquerque, New Mexico, AAS-95-141, 13-16 February 1995a.

Pinon III, E., Fowler, W.T., *Lunar Launch Trajectory Optimization using a Genetic Algorithm*, From: AAS/AIAA Spaceflight Mechanics Meeting, Albuquerque, New Mexico, AAS-95-142, 13-16 February 1995b.

Poe, R.F., Rucker, M.A., *Evaluation of Pressurized Vessels following Hypervelocity Particle Impact*, Proceedings of the First European Conference on Space Debris, Darmstadt, ESA SD-01, pp. 441-446, 1993.

Ratcliff, P.R., Cole, M.J., McDonnell, J.A.M., Shaw, H.A., Allahdadi, F., *Experimental Determination of Energy Partitioning in Microparticle Impacts at Velocities from 1 to 100 kms^{-1}*, presented at the 46th International Astronautical Congress, Oslo, Norway, October 1995.

Ratcliff, P.R., Reber, M., Cole, M.J., Murphy, T.W., Tsembelis, K., *Velocity Thresholds for Impact Plasma Production*, Adv. Space Res., Vol. 20, No. 8, pp. 1471-1476, 1997.

Reimerdes, H-G., Stecher, K-H., Lambert, M., *Ballistic Limit Equations for the Columbus Double-Bumper Shield Concept*, Proceedings of the First European Conference on Space Debris, Darmstadt, ESA SD-01, pp. 433-439, 1993.

Reimerdes, H-G., Wohlers, W., *Optimization of Micrometeoroid and Space Debris Protection Systems*, Third European Conference on Space Debris, March 2001.

Reynolds, R., Bade, A., Siebold, K., Johnson, N., *Debris Environment Interactions with Low Earth Orbit Constellations*, Proceedings of the Second European Conference on Space Debris, ESA SP-393, pp. 351-357, May 1997.

Richardson, A.J., *Theoretical Penetration Mechanics of Multisheet Structures based on Discrete Debris Particle Modeling*, AIAA 69-371, AIAA Hypervelocity Impact Conference, Cincinnati, Ohio, April-May 1969.

Rival, M., and Mandeville, J.C., *Modeling of Ejecta Produced upon Hypervelocity Impacts, Space Debris*, Vol. 1, No. 1, pp. 45-57, 1999.

Robinson, J.H., Nolen A.M., *An Investigation of Metal-Matrix Composites as Shields for Hypervelocity Orbital Debris Impacts*, International Journal of Impact Engineering, Vol. 17, pp.685-696, 1995.

Rossi, A., Anselmo, L, Pardini, C., Farinella, P., Cordelli, A., *Interaction of the Satellite Constellations with the Low Earth Orbit Debris Environment*, Mission Design & Implementation of Satellite Constellations, J.C. van der Ha (ed.), Kluwer Academic Publishers, pp. 327-335, 1997.

Schäfer, F., Schneider, E., Lambert, M., *An Experimental Study to Investigate Hypervelocity impacts on Pressure Vessels*, Proceedings of the Second European Conference on Space Debris, Darmstadt, ESA SP-393, pp. 435-443, May 1997.

Schäfer, F., *Hypervelocity Impacts on Shielded Pressure Vessels*, ESA Study, presented at the Protection Working Group meeting, 16th IADC, Toulouse, France, November 1998.

Schäfer, F., *Shape effects in hypervelocity impacts: Ongoing work*, presented at the 17th IADC, ESOC, Darmstadt, Germany, October 1999.

Schonberg, W.P., *Hypervelocity Impact Response of Space Composite Material Structures*, International Journal of Impact Engineering, Vol. 10, pp. 509-523, 1990.

Schonberg, W.P., Bean, A.J., Darzi, K., *Hypervelocity Impact Physics*, NASA CR-4343, 1991.

Schonberg, W.P., *Characterizing Material States in Orbital Debris Impacts*, In: Proceedings of the SPIE Conference on Space Environmental, Legal, and Safety Issues, Orlando, Florida, Vol. 2483, pp. 31-39, April 1995.

Scott, W.B., *Leonids Shower Triggers New Look at Space Debris*, Aviation Week & Space Technology, pp. 51-55, January 4, 1999.

Sdunnus, H., Drolshagen, G., Lemcke, C., *Enhanced Meteoroid/Debris 3-D Analysis Tool*, Proceedings of the Second European Conference on Space Debris, Darmstadt, ESA SP-393, pp. 655-660, May 1997.

Sdunnus, H., Griffiths, A., McDonnell, J.A.M., *Review of Existing Flux Models*, Work Package 1 Technical Note, ESA Contract 13145/98/NL/WK (Update of Statistical Meteoroid Debris Models for GEO), May 1999.

Settecerri, T.J., Stansbery, E.G., Matney, M., *Haystack Measurements of the orbital Debris Environment*, Adv. Space Res., Vol. 23, No. 1, pp.13-22, 1999.

Seywald, H., Kumar, R. R., *Genetic Algorithms and Equality Constrained Optimization Problems*, AIAA Paper 95-3225-CP, 1995.

Shiraki, K., Terada, F., Harada, M., *Space Station JEM Design Implementation and Testing for Orbital Debris Protection*, International Journal of Impact Engineering, Vol. 20, pp. 723-732, 1997.

Space Station Freedom Program Office, *Space Station Natural Environment Definition for Design*, Revision A, SSP 30425, Reston, VA, June 1991.

SpaceBase - Database of Spacecraft Designs and Failures, QinetiQ, 2001.

Spinks, G.A., *Hybrid Propulsion Research*, DERA Technical Report DERA/LWS/WS3/TR010439, DERA Fort Halstead, UK, May 2001.

Stilp, A.J., *Review of Modern Hypervelocity Impact Facilities*, International Journal of Impact Engineering, Vol. 5, pp. 613-621, 1987.

Stilp, A., *Hypervelocity Impact Research*, Proceedings of the Second European Conference on Space Debris, Darmstadt, ESA SP-393, pp. 399-404, May 1997.

Stokes, P. H., Swinerd, G. G., Crowther, R., Marsh, V., Walker, R. J., *A New Approach for Optimising Satellite Shielding and Configuration using Genetic Algorithms*, Presented at the Second European Conference on Space Debris, ESOC, Darmstadt, Germany, March 1997.

Stokes, P.H., *SHIELD Software: Software Project Management Plan*, DERA Report No. DERA/CIS/CIS2/4/82/PMP027/1.0, August 1997a.

Stokes, P.H., *SHIELD Software: User Requirements Document*, DERA Report No. DERA/CIS(CIS2)/URD/97083/1.0, July 1997b.

Stokes, P.H., *SHIELD Software: Software Requirements Document*, DERA Report No. DERA/CIS(CIS2)/SRD/97109/1.0, September 1997c.

Stokes, P.H., *SHIELD Software: Architectural Design Document*, DERA Report No. DRA/CIS(CIS2)/ADD/96/02/2.0, October 1997d.

Stokes, H., Swinerd, G., Walker, R., Wilkinson, J., *SHIELD: A New Model to identify Optimum Debris Protection for Unmanned Spacecraft*, IAA-99-IAA.6.5.02, presented at the 50th International Astronautical Congress, Amsterdam, The Netherlands, October 1999.

Stokes, H., Swinerd, G., Walker, R., Wilkinson, J., Martin, C., *Achieving Cost-Effective Debris Protection of Unmanned Spacecraft using SHIELD*, IAA-00-IAA.6.5.06, presented at the 51st International Astronautical Congress, Rio de Janeiro, Brazil, October 2000.

Swift, H.L., Bamford, R., Chen, R., *Designing Dual Plate Meteoroid Shields – A New Analysis*, Jet Propulsion Laboratory (JPL) Publication 82-39, March 1982.

Taylor, E.A., Herbert, M.K., Kay, L., *Hypervelocity Impact on Carbon Fibre Reinforced Plastic (CFRP)/Aluminium Honeycomb at Normal and Oblique Angles*, Proceedings of the Second European Conference on Space Debris, Darmstadt, ESA SP-393, pp. 429-434, May 1997.

Taylor, E.A., Herbert, M.K., Vaughan, B.A.M., McDonnell, J.A.M., *Hypervelocity Impact on Carbon Fibre Reinforced Plastic / Aluminium Honeycomb: Comparison with Whipple Bumper Shields*, Hypervelocity Impact Symposium, Huntsville, Alabama, November 1998.

Taylor, E.A., *Computational Study of Hypervelocity Impact onto Whipple Bumpers and Sandwich Plates with Honeycomb Core*, ESA Report, EWP 2029, Issue 1.0, June 1999.

Taylor, A.D., *Earth Encounter Velocities for Interplanetary Meteoroids*, Adv. Space Res., Vol. 17, pp. 205-209, 1995.

Terrillon, F., Warren, H.R., Yelle, M.J., *Orbital Debris Shielding Design of the Radarsat Satellite*, IAF-91-283, 42nd Congress of the International Astronautical Federation, Montreal, Canada, October 1991.

Tribble, A.C., *The Space Environment*, Princeton University Press, 1995.

Trufitt, M., *INTAVAL – Integrated Air-Target Vulnerability Assessment Library*, Presentation to Inter Agency Debris Committee (IADC) Protection Working Group, October 1999.

Turner, R., Berthoud, L., Griffiths, A.D., McDonnell, J.A.M., Marriott, P., Stokes, P.H., Taylor, E.A., Wilkinson, J.E., *Cost-Effective Debris Shields for Unmanned Spacecraft: Final Report*, Issue 1, Submitted to ESA under Contract No. 12378/97/NL, December 1999.

UniSpace Kent, *Update of Statistical Meteoroid / Debris Models for GEO*, Summary Report, ESA Contract 13145/98/NL/WK, 2000.

UNCOPUOS (United Nations Committee on the Peaceful Uses of Outer Space), *Report of the Scientific and Technical Subcommittee on the Work of its Thirty-Fourth Session*, February 1997.

UNCOPUOS Scientific and Technical Subcommittee, *Technical Report on Space Debris*, A/AC.105/720, May 1999.

US National Research Council (Committee on Space Debris), *Orbital Debris – A Technical Assessment*, ISBN 0-309-05125-8, National Academy Press, 1995.

Walker, R., Hauptmann, S., Crowther, R., Stokes, H., Cant, A., *Introducing IDES: Characterising the Orbital Debris Environment in the Past, Present and Future*, presented at the 6th AAS/AIAA Space Flight Mechanics Meeting, Texas, February 1996.

Walker, R., Crowther, R., Marsh, V., Stokes, P.H., Swinerd, G.G., *Satellite Constellations and their Long Term Impact on the Debris Environment in Low Earth Orbit*, Proceedings of the Second European Conference on Space Debris, ESA SP-393, pp. 359-366, May, 1997a.

Walker, R., Crowther, R., Cosby, M., Stokes, P.H., Swinerd, G.G., *The Long-Term Impact of Constellations on the Debris Environment after the Implementation of Debris Mitigation Measures*, paper IAF-97-1AA.6.4.3, 48th International Astronautical Congress, Turin Italy, October, 1997b.

Walker, R., *Comparison of Collision Flux Models for the ESA Cost Effective Debris Shields Study*, DERA/CIS/CIS2/TR9803, WP1 Report submitted to ESA under Contract 12378/97/NL, March 1998.

Walker, R., Stokes, P.H., Wilkinson, J., Swinerd, G.G., *Enhancement and Validation of the IDES Orbital Debris Environment Model*, Space Debris 1, 1-20, 1999.

Walker, R., *The Long-Term Interactions of Satellite Constellations with the Orbital Debris Environment*, Ph.D. Thesis, University of Southampton, February 2000.

Walker, R., Stokes, P.H., Wilkinson, J.E., *Long-Term Collision Risk Prediction for Low Earth Orbit Satellite Constellations*, Acta Astronautica, Vol. 47, Nos. 2-9, pp. 707-717, 2000a.

Walker, R.J., Swinerd, G.G., Wilkinson, J.E., Martin, C.E., *Long Term Space Debris Environment Prediction*, DELTA Final Report, ESA Contract 12808/98/D/IM(SC), July, 2000b.

Walker, R., Martin, C.E., Stokes, P.H., Wilkinson, J.E., Klinkrad, H., *Studies of Space Debris Mitigation Options using the Debris Environment Long Term Analysis (DELTA) Model*, IAA-00-IAA.6.6.07, 51st International Astronautical Congress, Rio de Janeiro, Brazil, 2-6 Oct., 2000c.

Whipple, F.L., *Meteorites and Space Travel*, Astronomical Journal, 1161: 131, 1947.

Wilkinson, J., Stokes, H., *Assessment of Revised Cost-Effective Debris Shields for Unmanned Spacecraft: WP7 Technical Note*, DERA/CIS/CIS2/TR990942/1.0, Submitted to ESA under Contract No. 12378/97/NL, November 1999.

Williamsen, J., Schonberg, W., *Empirical Models for Spacecraft Damage from Orbital Debris Penetration and Effects on Spacecraft Survivability*, Proceedings of the Second European Conference on Space Debris, Darmstadt, ESA SP-393, pp. 519-525, May 1997.

Williamsen, J.E., Evans, H.J., Schonberg, W.P., *Effect of Multi-Wall System Composition on Survivability for Spacecraft Impacted by Orbital Debris*, Space Debris, Vol. 1, No. 1, pp. 37-43, 1999.

Yakovlev, M., Kulik, S., Agapov, V., *Small Satellites and Space Debris Issues*, Third European Conference on Space Debris, March 2001.

Yamamoto, K., Osamu, I., *New Evolutionary Direction Operator for Genetic Algorithms*, AIAA Journal, Vol. 33, No. 10: Technical Notes, 1995.

Yew, C.H., Grady, D.E., Lawrence, R.J., *A Simple Model for Debris Clouds produced by Hypervelocity Particle Impact*, International Journal of Impact Engineering, Vol. 14, pp. 851-862, 1993.