IAC-22-B4.3.9

CONSTRAINT PROGRAMMING FOR SCHEDULING THE OPERATIONS OF STRATHCUBE: A NANOSATELLITE FOR DETECTING SPACE DEBRIS

Iain, Hall(1, a), **Cheyenne, Powell**(1,b), **Annalisa, Riccardi**(1,c)

(1) *Aerospace Centre of Excellence, Department of Mechanical and Aerospace Engineering, University of Strathclyde, 75 Montrose Street, Glasgow G1 1XJ, United Kingdom.*
(a) *iain.hall.2018@uni.strath.ac.uk.*
(b) *cheyenne.powell@strath.ac.uk*
(c) *annalisa.riccardi@strath.ac.uk*

Abstract

The ever-increasing quantity of satellites and space debris in orbit pose a serious threat to the sustainable use of the space environment. To mitigate this threat, we must improve our detection and tracking space debris in low earth orbit, and to do this new space-based tracking methods will be required. Subsequently, it raises the need to optimise the schedules of these in orbit tracking satellites to maximise the number and accuracy of the debris detected. STRATHcube is a nanosatellite currently in development at the University of Strathclyde that will use passive bi-static RADAR to detect space debris and act as a technological demonstrator. This satellite will be used to exhibit the space debris tracking technology and will use the iridium constellation as an illuminator. However, the complex interplay of satellite positions, with respect to the illuminator constellation and the ground stations, makes scheduling operations of the satellite very complex and difficult for a human to compute without the aid of automatic solvers. The whole space industry is moving towards developing more autonomy on-board satellites, also related to on-board task management. Constraint programming is the technique used to schedule STRATHcube tasks by optimising RADAR detections, ground station communications, and on-board data handling. This was done by mathematically defining the constraints on the satellite, simulating periods of the mission to find relevant orbital and space environment data. These were then used to manually define a baseline schedule, which was used as a starting point for the constraint's optimisation search. The optimised schedule significantly improved the satellite operations compared to the manually designed one. The improvements in scheduling will be applied to STRATHcube to improve its operations and allow it to better demonstrate the use of passive bi-static RADAR for space debris detection. The optimisation methods could also be applied to future possible passive bi-static RADAR satellites to maximise their efficiency in operations.

1 Introduction

Space debris is a major threat to the operation of satellites and without effective Space Situational Awareness (SSA) it is not possible to quantify the threat posed or take mitigating actions[1]. SSA describes our overall understanding of the space environment and relies upon regular observation to keep a track of all the debris in Earth's orbit. Current space debris tracking is limited to debris in the >5cm ranges and fail to quantify the threat posed by smaller debris. This is an issue as the number of debris at smaller sizes is estimated to be very large with 1 million objects $> 1cm$ estimated to be present while only 31,450 are regularly tracked [2].

There is a need for new methods of tracking space debris at smaller scales to be developed. STRATHcube is a satellite being developed at the University of Strathclyde to demonstrate Passive Bistatic Radar (PBR) as one potential method [3][4]. Bistatic radar is a radar configuration in which the receiver and the transmitter are at separate locations [5]. Pre-existing transmitters, such as communication satellites, can be used as illuminators of opportunity allowing for the system to operate passively saving on the cost of developing and operating a transmitter.

Bistatic operation also allows for a Forward scatter configuration to be utilised in which the target is between the transmitter and receiver, significantly increasing the radar cross section of the target. This is shown in figure 1, for the point where the targets bistatic angle, β, is close to 180°

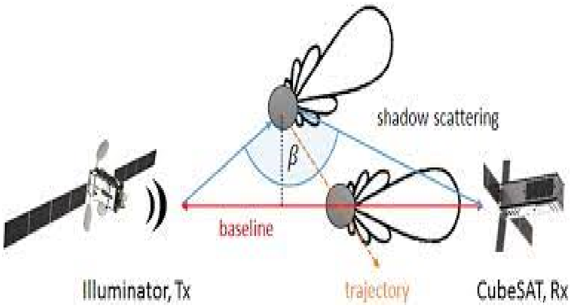


Figure 1: Diagram of PBR forward scatter setup

This forward scatter Radar Cross Section (RCS), σ, is defined equation 1 where A is the silhouette area and λ is the wavelength the radar uses[6].

$$\sigma = \frac{4\pi A^2}{\lambda^2} \tag{1}$$

The result of this is that forward scatter PBR can allow for significantly smaller objects to be detected. Forward scatter radar using satellite wavelengths has already successfully been used for detecting aircraft, proving that the technology works for moving targets and using weak satellite signals [7].

While this relation will allow PBR operating at shorter wavelengths in the K-band to detect objects down to $1cm^2$ in area [8], it will not be possible with STRATHcube because the limited budget has lead to the use of an L-band antenna. This will lead to the satellite only detecting objects with an area $> 7m^2$ [4]. This leads to a limitation in the amount of objects it may detect, limiting how well it will demonstrate the technology. To maximise STRATHcube's effectiveness as a technological demonstrator steps should be taken to ensure that it is possible to make as many observations of targets as possible and downlink them to ground stations. This will require the frequency at which detections will occur and how the satellite will handle them to be modelled.

The payload on STRATHcube will use a patch antenna, radar receiver, and software defined radio running on an onboard computer [4]. The use of an onboard computer provides flexibility in the design of the signals processing algorithm, and the ability to change it late in the design. The cost of this is that it does not achieve the processing rate an application specific integrated circuit with a set signal processing algorithm would [5].The performance of the payload using real time onboard processing, offboard processing, and non-real time onboard processing are considered in this investigation.

The limitations on when STRATHcube can communicate with the ground station, observe illuminators allowing the radar's operation, and constraints on data storage resources means that the problem of maximising the number of objects detected is a resource constrained scheduling problem. The resource constrained scheduling problem has applications in the scheduling of satellites in the form of the satellite range scheduling problem [9]. To quantify how well STRATHcube may operate it will be necessary to optimise the schedule. To optimise the operation of STRATHcube it was formulated using constraints programming for a scheduling problem [10].

The work in this paper aims to quantify the number of objects which could potentially be detected under different design circumstances and how the scheduling of operations could be optimised to maximise this.

The remainder of the paper is organised as follows: Section II describes the work to quantify the number of objects which may be detected, Section III describes the constraints upon the schedule and the method for optimising it, Section IV discusses the results, Section V concludes the paper, and Section VI outlines future work.

2 Quantifying Payload effectiveness

To quantify the number of objects STRATHcube may detect it is necessary to find the opportunities during which the radar can operate, the size of objects which may be detected during these opportunities, and the number of objects greater than this size expected to be within the region of observation.

The opportunities during which the radar can operate occur when the illuminator satellites are in view, in STRATHcube's case the Iridium constellation. To do this the orbits of the Iridium satellites and STRATHcube were propagated and the times when STRATHcube was within an Iridium satellite footprint were found. This was done using orbital simulation software developed as part of a digital twin of STRATHcube.

The minimum radar cross section of debris that could be detected was found using the bistatic radar range equation [6].

$$\sigma = \frac{(4\pi)^3 R_{Tx}^2 R_{Rx}^2 k T_0 B F_n L}{P_T G_T G_R \lambda^2} SNR \tag{2}$$

Where σ is the minimum detectable radar cross section. R_{Tx} is the range of the target from the transmitter, R_{Rx} is the range of the receiver from the target. k is Boltzmann's constant, $T_0 = 290°K$ is the signal noise temperature [11]. B is the noise bandwidth and is 177MHz for the antenna [12]. F_n is the noise figure and is assumed to be 1. L is the loss factor and is assumed to be 1. $P_T = 6dB$ is the power of the transmitter and $G_t = 24.87dB$ is the transmitter antenna gain for the Iridium satellites [11]. $G_R = 4dB$ is the receiver antenna gain. $\lambda = 185mm$ is the wavelength at which the satellite operates at [12]. $SNR = 10dB$ is the signal to noise ratio.

The result could then be used with the forward scatter radar cross section formula to find the area necessary to achieve the minimum RCS, and therefore the minimum size of object that could be detected. This was done at the midpoint between the illuminator and STRATHcube as this provides a conservative estimate of the detection capabilities.

$$A_{min} = \sqrt{\frac{\sigma\lambda^2}{4\pi}} \tag{3}$$

The volume of the region in which detection could occur had to be calculated. Forward scatter starts to occurs when the bistatic angle is greater than $176°$[6]. The region in which this occurs is bound by the arcs shown in figure 2 where SC is STRATHcube and IL is an illuminator. The actual volume was found by approximating the region as 2 cones which together had the volume: Vol_{FS}.

$$Vol_{FS} = 1/3\pi wL \tag{4}$$

Where L is the length of the baseline between SC and IL and w is the distance from the baseline to the edge of the region at the midpoint of the baseline. This is an underestimation but sufficient for the level of detail required by this investigation.

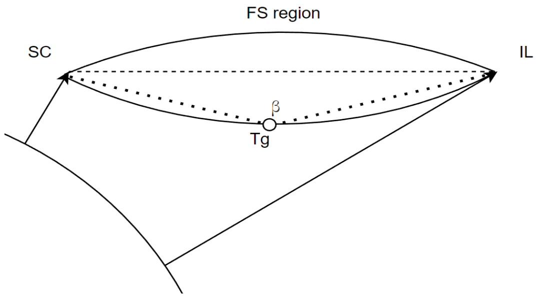


Figure 2: Forward scatter region diagram

The minimum area of an object for it to be detectable, A_{min} could then be used to estimate the number of objects greater than A_{min} within the forward scatter region. This allowed the average number of objects expected to be detected to be calculated. This was done using a homogeneous distribution of debris which took the number of objects larger than A_{min} in Low Earth Orbit (LEO), the volume of the forward scatter region(Vol_{FS}), and the volume of LEO (Vol_{LEO}). The value of Vol_{LEO} is defined in equation 5, where r_e is the radius of the Earth, and h_{LEO} is the maximum altitude of LEO and is 2000km.

$$Vol_{LEO} = \frac{4\pi}{3}\left((r_e + h_{LEO})^3 - r_e^3\right) \tag{5}$$

This allowed the fraction of objects within the forward scatter region to be found assuming a homogeneous distribution of LEO objects. The number of objects to be detected was done using data on the number of objects with different areas for LEO objects from European Space Agency (ESA) space environment report [2]. This was represented in an array α, and the sum of the objects in LEO that are larger than the minimum area was calculated to find the number of detectable objects in LEO. The product of this and the ratio of forward scatter region volume to LEO volume represented the number of objects on average expected to be detectable in the forward scatter region. This is shown in equation 6, where $\Theta_{t,n}$ is the average number of objects, Δt is the time discretisation, A_{max} is the size of the largest object in LEO, and α_i is the number of objects with an area i in LEO.

$$\Theta_{t,n} = \frac{Vol_{FS}}{Vol_{LEO}} \times \sum_{i=A_{min}}^{A_{max}} \alpha_i \Delta t \tag{6}$$

3 Optimisation of the Schedule

The scheduling problem was only solved from an off-board scheduling standpoint to simplify the problem. The scheduling of operations was done using a constraints satisfaction problem formulation, using google OR tool's CP-SAT solver [13]. This allowed the activities of the satellite to be constrained and then allowed the CP-SAT solver to optimise the control variables to maximise a cost function. A method using a manual heuristic defined schedule was also used to give a control for how effective CP-SAT solver was. This was done for 3 design cases for the radar signals processing:

- Real time onboard processing: where the signals processing is carried out during observation, requiring only processed data to be stored.
- Non-real time onboard processing: where signals processing is carried out after observation, requiring raw signal data to be stored until processing is done
- Off-board processing: where no signal processing is carried out and raw signal data must be downlinked.

The constraints and variables made were modified to allow all 3 cases to be modelled.

The scheduling problem was defined using two Boolean arrays as control variables. The first, X, was used to define which operation the satellite could take at any time and was one of 4 activities: observation, processing, down-linking, and idling. The second control variable, H, was used to define which illuminator of opportunity the satellite would chose to observe at a given time out of 66 possible Iridium satellites.

- $X \in \{0,1\}^{T \times A}$ where T is the length of the period to be optimised and A is the possible actions where:
    - a_0 is observation
    - a_1 is processing
    - a_2 is downlinking
    - a_3 is idling

- $H \in \{0,1\}^{T \times N}$ where T is the length of the time period and N is the number of possible targets $n \in \{0...65\}$

These variables were constrained such that only one action may be taken at once, and only one illuminator can be observed at once. This provides a conservative estimate of performance. To specify all the other constraints necessary additional constants and auxiliary variables had to be defined. The constants necessary are defined below:

- $M_{max} = 64GB$: Maximum memory available on the onboard computer dedicated to handling the payload
- $\dot{D}_m = 32kB/s$: The rate at which data is downlinked during down-linking activity
- $\dot{O}_m = 15,000kB/s$: The data produced during observation activity
- $C = 20MFLOPS$: Number of floating point operations that were possible to carry out per second
- $S = 200MFLOPs$: The number of floating point operations required to process the observation data.
- $\dot{P}_m = \frac{\dot{O}_m C}{S}$: The rate at which 1 second of observations are processed in kB/s of observation data.
- $Q = 0.002$: The ratio of processed data produced to observation data processed during processing. This was added to reflect that the processed data takes up significantly less space than the processed data.

It is assumed that the observation data can be dumped after processing only leaving the processed data ready to be downlinked. The values for C and S are estimated from the onboard computer's CPU and a Digital Signal Processing (DSP) method proposed for detecting fast moving objects in forward scatter bistatic radar.

The on board computer dedicated to the operation of the primary payload is the OBC-P3 [14] computer from space inventor and has 2 Arm Cortex-M7 processors[15]. These each have a floating point unit which is capable of carrying out two 16-bit multiply accumulate operations a cycle in its DSP unit and a clock speed of 5MHz, this comes to a theoretical number of floating point operations of 20 Mega Floating Point Operations per Second (MFLOPS). This allows us to make an estimation of how quickly observation data can be processed based on an estimation of the Floating Point Operations (FLOPs) necessary to carry out the specified DSP algorithm.

The number of FLOPs necessary to process the observed data will depend on the final digital signals processing algorithm implemented. An estimation of the number of FLOPs necessary was made from work on a DSP algorithm for forward scatter radar for fast moving objects [16]. The auxiliary variables were integer arrays as defined below.

- $O \in \mathbb{Z}^T$: the amount of data Observed in kB
- $P \in \mathbb{Z}^T$: the amount of data Processed in kB
- $D \in \mathbb{Z}^T$: the amount of data Downlinked in kB

Where Δt is the time discretisation, the element values of the auxiliary variables are defined as:

$$O_i = \sum_{t=1}^{i} \dot{O}_m X_{t,a_0} \Delta t - \dot{P}_{op} X_{t,a_1} \Delta t \tag{7}$$

$$P_i = \sum_{t=1}^{i} Q \dot{P}_m X_{t,a_1} \Delta t - \dot{D}_m X_{t,a_2} \Delta t \tag{8}$$

$$D_i = \sum_{t=1}^{i} \dot{D}_m X_{t,a_2} \Delta t \tag{9}$$

These could then be used to implement the following constraint on memory and the sequence of tasks

1. $O_i + P_i \leq M_{max},\ \forall i \in 0...T$: Used to constrain the used memory below the maximum memory at all times.
2. $O_i \geq \dot{P}_m \Delta t,$ if $X_{i,a_1} = 1$: To ensure there are observations to be processed before processing can occur.
3. $P_i \geq \dot{D}_m \Delta t,$ if $X_{i,a_2} = 1$: To ensure there is processed data to be downlinked before down linking can occur.

These constraints were applied to the case where non-real time onboard processing occurred. To model the other two design cases the model had to be modified. The second constraint was removed and the third constraint modified for the cases where real time onboard processing occurs or off-board processing occurs as shown below:

$$O_i >= \dot{D}_m \Delta t,\ if\ X_{i,a_2} = 1 \tag{10}$$

for the real time processing condition the constant for observed data rate was modified to reflect the fact that it is processed in real time, $\dot{O}_m = 30kB/s$

The final constraint added was to limit the rate at which the satellite could switch between observing different illuminators. This meant that if the satellite observed a given illuminator j at a time i then it would not be able to observe any other illuminators within a period δ This is shown in mathematical form below:

$$0 = \sum_{t=i-\delta}^{i+\delta} \sum_{n=0,n\neq j}^{N} H_{t,n} \text{ if } H_{i,j} = 1 \tag{11}$$

The number of detections at any time increment i was then defined as:

$$U_i = \sum_{t=0}^{T} \sum_{n=0}^{N} H_{t,n} \Theta_{t,n} \tag{12}$$

This could then be used to generate the cost function which was design to maximise the time spent downlinking and then the number of detections:

$$\max \left(\sum_{i=0}^{T} U_i + w_{a_2} X_{i,a_2} \right) \tag{13}$$

Where w_{a_2} was a weight used to ensure the solver prioritised downlinking as this is the most constrained resource of the mission. The CP-SAT solver was then used to optimise the constrained model for all three design cases.

The manual heuristics were developed to provide a comparison for the CP-SAT solved model. They were used to provide schedules that were within the constraints of the model. The manual heuristic followed the off-board processing case and the real time onboard processing case was applied:

1. When possible down-link images
2. Else when possible observe
   (a) If previously targeting an illuminator continue to target it
   (b) else target the satellite with the highest likelihood to achieve a detection
3. Else idle

The non-real time onboard processing case manual heuristic followed the algorithm below:

1. Schedule down-linking whenever it is possible to occur
2. Else schedule to observe
   (a) If previously targeting an illuminator continue to target it
   (b) else target the satellite with the highest likelihood to achieve a detection
3. Else schedule to process
4. Else schedule to idle.

4 Results

The optimisation program and the manual heuristics were run for all three design cases to produce results for their schedule and the performance (total detections). As the focus of the paper is on the macro effectiveness of the payload, the schedule was run with a times discretisation of 60 seconds over a horizon of two weeks. These were done for the case where the satellite is launched into a polar orbit from the Saxavord space centre with a signal gain of 40dB.

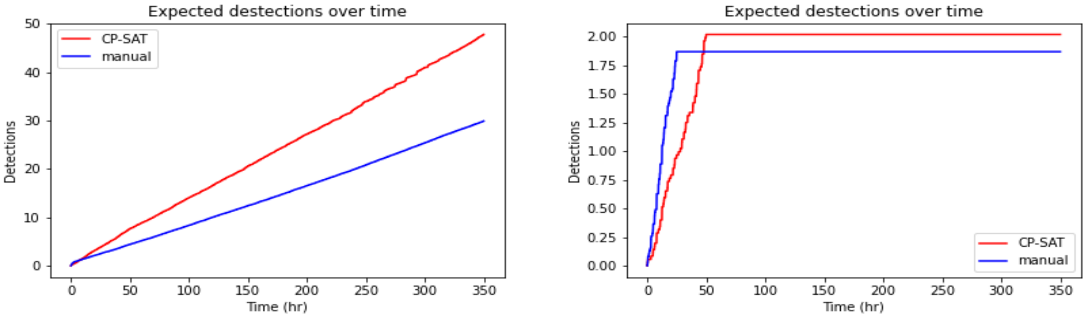
Non-real Time Onboard processing condition

Figure 3(a) the total number of detections achieved by the two schedules and by comparing them it can be seen that the CP-SAT schedule significantly out performs the manual schedule. The CP-SAT would be expected to detect 48 objects on average compared to only 28 by the manual schedule. Figure 4(a) shows the memory use of the two schedules, and the cause of the difference in effectiveness can be inferred from them. In Figure 4(a) we can see that the manual schedule rapidly fills up the memory, and keeps it saturated throughout the scheduled period. The CP-SAT defined schedule takes longer to fill, waiting for higher value observation opportunities. A manual heuristic which prioritises the best observation opportunities while avoiding violating the maximum memory constraint would likely achieve comparable results to the CP-SAT schedule but would increase the complexity of deriving the heuristic rules.

While the amount of processed data exceeds the link budget if only the detections which contain an object are downlinked the link budget and time scheduled to downlinking will be sufficient. The results for both show that STRATHcube will be able to achieve a reasonable number of detections for a technological demonstrator for the non-real time onboard processing case.

Off-board Processing

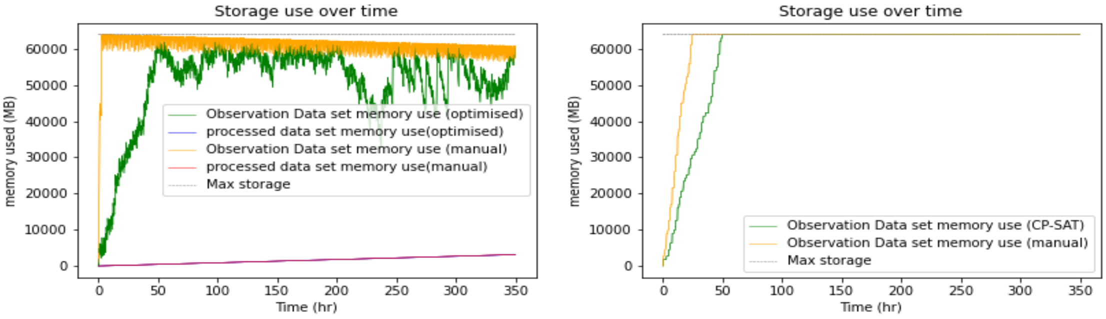
The effect of not being able to process and observations on board is a massive drop off in the number of detections. This can be seen in Figure 3(b) with both the manual heuristic and CP-SAT method achieving a very small number of expected detections over the period. This cause of this is seen in Figure 4(b) where the observation data rapidly saturates the memory, stopping any additional observations from occurring. The extremely low rate at which data is downlinked means that it is not possible for the memory to be cleared. In the case where no information is known about which observation data sets contain targets the mission will likely fail to detect targets. If instead the observation is scheduled at times when it is known a detectable object will be within the forward scatter zone then a slight improvement could be achieved. The link budget would allow for 17.1 seconds of unprocessed observation data to be downlinked over the 14 day period. This may make it possible for the a few demonstrations of the space-borne PBR technology for detecting



(a)

(b)

Figure 3: Number of detections for the schedules for non-real time processing case (a), off-board processing case (b)



(a)

(b)

Figure 4: Graph of memory used by unprocessed and processed data sets for non-real time processing case (a), off-board processing case (b)

space objects but would severely limit the mission. When compared with the results for non-real time processing, off-board processing performs more than 10 times worse.

Real Time Processing Condition

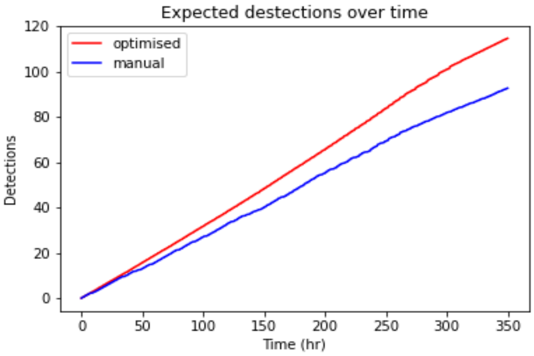


Figure 5: Detections with real time processing

Figure 5 shows the detections expected for the real time processing case. This shows the highest number of expected detections with the optimised method achieving 115 and the manual method achieving 88. This is more than double that achieved by the non-real time processing condition. If only successful observations are downlinked then the amount of memory to downlink will be 3.68MB, well within the 8MB possible to downlink with the schedule. This shows that this is clearly an ideal case for operation, however it requires either increasing the processing power or decreasing the number of FLOPs to carry out signals processing by 1 order of magnitude compared to the non-real time onboard processing case. This is a significant requirement to improve the number of objects detected by 2.4 times the non-real time processing case.

ISS Deployment Condition

For the situation where the satellite is deployed from the International Space Station (ISS) the results were found for the non-real time onboard processing case and the number of detections are shown in Figure 6. These show a significant improvement over the Saxavord launch condition and therefore it would be desirable to deploy from there rather than Saxaford.

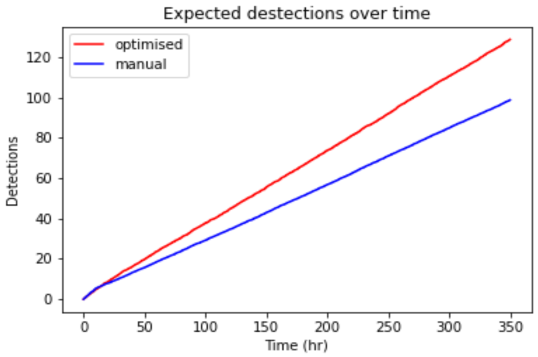


Figure 6: Detections Expected for ISS deployment with non-real time onboard processing

5 Conclusion

In this paper we have investigated the effectiveness of STRATHcube as a technological demonstrator under different design conditions and how the satellites scheduling effects its detection performances using a manual heuristic and the CP-SAT solver. A model of the expected number of objects to be detected has been created. This has been used with schedules under different design constraints to find the number of objects expected to be detected under three different circumstances: simulating non-real time onboard processing, real time onboard processing, and off-board processing. It demonstrated the effectiveness of using constraint programming for optimising the offline schedule of STRATHcube, with respect to a manual heuristic, and the number of objects detected for the different schedules. These prove that STRATHcube will be able to demonstrate the PBR technology for detecting space debris, to different levels of effectiveness under the different design circumstances.

6 Future Work

Future work in this area should look into developing different models for the number of objects either using inhomogeneous distribution of debris, or modelling all potential objects that could be detected. It should also explore alternative methods of defining the schedule that could potentially provide higher fidelity schedules while remaining computationally feasible. Improving the manual heuristics used should also be investigated as the ones developed as part of this investigation were sub-optimal in their approach.

Acknowledgements

The work was carried out with the support of the Engineering and Physical Sciences Research Council and at the University of Strathclyde. Thanks is given to Professor Massimiliano Vasile and Dr. Andrew R. Wilson for leading the STRATHcube project and all the students who have been involved in its development.

References

[1] A. Murtaza, S. J. H. Pirzada, T. Xu, and L. Jianwei, “Orbital debris threat for space sustainability and way forward (review article),” *IEEE access*, vol. 8, pp. 61 000–61 019, 2020.

[2] ESA office of space debris, “ESA’s annual space environment report,” 2022.

[3] C. Jenkins, L. Creed, J. Graham, S. Kirk, D. McGarrity, G. Stewart, R. Hope, A. R. Wilson, and M. Vasile, “Strathcube : a student cubesat that encourages the sustainable usage of space,” in *72nd International Astronautical Congress*, ARE, October 2021. [Online]. Available: https://strathprints.strath.ac.uk/78386/

[4] L. Creed, J. Graham, C. Jenkins, S. Diaz Riofrio, A. R. Wilson, and M. Vasile, “Strathcube : the design of a cubesat for space debris detection using in-orbit passive bistatic radar,” in *72nd International Astronautical Congress*, ARE, October 2021. [Online]. Available: https://strathprints.strath.ac.uk/78388/

[5] M. A. Richards, J. A. Scheer, and W. A. Holm, *Principles of modern radar: Basic principles*. Edison, NJ: SciTech, 2010.

[6] N. J. Willis, *Bistatic radar*. Raleigh, NC: SciTech, 2005.

[7] M. Liu, Z. Zhang, Y. Chen, S. Zheng, and J. Ge, “Forward scatter radar meets satellite: Passive sensing of aerial target using satellite communication waveforms,” *Remote sensing (Basel, Switzerland)*, vol. 14, no. 6, p. 1375, 2022.

[8] I. Theodorou, C. Clemente, and M. Vasile, “Space object detection and characterisation with a passive space-borne bistatic radar,” in *68th International Astronautical Congress, IAC 2017*, ser. Proceedings of the International Astronautical Congress, IAC. AUS: International Astronautical Federation (IAF), September 2017, pp. 341–345. [Online]. Available: https://strathprints.strath.ac.uk/76391/

[9] A. J. Vazquez Alvarez and R. S. Erwin, *An Introduction to Optimal Satellite Range Scheduling*, ser. Springer Optimization and Its Applications. Cham: Springer International Publishing AG, 2015, vol. 106.

[10] F. Rossi, P. V. Beek, and T. Walsh, *Handbook of Constraint Programming*. Pisa, Italy: Elsevier, 2006.

[11] V. J. Riot, L. M. Simms, and D. Carter, "Lessons learned using iridium to communicate with a cubesat in low earth orbit," *Journal of Small Satellites*, vol. 10, no. 1, 2 2021.

[12] "Pulse larsen datasheet." [Online]. Available: https://www.mouser.co.uk/datasheet/2/447/W3227-2904381.pdf

[13] Google OR Tools. (2022) CP-SAT Solver. [Online]. Available: https://developers.google.com/optimization/cp/cp_solver

[14] "OBC-P3 datasheet," 2020. [Online]. Available: www.space-inventor.com

[15] "Arm cortex-m7 processor datasheet." [Online]. Available: www.arm.com

[16] A. Myakinkov, "Optimal detection of high-velocity targets in forward scattering radar," in *2005 5th International Conference on Antenna Theory and Techniques*, 2005, pp. 345–347.