NASA/TM—2019-220226

Interference Mitigation Using Cyclic Autocorrelation and Multi-Objective Optimization

Mick V. Koch and Joseph A. Downey
Glenn Research Center, Cleveland, Ohio

July 2019

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program plays a key part in helping NASA maintain this important role.

The NASA STI Program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI Program provides access to the NASA Technical Report Server—Registered (NTRS Reg) and NASA Technical Report Server—Public (NTRS) thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counter-part of peer-reviewed formal professional papers, but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., "quick-release" reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at http://www.sti.nasa.gov

- E-mail your question to help@sti.nasa.gov

- Fax your question to the NASA STI Information Desk at 757-864-6500

- Telephone the NASA STI Information Desk at 757-864-9658

- Write to:
  NASA STI Program
  Mail Stop 148
  NASA Langley Research Center
  Hampton, VA 23681-2199

NASA/TM—2019-220226

Interference Mitigation Using Cyclic Autocorrelation and Multi-Objective Optimization

Mick V. Koch and Joseph A. Downey
Glenn Research Center, Cleveland, Ohio

National Aeronautics and
Space Administration

Glenn Research Center
Cleveland, Ohio 44135

July 2019

Level of Review: This material has been technically reviewed by technical management.

Available from

NASA STI Program
Mail Stop 148
NASA Langley Research Center
Hampton, VA 23681-2199

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
703-605-6000

This report is available in electronic form at http://www.sti.nasa.gov/ and http://ntrs.nasa.gov/

Interference Mitigation Using Cyclic Autocorrelation and Multi-Objective Optimization

Mick V. Koch and Joseph A. Downey
National Aeronautics and Space Administration
Glenn Research Center
Cleveland, Ohio 44135

Abstract

Radio frequency interference on space-to-ground communications links can degrade performance and disrupt the transfer of critical data. These interference events become increasingly likely as more users enter the spectrum, due in part to shared spectrum allocations and scheduling conflicts. If this interference could be detected and mitigated by an automated system, then link performance and reliability in these scenarios could be improved. This report describes the implementation and evaluation of an automated interference mitigation system that provides this functionality. The system uses Cyclic Autocorrelation (CAC) signal processing techniques to monitor the spectrum and detect interfering signals, and it applies a multi-objective optimization approach to mitigate interference by changing link parameters to continuously optimize the link. The implementation was evaluated to characterize its signal detection capabilities for various link qualities and to compare its link management performance to Adaptive Coding and Modulation (ACM) and Constant Coding and Modulation (CCM) when in the presence of randomized interference. In the latter evaluation, the interference mitigation system achieved the highest average throughput in each tested scenario. With these results, the proposed solution provides the groundwork for further automated link management capabilities and continued investigation into interference mitigation approaches.

1 Introduction

Space-to-ground communication links can be susceptible to co-channel interference during critical data transfers. This interference may be produced by both terrestrial and space-based sources, and it may arise due to the realities of shared spectrum allocations and scheduling or equipment configuration errors. This co-channel interference can degrade and disrupt the link and require manual intervention to identify the interferer and mitigate it. In these situations, an automated interference detection and mitigation system could lessen the performance degradation on the link. This report describes the investigation and evaluation of an automated system like this applied to a model of the link between the ground station at NASA Glenn Research Center (GRC) and the Space Communications and Navigation (SCaN) Testbed aboard the International Space Station (ISS). The system uses CAC signal

processing for interference detection and a multi-object optimization approach for mitigation. These topics are described in greater detail in the following sections.

The rest of this document is structured as follows. Section 2 provides further details on the motivations behind this research and presents work that the system builds upon. Section 3 details the implementation of the system and the algorithms that were applied. Section 4 describes the experiments used to evaluate the system. Section 5 then presents the results of these experiments. Finally, Sections 6 and 7 reiterate the accomplishments of the research and detail ways the system could be improved in further revisions.

2 Background

This section will provide some context for the investigation of the interference mitigation system, present work on some of the technologies that were integrated, and finally detail the assumptions that limit the system's scope.

2.1 Past examples of interference

Part of the motivation for investigating automated interference mitigation came from unexpected interference events that affected space-to-ground experiments at GRC. This type of scenario is illustrated in Figure 1. Although these occurrences were relatively low-impact due to the experimental nature of the SCaN Testbed, they illustrate that interference does happen in real-world scenarios. One such event occurred in 2016 during a Space Network communciation event with the GRC S-band ground station. During the experiment, the link became degraded and an interfering signal was observed on the spectrum analyzer overlapping with NASA's signal [1]. The source of the interference is unknown, but is believed to have been terrestrial communication system whose signal leaked into the antenna lobes.

Other cases of interference have occurred during SCaN Testbed experiments when the GRC ground station was communicating directly with one of the testbed radios on the ISS. While the experiment was ongoing, an interfering signal appeared near the center of the assigned channel at a relatively low symbol rate, then a few minutes later the symbol rate increased. It was later determined that a visiting vehicle near the ISS began transmitting on the same channel. In both of these cases the interfering signals were modulated data, and in both cases there was empty spectrum left in the allocated channel. Ideally, an automated system could detect the cyclic properties of these signals and utilize the available spectrum to reduce degradation caused by the interferers.

2.2 Adaptive coding and modulation with DVB-S2

The system presented in this report builds on ACM and Digital Video Broadcasting Satellite - Second Generation (DVB-S2) work previously completed at NASA GRC. ACM, when applied to a communication link, continuously adjusts modulation and coding (modcod) in response to link performance feedback (in this case, measured symbol energy to noise power spectral density ratio (E_s/N_0.)) When link conditions

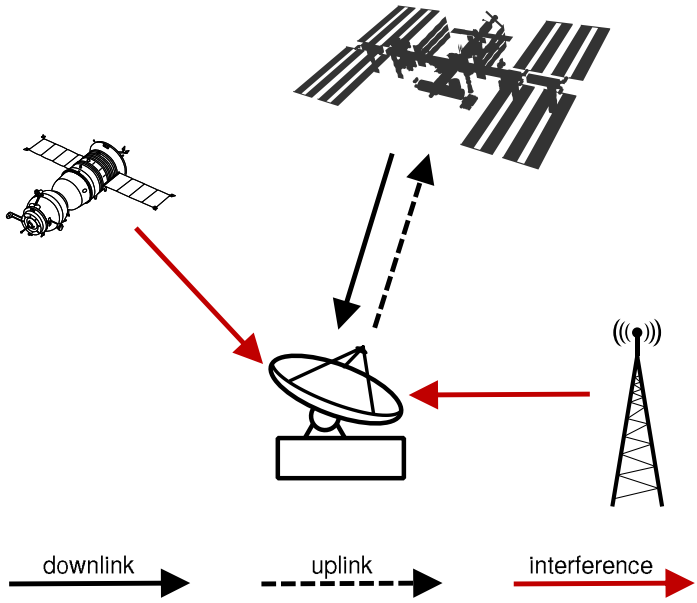

Figure 1. During space-to-ground communications with a craft such as the ISS, RF interference may appear from space-based or terrestrial sources.

are poor, ACM adjusts to robust modcods that increase error protection and maintain the link (at the expense of throughput), and when link conditions improve, ACM adjusts to higher-order modulations and minimal overhead coding schemes that maximize throughput (at the expense of error protection.) This technique was evaluated with an experiment on NASA's SCaN Testbed aboard the ISS and was found to significantly improve throughput on this dynamic space-to-ground link [2].

The implementation of ACM used in these studies and in this system uses the DVB-S2 standard [3] and the associated modulation and coding schemes. DVB-S2 is an open commercial standard developed by the European Telecommunications Standards Institute that provides the features required by ACM: specifically, a frame header with signalling to indicate the current modcod and a well-defined set of modcods to choose from. Each modcod specifies a modulation order (one of QPSK, 8PSK, 16APSK, or 32APSK) and a Forward Error Correction (FEC) coding rate, and there are currently 28 modcods defined in the standard. The modcod can be changed on the granularity of a DVB-S2 frame, which is either 64,800 bits in normal frame mode or 16,200 bits in short frame mode [3]. In this experiment, the DVB-S2 standard is used in the manner suggested by the Consultative Committee for Space Data Systems (CCSDS) standard 131.3-B-1 [4].

Previous work with DVB-S2 at GRC led to the in-house development of a DVB-S2 transmit waveform implemented on a Software Defined Radio (SDR). This waveform was also used to provide the DVB-S2 transmitter in this system. In addition to on-the-fly adjustment of the modcod required for ACM, the waveform also supports adjusting the symbol rate, center frequency, pulse-shape filter rolloff, and output power via a feedback control message. This enables full control of the link parameters

by the interference mitigation system. On the ground, this waveform is paired with a network-based Remote Procedure Call (RPC) system to command these changes. On a space radio, it is paired with an additional waveform that provides a BPSK uplink path and uses the CCSDS Space Data Link Protocol to send the commands. This is necessary because, although the CCSDS 131.3-B-1 standard specifies how to use the DVB-S2 standard for space telemetry, it does not standardize a means for communicating modcod from ground to transmitter. Using these technologies, a dynamic link controller such as the system described in this report can be prototyped on a ground-based link and then be ported relatively easily to command a space-based radio.

2.3 Cyclic autocorrelation for signal detection

In contrast to reinforcement learning systems which might build up knowledge about the spectrum environment by trying out many parameter states [5], this system directly observes the spectrum through signal processing to form its knowledge. In particular, a CAC approach is applied in order to detect interfering signals and estimate the symbol rate and carrier frequency for each detected signal. Unlike a simple spectrum sweep, this processing takes advantage of periodic features and allows us to pick out individual signals even in the presence of partial or full overlap of frequency components in the spectrum.

This system uses the CAC function to estimate the symbol rate. This function is very consuming in both compute and memory resources, but can be computed efficiently if modeled in terms of a Fast Fourier Transform (FFT), which can use heavily-optimized libraries like `fftw` [6]. The typical [7] expression for the symbol rate detection CAC is given by

$$f_R(x) = \left| \sum_{n=0}^{N-1} |x[n]|^2 \exp\left(-j2\pi nR/f_s\right) \right| \tag{1}$$

where $R \in \{R_0, R_2, \ldots, R_{M-1}\}$ are the symbol rates (in Hz) to be investigated. If these rates are spaced evenly from DC to $f_s/2$ with resolution $r = (f_s/2)/(M-1)$, such that $R_k = kr$, then this expression can be modeled into the typical FFT equation,

$$X_k(z) = \sum_{n=0}^{N-1} z[n] \exp\left(-j2\pi nk/N\right) \qquad k = 0, 1, \ldots, N-1 \tag{2}$$

by taking $f_R(z) = |X_k(z)|$, where $z[n] = |x[n]|^2$ and $N = f_s/r$. With this in mind, we can choose our capture size N appropriately, perform preprocessing to obtain $z[n]$, feed the data into a FFT library, then perform postprocessing to obtain the final result $f_R(x)$. This particular algorithm will be referred to as CAC-FFT in the rest of the report.

Although not currently used in this system, there are also other CAC-based functions that exist for estimating carrier frequency and detecting modulation orders. We performed some preliminary analysis of a "center frequency preprocessor"

function [7] to estimate carrier frequency but found it very sensitive to modulation order. As an alternative, we implemented carrier frequency detection using the symbol rate CAC, as detailed in Section 3.5.

2.4 Assumptions and limitations

We consider the interference mitigation system described in this report to be the first version and, in order to allow a reasonable development time, the scope of functionality was correspondingly limited. This section describes these limiting assumptions and provides an idea how later stages of development could increase functionality by relaxing these limitations. The following set of assumptions relate to the general hardware and spectrum environment the system is running in:

- The main signal has full ability and authority to move around within a typical [8] 6 MHz channel allocation.
- The system is running in a downlink scenario, where the transmitter is in space and the receiver is on the ground.
- The receiver provides accurate E_s/N_0 metrics at a sufficient update rate. This is necessary to ensure good ACM performance and action selection.
- The command uplink between ground station and spacecraft always succeeds in delivering commands to the transmitter.
- The noise floor is flat across the bandwidth of the channel.

The next set of assumptions relate to the interference scenarios affecting the system:

- The channel is affected by no more than one interfering signal at a time.
- The interfering signal is digitally modulated with some symbol rate. This is consistent with the premise that the interferer is a data signal also attempting to transfer some bits and has begun occupying the same channel spectrum, accidentally or otherwise.
- The interferer's carrier frequency and symbol rate are constant for a duration on the order of several seconds. This matches the previous real-world events that have been discussed in Section 2.1. As a result, the system is currently unable to handle signals that rapidly sweep frequency, symbol rate, or other parameters, which may occur in jamming scenarios or other less-benign situations.
- The interfering signal is not affected by any multipath fading, Doppler, or other channel effects seen by the main signal. This models, for example, a fixed ground-based signal interfering with a fixed ground station.

3 System Description

The interference mitigation system is composed of several algorithms that allow it to detect and respond to interference and generally optimize the link based on a given set of mission objective weights. The system runs in real-time, continuously ingesting link information from the receiving modem and spectrum information from the spectrum monitoring subsystem and feeding this data into an ACM loop and a mitigation subsystem. The ACM loop acts as a lightweight link optimizer for quick modcod adaptation, and the mitigation subsystem handles larger mitigation actions such as changing center frequency and symbol rate. This subsystem uses some straightforward algorithms to decide if such mitigation should take place and, if so, what mitigation actions are available to consider. These actions are then ranked using a multi-objective weighted-sum algorithm, and the best-ranked action is chosen and applied to the link by the link controller. This software runs as part of an experiment framework on a hardware lab testbed, which simulates a real-world link and allows introduction of interference and channel impairments. These topics will be discussed in more detail in the following sections.

3.1 Implementation overview

This system is implemented mostly in Python, with some performance-critical elements written in compiled code. As shown in Figure 2, it uses two Operating System (OS) processes to allow concurrent execution of subsystems, and further leverages the asynchronous input/output (`asyncio`) functionality built into modern Python to allow concurrent I/O and event loop tasks within the same process. Interprocess communication channels are implemented with `zeromq` sockets that also live on the `asyncio` event loop, and Hypertext Transport Protocol (HTTP) clients can connect to various embedded HTTP servers to retrieve information about the current state, subscribe to Server-Sent Events (SSE) event streams, and execute commands and configuration changes. A web-based Graphical User Interface (GUI), shown in Figure 3, has been created to provide a streaming dashboard for the system using these interfaces.

3.2 Multi-objective weighted-sum method

There may be many ways to respond to an interference scenario, and in general the ideal response for one mission may not be the ideal response for other missions: each mission may have different objectives. Because of this, the interference mitigation problem is modeled as a multi-objective optimization problem, and the weighted sum method [9] is applied as an initial, lightweight solution. This allows the user to weight the importance of each objective relative to the others for the target mission, and the system uses those weights to determine a solution most optimal for the objectives (within the relatively simple model of the linear weighted sum.) Since the objectives may be conflicting, such as maximizing throughput and minimizing bandwidth, this also allows the user to weight the outcome of these tradeoffs. This multi-objective model has been used previously as a link optimizer for an experiment

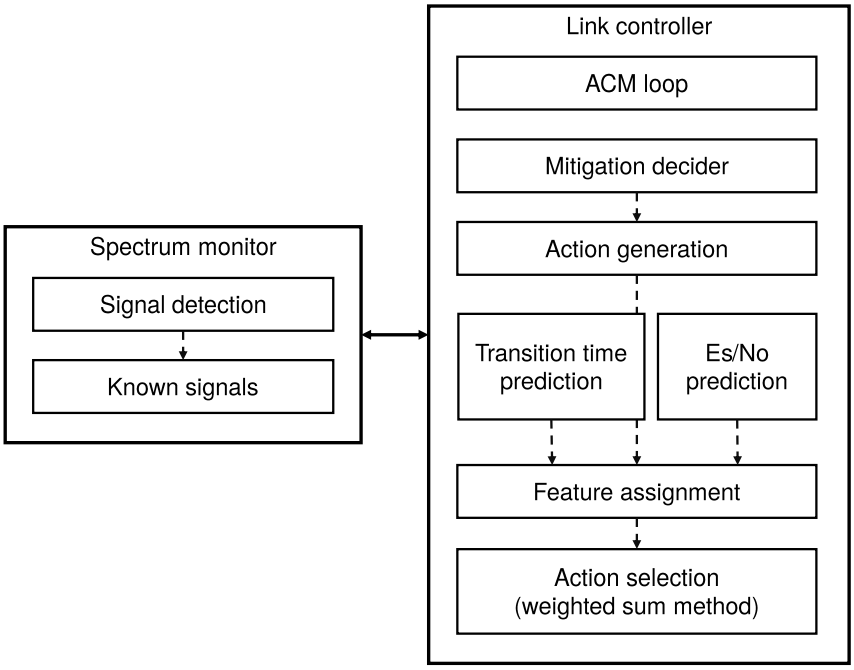

Figure 2. The software stack of the interference mitigation system is divided into two main processes that run concurrently.

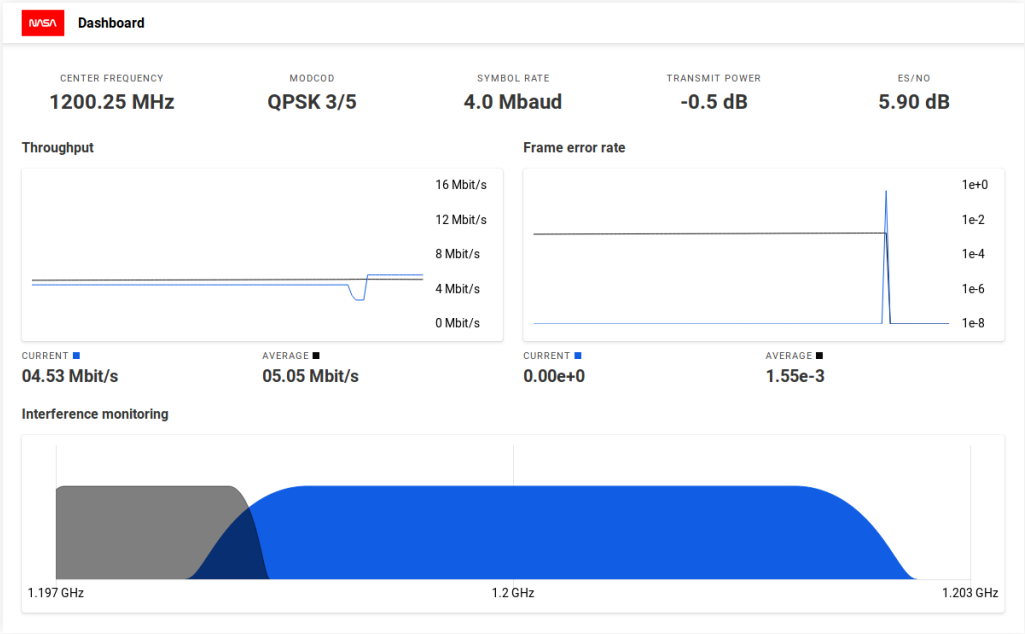

Figure 3. A web-based GUI uses HTTP APIs and event streams embedded in the system to provide a streaming dashboard.

on NASA's SCaN Testbed, although that system used reinforcement learning and neural networks to optimize for the user objective weights [10, 5].

As part of the interference mitigation system, this multi-objective optimization is applied at each mitigation instant, when the system has decided it needs to take some action to mitigate interference or optimize the link (this is detailed in Section 3.6.) Using the algorithms described in Section 3.7, it first generates a set of candidate actions $A = \{a_1, a_2, \ldots, a_M\}$. Each action a_i is then associated with a vector of feature scores $\mathbf{x}_i = [x_i(1), x_i(2), \ldots, x_i(N)]$. Then the overall score for each action is computed with the typical weighted-sum expression [9],

$$f(\mathbf{x}_i; \mathbf{w}) = \mathbf{w}^T \mathbf{x}_i = w(1)x_i(1) + w(2)x_i(2) + \cdots + w(N)x_i(N) \tag{3}$$

where $\mathbf{w} = [w(1), w(2), \ldots, w(N)]$ are the user-configured feature weights. Finally, the action with the highest score,

$$a = \underset{a_i \in A}{\operatorname{argmax}}\, f(\mathbf{x}_i; \mathbf{w}) \tag{4}$$

is chosen and applied to the link. The complete set of these features as well as feature processing is detailed in the next section.

3.3 Features and weights

There are currently 5 features available for multi-objective weighting:

Throughput Maximize data volume through the link. This generally results in higher bandwidth usage and transmit power.

Occupied bandwidth Minimize spectrum usage. This comes at the cost of data throughput but can allow multiple users to share a channel.

Transmit power Minimize transmit power, which may correspond to lower power draw on a spacecraft. This generally reduces potential throughput.

Lost potential Minimize delays in the data stream due to time spent transitioning between configurations. In particular, requesting spectrum outside the assigned channel would take significantly longer than staying within the channel. This is a unitless metric that differs from lost throughput in that it becomes more significant as the data buffer empties.

Glitch-free transition Avoid disruptions in the data stream caused by actions that require resynchronization between the transmitter and receiver. This may be desirable in real-time streaming applications.

After these feature values are computed for each action, they are normalized feature-wise across all actions to between 0 and 1. This brings all the features to the same range, which prevents one feature from numerically dominating the others and biasing the score as a result. After this processing, a value of 1 represents the most optimal value for that feature (depending on whether the feature is meant to be minimized or maximized), and a value of 0 represents the least optimal value. In contrast to this, the user weights can take any value to allow full control of how features are ranked.

3.4 Adaptive coding and modulation

The ACM loop runs in a background task and actively adapts the link modcod while the system is steady and not applying other mitigation actions. It monitors the E_s/N_0 measurements provided by the receiving modem, which are produced every 10 ms, and updates the link modcod any time the E_s/N_0 sufficiently changes. As in previous ACM implementations [2], each E_s/N_0 is applied to a lookup table with a 2 dB link margin in order to select the corresponding modcod. This table is created from characterization data published by the modem manufacturer that associates each modcod with a minimum E_s/N_0 required for quasi error free operation. Once a modcod is selected, it is sent over the simulated-delay uplink channel (as described in Section 3.8) to the transmitter.

3.5 Spectrum monitoring

The spectrum monitor task continuously monitors the communication channel of the link and notifies the link controller of active interfering signals. To do so it employs the CAC algorithm described in Section 2.3 and extends it with additional logic and processing as detailed in the rest of this section. This allows the task to recognize the presence of an interfering signal, estimate its symbol rate and carrier

frequency, and prune out the main transmitting signal as well as signals that are already known to be interferers.

The signal detection functionality of this task requires the use of several signal processing algorithms. Because these algorithms need considerable compute resources and concurrency, the task is isolated into its own process and communicates with the link controller via interprocess channels. To start the signal processing, a block of complex In-phase/Quadrature (I/Q) samples is captured from a spectrum analyzer monitoring the channel. The capture is sampled at a rate of 10 Msps, which results in a usable analysis bandwidth of 8 MHz (the theoretical bandwidth is 10 MHz, but filtering on the edges of the spectrum reduces this accordingly.) The capture size is determined by the CAC-FFT bin size, in order to allow evaluation of the FFT at full resolution without zero padding. With a bin size of 40 baud at this sample rate, the capture size ends up at 250 kilosamples.

Once these samples are captured, a coarse segmentation step is performed to locate areas of sufficient energy in the channel spectrum. The main purpose of this segmentation is to allow bandpass filtering of each segment, which helps isolate the signals contained within from the rest of the channel. To perform this step, a Welch Power Spectral Density (PSD) estimate is first generated for the channel. This PSD is then examined to find contiguous regions of power above a specified threshold. This threshold is currently set to 5 dB above the estimated noise floor, which trades off detection of lower-power signals for the detection of fewer false-positive segments. The processing of these false-positive segments was found to add considerable overhead to the signal detection execution time, so the current threshold was chosen to heavily decrease their likelihood.

Once this set of segments is generated, each one is processed using the following steps. First, the spectrum is shifted so the center of the segment lies at DC. Then, the spectrum is filtered using a dynamically-generated lowpass filter whose bandwidth is approximately equal to the bandwidth of the segment. These two steps ideally leave the spectrum containing only the components within the target segment. After this processing, the samples are evaluated with the CAC-FFT algorithm. The output of this is finally processed with a peak-finding algorithm to generate a set of detected symbol rates.

Each of these detected symbol rates is then investigated with the following processing to determine if it appears to be a legitimate detection and, if so, what the carrier frequency of the corresponding signal appears to be. This carrier frequency detection is implemented with a "sweeping" algorithm, where the segment containing the target symbol rate is systematically shifted and lowpass filtered from one band edge to the other, and a single-rate CAC for the target symbol rate is evaluated on the signal at each offset. This curve is then normalized by the signal power at each shift, and the peak value is taken as the result. This normalized peak is then compared to an empirically-measured threshold to determine whether the signal appears to be a legitimate detection and if so, the corresponding offset frequency of the peak is taken to be the signal's carrier frequency.

Due to the lack of filter rolloff information, the bandwidth of the lowpass filter used when sweeping is set to the target symbol rate itself (i.e., assuming no rolloff factor.) Additionally, although this algorithm is parallelized to take full advantage

of the processing cores available to it, it is still a very intensive routine. This fact must be traded off against the step size of frequency offsets over which the signal is evaluated. Since this step size determines resolution, a larger step size will decrease running time but also decrease detection accuracy. In the current implementation, a step size of 20 kHz is used, which leads to reasonable latency on the current hardware.

The result of all this processing is a set of symbol rate and carrier frequency pairs for each suspected interfering signal. Because these values typically don't match up exactly for the same signal over multiple evaluations, an additional postprocessing step is used to "merge" these signals with the current spectrum knowledge. To support this processing, the spectrum monitor maintains a set of known active interferers and also tracks the state of the main transmitting signal. Then, the symbol rate and carrier frequency of each new candidate signal is compared with the corresponding parameters of these known signals and, if both differences are within their respective thresholds, the candidate is discarded. These thresholds are currently configured to 10 kbaud and 250 kHz, respectively. Once a new signal is detected in this way, it is added to the set of known signals. Similarly, if one of the known signals isn't present in the detected signals, it is removed from the set. In both cases, the link controller is notified of the current set of known interfering signals.

Along with this communication, the spectrum monitor task also coordinates with the link controller on state transitions, only allowing them to proceed while no sample capture is ongoing, and preventing captures until the transition has been completed. This, combined with the state updates broadcast by the link controller, ensures that the expected state of the main signal at each sample capture is synchronized with the true state present on the channel.

3.6 Mitigation decider

The mitigation decider subsystem monitors the available information about the environment to decide when to initiate a mitigation action. When the link controller receives new information about the environment, such as interfering signals and state changes, it notifies the mitigation decider of this information. Then, as part of its main loop, the link controller queries the decider to determine whether it should enter the mitigation process. If any of the following are true, then the decider will recommend mitigation:

- The current state differs from the expected state. Most importantly, this includes modcod and receiver synchronization lock. The expected state is updated before beginning a state transition, so state discrepancy can be due to causes such as state transition failure, inaccuracies in E_s/N_0 prediction, or channel fading.
- The current set of interfering signals has changed since the last query. Applying a mitigation action in this case allows the system to account for the appearance or disappearance of other signals on the channel.
- The watchdog timer has expired. This timer repeatedly activates on a specified interval, which ensures that system regularly reevaluates its current situation.

This can help bring the system back to an operational state in unforeseen situations that fail to trigger the other conditions. The interval is currently configured for 10 seconds since the previous mitigation instant.

In addition to these conditions, the decider also implements a "hang time" after state transitions during which state updates are ignored. This allows the receiver's E_s/N_0 estimation to stabilize and adds some hysteresis to the mitigation process. The hang time is currently configured to expire 3 seconds after each state transition.

3.7 Action generation, scoring, and selection

Once the system has decided it should take some mitigation action, known as the mitigation instant, it begins the mitigation process. To do this, it uses current knowledge of the environment to generate a set of actions it can take, then ranks those and selects the most optimal as described in Section 3.2. The actions are generated with a brute-force algorithm, but even this naïve method is able to evaluate the whole decision space with sub-second runtime.

Each action is associated with a state that the system should transition to, which characterizes the transmitter parameters: symbol rate, center frequency, filter rolloff, transmit power, and modcod. The first step in generating these candidate states is to determine the set of candidate center frequencies f and bandwidths W. To start, the known interfering signals are subtracted from the channel spectrum to generate a sequence of "holes" where the main signal should have zero overlap with any interferers. This results in a set of (f, W) pairs that maximize the bandwidth in each of these spectrum areas. Additional candidate center frequencies are generated at regular intervals (currently 250 kHz) through the spectrum, and the corresponding bandwidth for each is taken as the maximum amount that will fit in the channel.

Then for each of these (f, W) pairs, a set of candidate symbol rate and rolloff combinations is generated. These combinations are produced with discretized variants of the parameters, with symbol rate choices being between 1 Mbaud and 4.5 Mbaud with a step size of 500 kbaud,

$$C_R = \{1\,\text{Mbaud}, 1.5\,\text{Mbaud}, 2\,\text{Mbaud}, \ldots, 4\,\text{Mbaud}, 4.5\,\text{Mbaud}\}$$

and the rolloff choices being those available in DVB-S2, $C_\alpha = \{0.2, 0.25, 0.35\}$. Then a combination of $C_R \times C_\alpha$ is used if the pulse shape filtered bandwidth is contained within the limits of W.

Each of these combinations is then assigned a transmit power of $p = 0\,\text{dB}$ relative to the maximum transmit power, which results in the tuple (f, R, α, p). If the receiver currently has synchronization lock, then additional tuples are also generated based on the current state. These parameter combinations keep (f, R, α) constant and vary p and also keep (f, p) constant and vary (R, α) within the channel bandwidth. In the former case, the transmit power is varied from $-6\,\text{dB}$ to $0\,\text{dB}$ in $0.5\,\text{dB}$ steps. This choice of transmit power is restricted to this case so that the system can ensure it is in a stable, established state before optimizing power levels.

All these tuples are then evaluated with an E_s/N_0 prediction algorithm to generate the expected E_s/N_0 delivered by the target state. This takes into account the

current E_s/N_0 as well as the change in transmit power, symbol rate, and interferer overlap. This overlap E_s/N_0 estimator was determined empirically using a large set of scenario captures with random signal parameters. Due to the lack of signal power estimation, it currently assumes that all signals are equal power. This limitation leads to an overestimation of the expected E_s/N_0 if the interferer is higher power and an underestimation if the interferer is lower power, both of which can cause a suboptimal decision. Once the expected E_s/N_0 has been estimated, this value is processed by the ACM subsystem to determine which modcod would be applied in the target state. This final parameter completes the full state tuple for the action.

Once action generation has completed in this way, each action state is used to compute the corresponding feature scores. Finally, as described in Section 3.7, each action is given an overall score based on these features and the configured user weights, then the most optimal action is chosen and applied by the link controller, as described in the next section.

3.8 Link controller

The link controller implements the main loop of the interference mitigation system. It ingests updates from the receiving modem and spectrum monitor, coordinates state transitions, and applies ACM and mitigation actions. To simulate the ground-space uplink path, a configurable software channel delay is is applied to all ACM and state updates. This delay is currently configured for 40 ms to mimic a real-world ISS link [2]. Additionally, for state updates, it implements the data stream pause and resume functionality found on the DVB-S2 waveform. This functionality allows the controller to pause data transmission on the transmitter just before applying new state parameters, then resume data transmission just after. This reduces the chance of frames being dropped during the transition due to loss of synchronization between the transmitter and receiver. If the link does end up being disrupted during a transition such that that the receiver fails to achieve synchronization lock after a configurable timeout period, then the system abandons the current action and immediately selects another. This timeout is currently configured for 10 seconds.

3.9 Lab testbed

This system was developed and evaluated on a hardware lab testbed that simulates a satellite telemetry DVB-S2 link (such as the one described in Section 2.1) with programmable interference and fading effects. The main hardware components are shown in Figure 4 along with the software components that make use of them. This includes the software described in earlier sections as well as software used by the testbed experiment framework: the channel simulator task for applying link impairments and the statistics collection task for evaluating performance. Independent of these, the link controller interacts with a DVB-S2 transmitter through a simulated (as described in Section 3.8) uplink channel. This transmitter is a SDR chassis running a custom-developed DVB-S2 waveform that supports on-the-fly configuration of modcod, symbol rate, and other parameters. It currently outputs a stream of Pseudorandom Binary Sequence (PRBS) data over the Radio Frequency (RF) link,

with optional CCSDS Advanced Orbiting Systems (AOS) framing [11].

This link then passes through a variable attenuator and is split between a commercial DVB-S2 modem and a spectrum analyzer. The variable attenuator is controlled by the channel simulator task and simulates a fading profile on the channel. In addition to this, the resulting signal is also summed with an interferer and a noise source – also controlled by the channel simulator task – to complete the channel impairments. The interferer is a commercial DVB-S2 modem that provides many knobs for custom signal generation. The noise source is a white noise generator passed through a variable attenuator, which allows adjustment of the channel noise floor.

At the other side of the link are the receiving modem and spectrum analyzer. The receiving modem streams decoded DVB-S2 frames out the "transport bypass" port to the statistics collection task. This task synchronizes to the Attached Sync Marker (ASM) sequence and uses the decoded CCSDS AOS frame counter to infer throughput and drop rates. In addition, this modem streams Received Signal Strength Indicator (RSSI) information, which includes a synchronization lock indicator and estimated E_s/N_0, to the link controller. The spectrum analyzer sees the same signal as the receiver and provides I/Q captures of the channel to the spectrum monitor task.

4 Experiment Plan

The interference mitigation system was evaluated using the following three experiments described in this section. The first two experiments examine the system's signal detection capabilities, while the third experiment examines the interference mitigation system as a whole.

4.1 Characterization of signal detection algorithm with one signal

The purpose of this experiment was to determine how the signal detection algorithm performs at various Signal to Noise Ratios (SNRs), as well as to investigate how this performance is affected by modulation and filter rolloff parameters of the signal. The lab testbed described in Section 3.9 was used to perform this test, with the main DVB-S2 transmitter disabled and the interferer and noise floor varied as needed for each scenario. The signal detection subsystem was then evaluated on each scenario using I/Q samples from the spectrum analyzer. These scenarios were generated using all possible combinations of modulation, rolloff, and SNR under test, $C_M \times C_\alpha \times C_{SNR}$, where

$$\begin{aligned} C_M &= \{\text{QPSK}, \text{8-PSK}, \text{16-APSK}, \text{32-APSK}\} \\ C_\alpha &= \{0.20, 0.25, 0.35\} \\ C_{SNR} &= \{0\,\text{dB}, 1\,\text{dB}, \ldots, 24\,\text{dB}, 25\,\text{dB}\} \end{aligned}$$

Then for each of these combinations, 40 trials were conducted using a random symbol rate and center frequency for each trial. These random parameters were chosen in the following way:

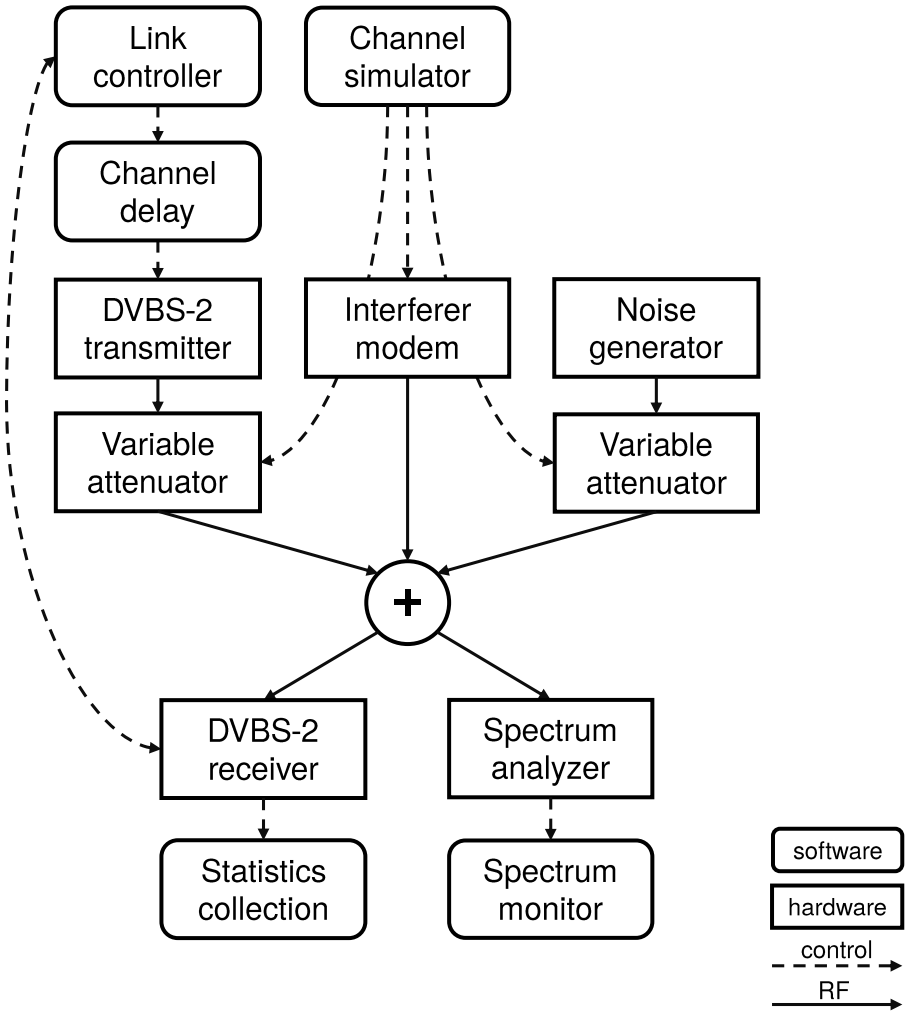

Figure 4. The lab testbed, as outlined in this diagram, was used for developing and evaluating the interference mitigation system. It is composed of several hardware components and a software experiment framework that allows simulation of channel impairments and collection of performance statistics.

- Symbol rate chosen with uniform probability between 100 kbaud and 4.5 Mbaud at 1 baud resolution
- Center frequency chosen with uniform probability between the channel bounds at 1 Hz resolution

For each of these trials, the signal detection subsystem was allowed to evaluate the channel for a fixed period of 10 seconds, and the first detection output was taken as the result. If the algorithm detected more than one signal or exceeded the timeout period, then the result was stored as a failed detection (i.e., infinite symbol rate error and infinite carrier frequency error.) Otherwise, the detected symbol rate and carrier frequency were stored for later comparison with the true values. These results were then sliced up along various dimensions to determine the effects of different parameters.

4.2 Characterization of signal detection algorithm with two signals

This experiment was carried out to determine how the signal detection algorithm performs at detecting an interfering signal in the presence of a co-channel main signal transmitted at various relative power levels, known as Signal to Interference Ratio (SIR). The lab testbed was used as in Section 4.1 to perform this experiment, with the addition of the DVB-S2 transmitter to provide the channel's primary signal. The testbed's noise source was disabled, which resulted in an SNR of 20 dB for a 4.5 Mbaud signal at maximum transmit power. The test scenarios for this experiment were generated so that there were 40 trials of random parameters for each SIR choice in $\{-10\,\text{dB}, -9\,\text{dB}, \ldots, 9\,\text{dB}, 10\,\text{dB}\}$. These random parameters were chosen independently for the main and interfering signals from the following distributions:

- Modulation chosen from $\{\text{QPSK}, \text{8-PSK}, \text{16-APSK}, \text{32-APSK}\}$ with uniform probability
- Symbol rate chosen with uniform probability between 100 kbaud and 4.5 Mbaud at 1 baud resolution
- Center frequency chosen with uniform probability between the channel bounds at 1 Hz resolution
- Filter rolloff chosen from $\{0.20, 0.25, 0.35\}$ with uniform probability

Each of these scenarios was then evaluated by the signal detection subsystem to determine its performance. The algorithm was allowed to run in each case for a fixed timeout period of 10 seconds, and the first detection output was taken as the result. If the timeout expired without a result or the algorithm detected more than 2 signals, then the result was recorded as a failed detection. Otherwise, the detection result was stored for comparison with the true scenario parameters.

These results were interpreted to measure the detection performance with respect to the interfering signal, as the parameters of the main signal are known and controlled by the system. This came into play when the algorithm only produced a single detected signal for a scenario. In this case, the detected parameters were

compared with the interferer's true parameters to determine the performance. Otherwise, when two signals were detected, one was compared against the main signal and the other was compared against the interferer to determine if the interferer was detected and if the other signal would be recognized as the main signal (and not a spurious third signal.)

4.3 Evaluation of interference mitigation system

In contrast to the previous experiments that evaluated a specific subsystem in isolation, this experiment was performed to evaluate the overall interference mitigation system and compare its performance to alternative link management strategies. To achieve this, the system was applied to the DVB-S2 link on the lab testbed, along with two other alternative strategies. The link was then impaired with both fading and interference to determine the overall performance of each strategy.

These two alternative strategies used for comparison are ACM and a CCM waveform using QPSK with a rate ½ code. Because the interference mitigation system builds on top of ACM, evaluating ACM performance alone allows us to see the effects of the additional interference mitigation and multi-objective link optimization subsystems. Similarly, including the CCM QPSK rate ½ strategy allows us to see how the system compares to current practice.

The three link strategies under test were evaluated on several combinations of link fading profiles and simulated interference. One of the link profiles, as shown in Figure 5, was generated to simulate an ideal Direct-to-Ground (DTG) link. The DTG scenarios simulates a satellite passing over a ground station with some Additive White Gaussian Noise (AWGN) and no multipath effects. Along with these, three real-world fading profiles, shown in Figure 6, were also applied. These profiles are based on recorded power envelopes seen during overhead passes of the ISS when the SCaN Testbed ACM experiment was being conducted [2]. The profiles reflect a highly dynamic link, with varying path loss, obstruction losses, and multipath effects from the ISS structure. They generally follow the same pattern, with a front-lobe section at the beginning, followed by a dip and recovery due to ISS structural blockages, then finally heavy attenuation on transition to the antenna back-lobe [12]. For this evaluation test, additional noise was added to the profiles, yielding a maximum E_s/N_0 of around 12 dB with a symbol rate of 1 Mbaud.

Each of these fading profiles was combined with two interference scenarios: a baseline scenario with no interferer and a random scenario with a Markov-based interferer. This random interference scenario is based on a real-world interference event that occurred during a SCaN experiment, as described in Section 2.1, where the interferer first appeared about halfway through the experiment at a relatively low symbol rate, then shifted to a higher symbol rate some time later. This scenario attempts to capture the main characteristics of that event while also introducing some randomness into the interferer parameters and timing. To achieve this, a 3-state Markov model was applied using the transition matrix

$$P = \begin{bmatrix} 0.98 & 0.02 & 0.00 \\ 0.00 & 0.99 & 0.01 \\ 0.01 & 0.00 & 0.99 \end{bmatrix}$$

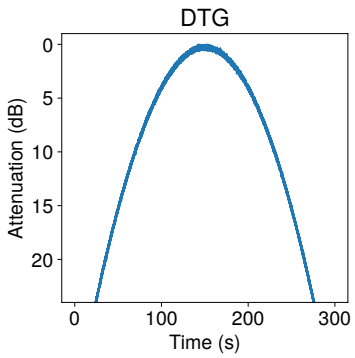

Figure 5. The synthesized Direct-to-Ground (DTG) profile simulates the link in an ideal overhead satellite pass.

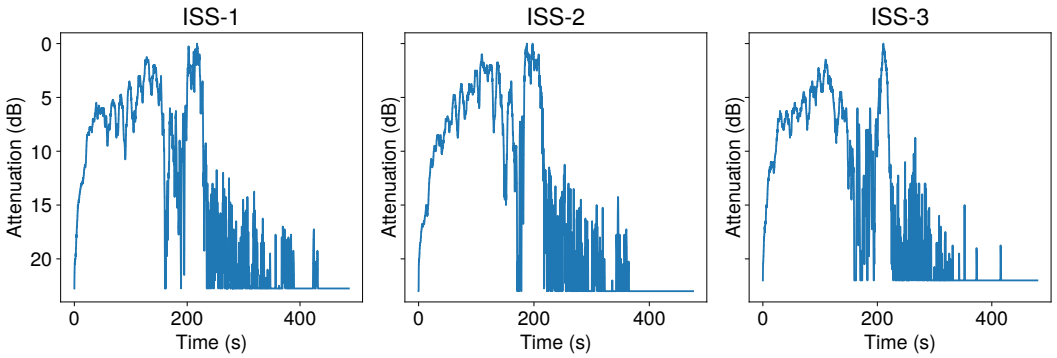

Figure 6. These real-world fading profiles were also used for evaluating the interference mitigation system. They were recorded during a previous [2] SCaN Testbed experiment that used an ISS-to-ground link.

The first state indicates that the interferer is disabled, and the other two states indicate that it is enabled with parameters selected as follows:

- Modulation chosen from $\{\text{QPSK}, \text{8-PSK}, \text{16-APSK}, \text{32-APSK}\}$ with uniform probability
- Symbol rate chosen with uniform probability at 1 baud resolution, within different bounds depending on the state: between 500 kbaud and 1.0 Mbaud for state 2, and between 2.5 Mbaud and 3.5 Mbaud for state 3
- Center frequency chosen with uniform probability between the channel bounds at 1 Hz resolution
- Filter rolloff chosen from $\{0.20, 0.25, 0.35\}$ with uniform probability
- Transmit power relative to the main signal fixed to 0 dB

This model is evaluated on a 1 second interval to determine if a state transition should occur based on the transition matrix probabilities. For this experiment, we applied 30 unique executions of the model to each fading profile. Each execution was allowed to run for the full length of the profile, and each link management strategy saw the same 30 executions. Along with this, we executed each fading profile for 30 trials with interference disabled.

These evaluation scenarios were applied to the link management strategies using the following parameters. In all cases, the DVB-S2 link was configured to enable pilot symbols and short frames, as previous experiments determined that pilots improve acquisition and tracking performance and short frames reduce latency in modcod changes [2, 1]. Additionally, AOS framing was enabled in order to collect performance metrics, using a frame size of 260 bytes (10 bytes of which are used by the ASM and frame header.) Finally, the interference mitigation system was configured to use the weights

$$\mathbf{w} = [100, 1, 1, 1, 15] \tag{5}$$

This weights **Throughput** by 100, **Glitch-free transition** by 15, and the other features by 1. As a result, the system heavily favors actions that improve throughput (at the expense of errors and spectrum usage) while also discouraging state transitions that would result in a low marginal throughput improvement.

The AOS frames transmitted across the link in these scenarios were used to evaluate the throughput and error rate performance of the various link management strategies. These frames contain a frame counter field, which can be tracked to infer the number of frames transferred over time. The counter also reveals gaps where AOS frames have been "dropped" due to loss of lock or DVB-S2 frame corruption. This information provides an error metric to contrast against the overall throughput. Note that the payloads of the AOS frames are not inspected due to processing constraints, so no error checking is performed at the application level. The result of this data collection is two metrics used for comparison of the strategies: throughput performance and error performance. Throughput performance is calculated to be the

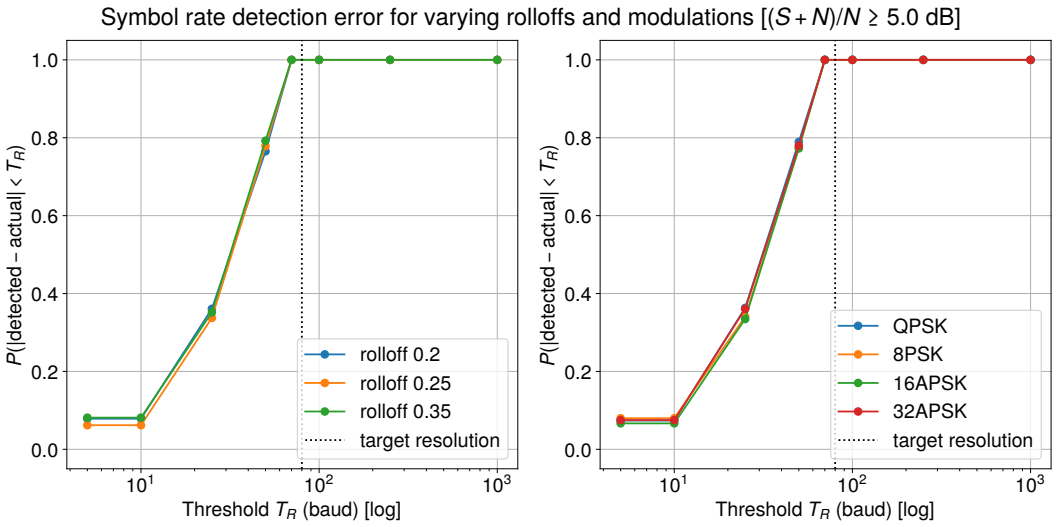

Figure 7. Filter rolloff and modulation scheme have practically zero effect on the performance of symbol rate detection.

overall number of frames that successfully made it across the link from transmitter to receiver, per unit time, which excludes dropped frames from the total. Error performance is calculated to be the total number of dropped frames detected at the receiver, as a percentage of the total frames sent by the transmitter.

5 Results

The following sections present the results of the experiments performed in Section 4, in respective order, and provide interpretation of the findings where applicable.

5.1 Performance of signal detection algorithm with one signal

The first several plots in this section characterize, based on the collected data, the probability that each detection algorithm will locate the target signal and produce output values for symbol rate and carrier frequency that differ from the actual values by less than the error on the x-axis. Then, for a given error magnitude on the x-axis, we can see the expected performance of the algorithm by the corresponding point on the y-axis.

By interpreting Figures 7 and 8 in this way, we can see that modulation and rolloff have little significant impact on the performance of symbol rate or carrier frequency detection. In the case of the former, there is practically zero effect from either parameter, and in the case of the latter, only rolloff slightly affects the performance before maximum performance is reached with an error of around 100 kHz.

With this in mind, we can then evaluate the error performance with all these parameters combined and averaged over to get the plots in Figure 9. These plots let us compare the overall symbol rate and carrier frequency detection performance relative to the "target resolution" configured for each algorithm. In the case of the

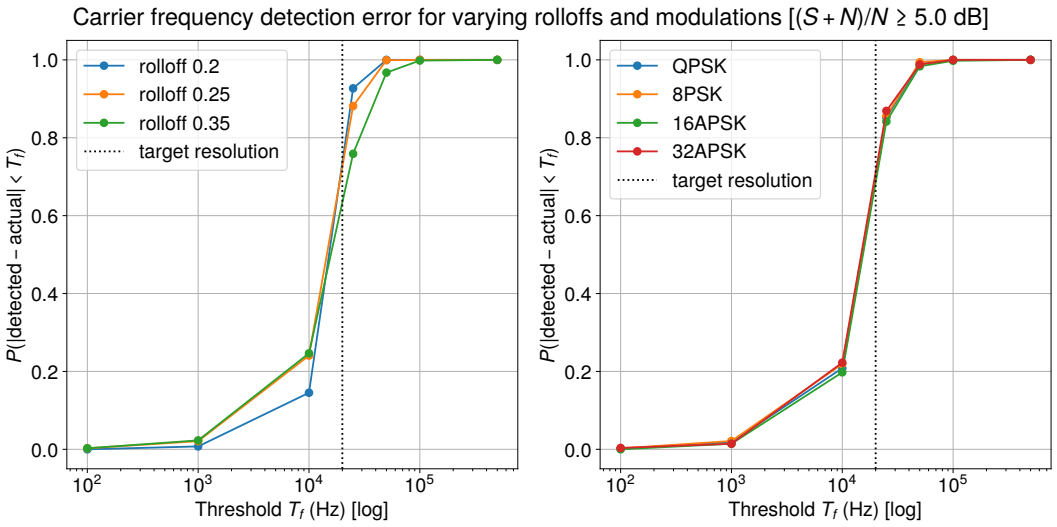

Figure 8. Rolloff has a slight effect on carrier frequency detection, while modcod again has practically no effect.

symbol rate detector, this resolution is equal to the bin resolution of the FFT used to compute the CAC, multiplied by two to account for the fact that peak bins need to be separated by at least one lower power bin in order to be recognized as peaks. In the case of the carrier frequency detector, however, the resolution is simply taken as the frequency step size used when determining which frequencies to shift the signal by in the detection routine.

Continuing with the comparison of these plots, we can see that symbol rate detection generally outperforms carrier frequency detection relative to the configured target resolution of each algorithm. In particular, if the acceptable detection error is taken to be equal to the target resolution, then symbol rate detection is able to meet this requirement in all cases. In contrast, the acceptable error for carrier frequency must be several times the target resolution in order to reach the same performance. One factor that may be contributing to this poorer performance is the lack of filter rolloff information available to the carrier detection routine. The lowpass filter applied after shifting the signal uses only the detected symbol rate to determine its bandwidth, so a variable portion of the useful spectrum is being removed depending on the signal's filter parameters. If this is a significant factor, then rolloff estimation may improve the algorithm's performance.

One final item to note about the results presented so far (as noted in the figure titles) is that they only include cases where the signal plus noise was 5 dB above the noise floor (corresponding to an SNR of 3.35 dB). Below this the performance degrades rapidly due to the initial segmentation step using a 5 dB threshold. This can be seen in Figure 9, where including the lower SNR cases clamps the maximum accuracy to under 90%. An alternative segmentation algorithm will likely improve these results in the lower-SNR cases.

Next, we can examine this data from a different angle and see how the detection algorithms perform, on average, for a given $(S + N)/N$. This interpretation of the

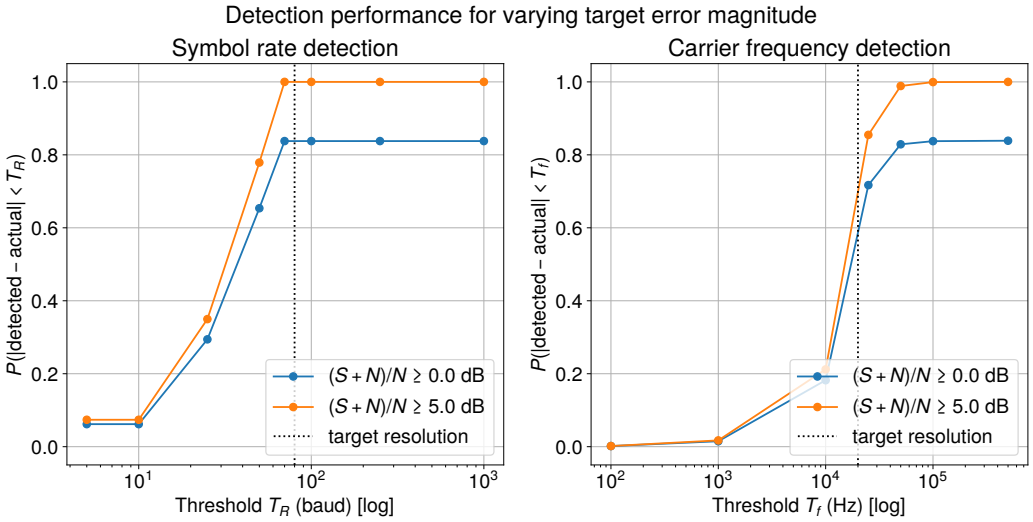

Figure 9. Symbol rate detection outperforms carrier frequency detection relative to the configured target resolutions. Additionally, including cases where $(S+N)/N < 5\,\text{dB}$ heavily impacts performance compared to Figure 9 due to the initial spectrum segmentation threshold.

results can be seen in Figure 10. These plots show that the symbol rate detection algorithm is able to produce an output within 80 baud of the true symbol rate when $(S+N)/N \geq 5\,\text{dB}$. In addition, the carrier frequency detection algorithm is able to produce an output within 80 kHz of the true frequency in almost all cases within the same $(S+N)/N$ range. These plots also show the performance drop off below the 5 dB threshold which, as mentioned before, is due to the initial segmentation step, and an alternative implementation will likely improve performance in these cases. A final takeaway from this figure is that when $(S+N)/N \geq 5\,\text{dB}$, enough signal power is apparently supplied for the algorithms to have generally consistent and flat performance through the higher SNRs. It seems likely that the artificial dropoff below 5 dB is hiding the curve leading up to this plateau.

5.2 Performance of signal detection algorithm with two signals

The results in this section characterize the performance of the symbol rate and carrier frequency estimation algorithms at detecting the parameters of an interfering signal in the presence of a main transmitting signal. The performance is measured in terms of the interferer because the parameters of the main signal are fully known to the interference mitigation system without the need for estimation. These results are then divided into positive and negative SIR groups as well as into overlapping and nonoverlapping groups. The SIR divide helps reduce the clutter of having many SIR curves on the same plot, but it also helps illustrate the effects of relative power on detection performance. The overlapping/nonoverlapping divide allows investigation into how overlap affects performance and also facilitates comparison with the results in Section 5.1. Due to the random nature of parameter choice in the ex-

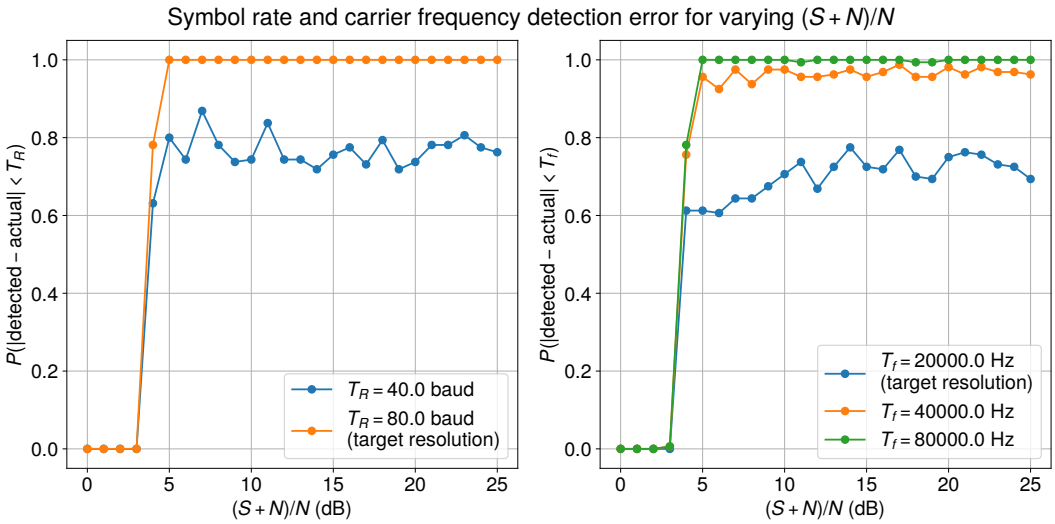

Figure 10. When $(S + N)/N \geq 5\,\text{dB}$, enough signal power is delivered to allow consistent performance from both algorithms at reasonable detection errors.

periment, there isn't an equal number of cases across this divide: 88% of the cases are overlapping and 12% are nonoverlapping. These results are plotted in the same format as the characterization plots described in Section 5.1.

Comparing Figure 11 to Figure 12, we can see that symbol rate detection generally has better performance relative to the configured target resolution compared to carrier frequency detection, as was also seen in Section 5.1. Next, comparing the nonoverlapping results in these plots to Figure 9, we see that both algorithms have similar performance to the one-signal case when the target signal is at a higher power (in the negative SIR scenario.) That performance is degraded, however, when the target signal has lower relative power, even in the case that the signals have no overlap. When the signals begin overlapping in this positive SIR scenario, we can see that the performance is further degraded. On the other hand, overlap appears to have almost no effect in the negative SIR case.

We expect both higher SIR and overlap to negatively affect detection of the interfering signal. In the case of higher SIR, the co-channel signal effectively reduces the SNR of the interferer when performing the CAC and can result in the interfering signal being completely overpowered. This sensitivity to power difference in mixed-signal CAC processing has been noted before [13]. The effect is compounded when the signals overlap, since the lowpass filter step that occurs after segmentation is unable to attenuate the higher-power signal to the same extent. Additionally, although the filter is able to improve matters in the nonoverlapping case (since the signals are processed in separate segments), the current normalization strategy likely results in the filtered signal having too little power to make it through the rest of the CAC processing successfully. This appears to be the cause for the performance degradation in these cases compared to Figure 9, and an alternative strategy may improve this.

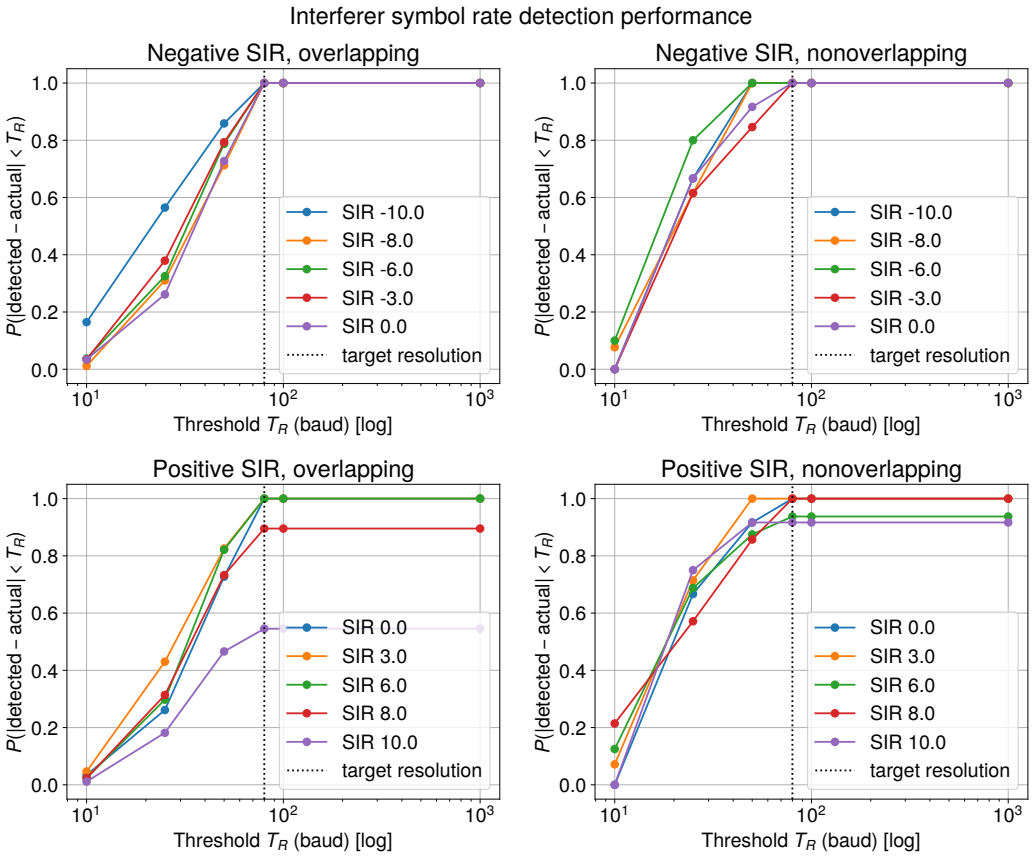

Figure 11. As expected, detection of the interfering signal is degraded when the main signal and interferer overlap and moreso when the interferer becomes weaker in relative signal power.

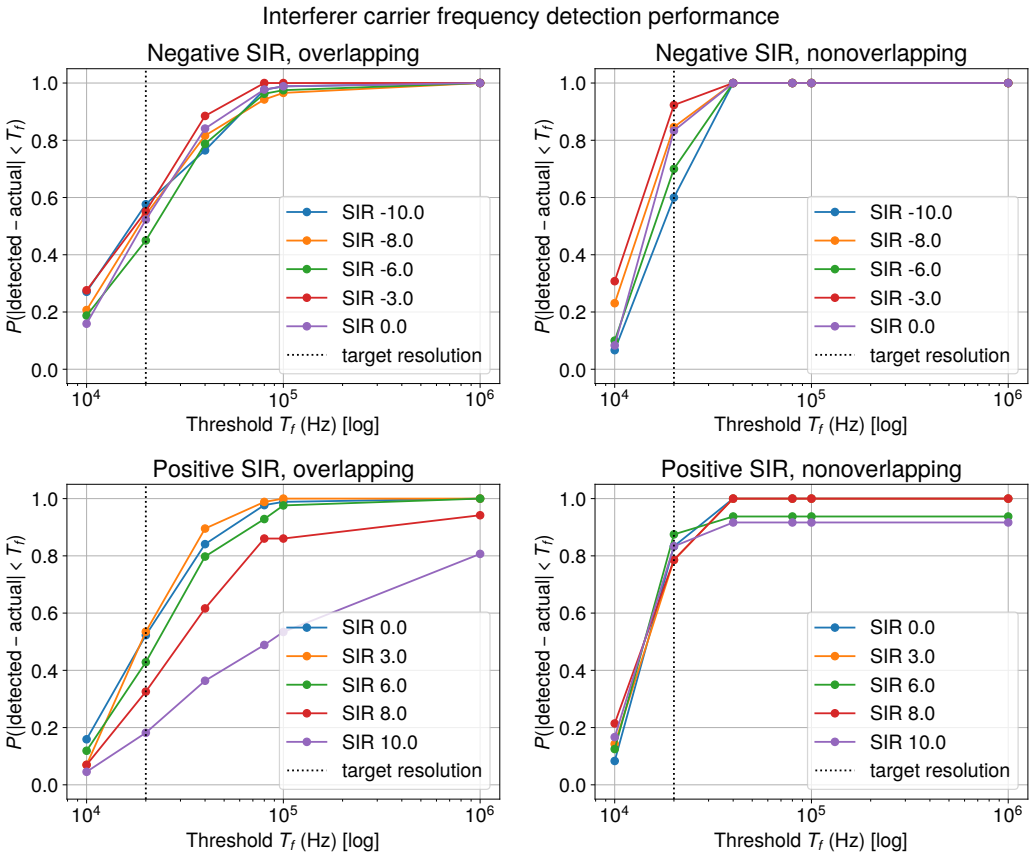

Figure 12. Similar to the outcome of Section 5.1, carrier frequency detection performance is degraded compared to symbol rate detection, and performance is affected by overlap and SIR as in Figure 11.

5.3 Performance of interference mitigation system

The experiment in Section 4.3 resulted in the average throughput and error performance results shown in Figure 13. These charts group the three link strategies together for each fading profile/interference combination, and the two interference scenarios for each fading profile are placed adjacent to show the effects of the random interferer. Each bar gives the mean value of the corresponding metric over the 30 trials, and the error bars give the standard deviation.

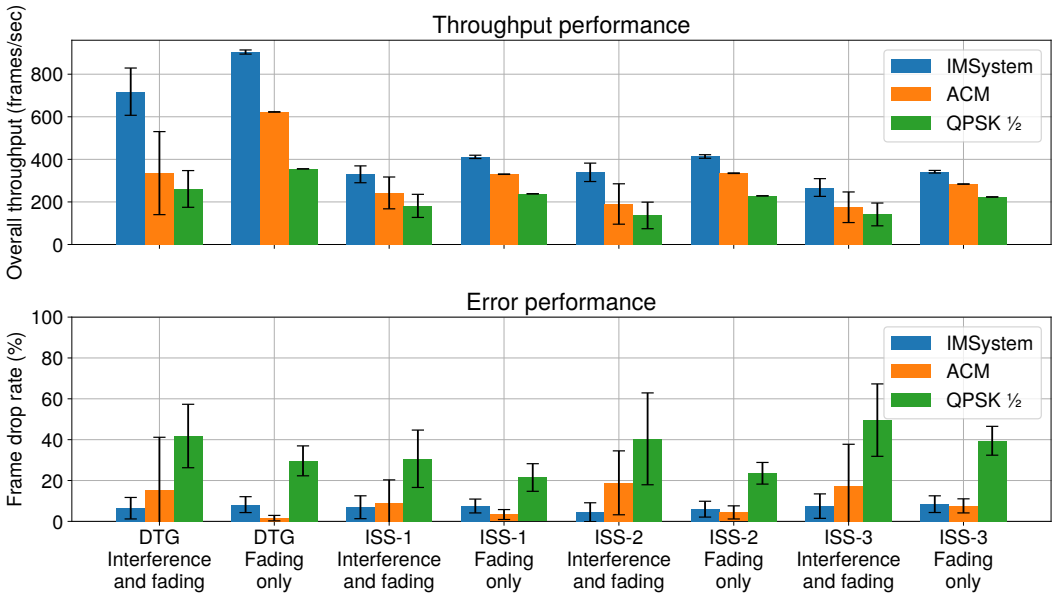

Figure 13. On average, the interference mitigation system had the highest throughput across all tested scenarios. The system also had the lowest error rates in the presence of interference. In scenarios without interference, the system had error rates higher than ACM and lower than QPSK ½, although the throughput in these cases was still higher even with the errors taken into account.

These results show that, on average, the interference mitigation system obtained the highest throughput in all the tested scenarios. The average throughput improvement over these scenarios is further summed up in Table 1. In cases with interference present, the system also obtained the lowest average frame drop rate. In contrast, with interference disabled, the system's resulting error rate ended up between ACM and QPSK ½. The throughput results, however, show that even with those frame drops on the link, the interference mitigation system was still able to successfully transfer more data from transmitter to receiver in scenarios without interference. As the objectives **w** were weighted to heavily favor throughput, this seems to be an acceptable result.

Another observation from the results in Figure 13 is the relatively large variances, as seen in the error bars. As this is more prominent in the scenarios with interference present, the cause seems to be the Markov randomness paired with a relatively small sample size. The Markov model leads to widely varying interference scenarios, where

Table 1. Average throughput improvement of interference mitigation system compared to other link management strategies.

Channel scenario	Improvement over ACM (dB)	Improvement over QPSK ½ (dB)
Interference and fading	2.46	3.50
Fading only	1.07	1.79

one trial may have almost no disruption by the interferer, and the next trial may lead to heavy degradation of the link. In addition, time constraints led to a relatively low amount of trials for this random model: each fading profile is ran in real time and models a 6 minute direct-to-Earth pass of the SCaN Testbed, so evaluating a large amount of these profiles is challenging.

6 Conclusions

This report detailed the design and evaluation of an automated interference mitigation system targeted for a satellite space-to-ground link. It was shown how satellite communication links can experience unexpected interference events due to shared spectrum allocations, radio configuration errors or scheduling conflicts, and it was proposed that a mitigation system could monitor the spectrum to detect these scenarios and respond to them in an automated way. The system implemented for this investigation integrates CAC signal processing for spectrum monitoring and applies a weighted-sum, multi-objective model to mitigate interference and continuously optimize the link. This system was evaluated to gauge its signal detection capabilities for various SNRs and SIRs, and it was tested alongside two other link management strategies on a suite of fading profiles and randomized interference scenario combinations. In this latter evaluation, it was shown to achieve, on average, the highest throughput over all test cases compared to the other strategies. These results are promising for this first version of the system, and there is plenty of future work that could be done to increase its capabilities and performance. The next section will cover these potential improvements.

7 Future Work

There are several areas of the interference mitigation system that could be improved with further revisions. One category of improvements would be to relax some of the limiting assumptions that currently affect the system, as identified in Section 2.4. Potential improvements include adding the ability to change from the current frequency to a new allocation, implementing support for detecting multiple interferers, improved handling of feedback command link failures, and evaluating system performance with dynamic interferers with Doppler and/or fading.

A second category of general enhancements will likely improve performance, based on the evaluations performed in this report. These improvements could include

providing a cooperation/overlap-avoidance feature, a Bit Error Rate (BER)/drop-avoidance feature, improving the signal detection routines, adding additional features to describe the interference (i.e. filter rolloff, relative power ratio), and finally, improving the overlap E_s/N_0 prediction to cover wider SNR ranges.

A final category of improvements could enhance the machine learning capabilities of the system. This includes replacing the weighted-sum algorithm with some other method for solving the multi-objective optimization problem. This may be some supervised or unsupervised learning algorithm. In any case, the performance can be compared with the current solution, including tradeoffs in execution speed and complexity. In addition, the system could be extended to include modulation classification [14] and pattern recognition in order to build a signal database and learn periodic schedules of recurring interferers. This would ideally allow the system to preemptively mitigate expected interference before the link is affected.

These improvements illustrate the potential that the current system has for improved performance and application to additional use cases.

References

[1] Joseph A. Downey, Michael A. Evans, and Nicholas S. Tollis. *DVB-S2 Experiment over NASA's Space Network*. Technical Report. NASA Glenn Research Center, Jan. 7, 2017.

[2] Joseph A. Downey et al. "Adaptive Coding and Modulation Experiment With NASA's Space Communication and Navigation Testbed". In: *34th International Communications Satellite Systems Conference*. Oct. 18, 2016.

[3] *Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications; Part 1: DVB-S2*. Standard ETSI EN 302 307-1 V1.4.1. European Telecommunications Standards Institute, Nov. 1, 2014.

[4] *CCSDS Space Link Protocols over ETSI DVB-S2 Standard*. Standard 131.3-B-1. Consultative Committee for Space Data Systems, 2013.

[5] Paulo Victor R. Ferreria et al. "Multi-Objective Reinforcement Learning-Based Deep Neural Networks for Cognitive Space Communications". In: *Cognitive Communications for Aerospace Applications Workshop*. June 27, 2017.

[6] Matteo Frigo and Steven G. Johnson. "The Design and Implementation of FFTW3". In: *Proceedings of the IEEE* 93.2 (2005). Special issue on "Program Generation, Optimization, and Platform Adaptation", pp. 216–231.

[7] E. Rebeiz et al. "Energy-Efficient Processor for Blind Signal Classification in Cognitive Radio Networks". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 61.2 (Feb. 2014), pp. 587–599. ISSN: 1549-8328. DOI: 10.1109/TCSI.2013.2278392.

[8] *Space Network User's Guide*. Standard 450-SNUG. NASA Goddard Space Flight Center, 2012.

[9] Il Yong Kim and Olivier de Weck. "Adaptive weighted sum method for multi-objective optimization: a new method for Pareto front generation". In: *Structural and Multidisciplinary Optimization* 31.2 (Feb. 1, 2006), pp. 105–116.

[10] Timothy M. Hackett et al. "Implementation of a Space Communications Cognitive Engine". In: *Cognitive Communications for Aerospace Applications Workshop*. June 27, 2017.

[11] *AOS Space Data Link Protocol*. Standard 732.0-B-3. Consultative Committee for Space Data Systems, 2015.

[12] Joseph A Downey et al. *Variable Coding and Modulation Experiment Using NASA's Space Communication and Navigation Testbed*. Technical Report. NASA Glenn Research Center, July 1, 2016.

[13] D. Li et al. "Mixed signal detection and carrier frequency estimation based on spectral coherent features". In: *2016 International Conference on Computing, Networking and Communications (ICNC)*. Feb. 2016, pp. 1–5.

[14] Aaron Smith, Michael Evans, and Joseph Downey. "Modulation Classification of Satellite Communication Signals Using Cumulants and Neural Networks". In: *Cognitive Communications for Aerospace Applications Workshop*. June 27, 2017.