

SRML: Space Radio Machine Learning

Paulo Victor Rodrigues Ferreira

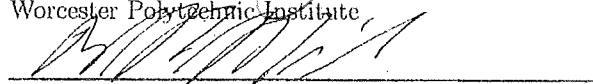


A Dissertation
Submitted to the Faculty
of the
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the
Degree of Doctor of Philosophy
in
Electrical and Computer Engineering

April 2017

APPROVED:


Alexander M. Wyglinski, Ph.D., Primary Advisor
Worcester Polytechnic Institute


William Michalson, Ph.D.
Worcester Polytechnic Institute


Randy Paffenroth, Ph.D.
Worcester Polytechnic Institute


Sven G. Bilén, Ph.D.
The Pennsylvania State University

SRML: Space Radio Machine Learning

Paulo Victor Rodrigues Ferreira



A Dissertation
Submitted to the Faculty
of the
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the
Degree of Doctor of Philosophy
in
Electrical and Computer Engineering
April 2017

APPROVED:

Alexander M. Wyglinski, Ph.D., Primary Advisor
Worcester Polytechnic Institute

William Michalson, Ph.D.
Worcester Polytechnic Institute

Randy Paffenroth, Ph.D.
Worcester Polytechnic Institute

Sven G. Bilén, Ph.D.
The Pennsylvania State University

Abstract

Space-based communications systems to be employed by future artificial satellites, or spacecraft during exploration missions, can potentially benefit from software-defined radio adaptation capabilities. Multiple communication requirements could potentially compete for radio resources, whose availability of which may vary during the spacecraft's operational life span. Electronic components are prone to failure, and new instructions will eventually be received through software updates. Consequently, these changes may require a whole new set of near-optimal combination of parameters to be derived on-the-fly without instantaneous human interaction or even without a human in-the-loop. Thus, achieving a sufficiently set of radio parameters can be challenging, especially when the communication channels change dynamically due to orbital dynamics as well as atmospheric and space weather-related impairments.

This dissertation presents an analysis and discussion regarding novel algorithms proposed in order to enable a cognition control layer for adaptive communication systems operating in space using an architecture that merges machine learning techniques employing wireless communication principles. The proposed cognitive engine proof-of-concept reasons over time through an efficient accumulated learning process. An implementation of the conceptual design is expected to be delivered to the SDR system located on the International Space Station as part of an experimental program.

To support the proposed cognitive engine algorithm development, more realistic satellite-based communications channels are proposed along with rain attenuation synthesizers for LEO orbits, channel state detection algorithms, and multipath coefficients function of the reflector's electrical characteristics. The achieved performance of the proposed solutions are compared with the state-of-the-art, and novel performance benchmarks are provided for future research to reference.

Acknowledgments

The research presented by this dissertation was made possible through the great advising of Professor Alexander M. Wyglinski. As my primary research and academic advisor, I am grateful and honored to be given the opportunity to work in the Wireless Innovation Laboratory. Thank you for all the teaching and for leading me towards my professional goals as an electrical research engineer. Thank you for making me a better person.

This research was supported by the Brazilian government through a Ph.D. fellowship, grant number BEX 18701/12-4, awarded by CAPES, as part of the Ciencias sem Fronteiras (Science without Borders) program, from August 2013 to July 2017.

The generous contributions provided by NASA's John H. Glenn Research Center, grant number NNX15AQ41H, partially supported the research presented in this dissertation. Special thanks to NASA GRC SCaN Testbed's principal investigator, Mr. Richard C. Reinhart, and his team, for all the guidance related to space communications.

I am thankful for collaborating with Professor Sven Bilén, of The Pennsylvania State University, and his master's student, Timothy Hackett (recipient of a NASA Space Technology Research Fellowship, grant number NNX15AQ41H). They are responsible for the hardware implementation of our algorithms.

I was fortunate to have Professor Randy Paffenroth in our research team. His guidance on the mathematical and machine learning background was key to the success of this research.

Special thanks for my new friends from the Newman Club, and from the lab, Travis, Bengi, Raquel, Le, Zoe, Renato, Guilherme, Kuldeep, and Skrikanth, for all technical and non-technical related discussions. We had lots of fun!

Thank you for my family, for being patient during these 4 years of absence, especially for my parents, for whom I owe my life. Thank you for all your love and teaching.

In God we trust, and believe that His guidance may allow humans to develop science, aiming to improve life quality and safety on Earth and beyond, with love.

Contents

List of Figures	vi
List of Tables	xii
List of Abbreviations	xiii
1 Introduction	1
1.1 Motivation	1
1.2 Current State-of-the-Art	2
1.2.1 Satellite Communications	2
1.2.2 Kalman Filter and Its Derivatives	4
1.2.3 Cognitive Radios	5
1.3 Technological Challenges	6
1.4 Research Contributions	7
1.5 Dissertation Outline	9
2 Satellite Communications: A Brief Tutorial	10
2.1 Satellite Orbits	11
2.2 Satellite Communications Systems	13
2.3 Link Budget Basics and Propagation Effects	14
2.4 Chapter Summary	18
3 Satcom Channel Modeling and Simulator Implementation	19
3.1 A More Realistic Multipath Rician Factor	20
3.1.1 Ionospheric Scintillation	22
3.1.2 Multipath Fading	22
3.1.3 Proposed Channel Model	23
3.1.4 Simulation Results	25
3.1.5 Conclusions	29
3.2 Step-by-Step Implementation of a 3D Rain Field Model for LEO Orbit Attenuation Time Series Synthesizers	30
3.2.1 Orbit Simulator, Ground Path Acquisition and Rain layer Intersection	33
3.3 Proposed Rain Fading Time Series Synthesizer for LEO Orbits Based on ITU-R P.1853-1	37

3.3.1	Overview of ITU-R P.1853	37
3.3.2	Proposed Rain Fading Time Series Synthesizer for LEO Orbits	38
3.3.3	Conclusions	42
3.4	LMS Channel Experiencing Rain Fading at Ka-Band from a GEO Satellite Downlink	43
3.4.1	LMSS Channel Simulation at Ka-Band	43
3.4.2	Conclusions	47
3.5	Chapter Summary	47
4	Kalman Filter and Interactive Multiple Model Filter for Satellite Communications	50
4.1	QoS Improvement in Satellite Communication Link Adaptations Using Kalman Filter	51
4.1.1	Channel Adaptation Concept	52
4.1.2	Prediction Using Kalman Filters	54
4.1.3	Decision Logic	57
4.1.4	Complete Simulator Algorithm	58
4.1.5	Simulator Setup	58
4.1.6	Simulation Results for GEO Orbit	58
4.1.7	Simulation Results for LEO Orbit	62
4.1.8	Conclusion	69
4.2	Shadowing Detection Using Interactive Multiple Model Filter	70
4.2.1	Defining Number of Filters in IMM	74
4.2.2	Automatic Search for Q	74
4.2.3	Simulation Results	76
4.2.4	Conclusions	82
4.3	Chapter Summary	83
5	Cognitive-Radio Engine Architecture for Autonomous Space Communications	85
5.1	Intelligent MAC Protocol for SDR-based Satellite Communications	86
5.1.1	Applications of CE for NASA	87
5.1.2	Project Overview	88
5.1.3	Scenario Description	89
5.2	Multi-Objective Reinforcement Learning (MORL) for Cognitive Radio-based Satellite Communications	90
5.2.1	Reinforcement Learning Overview	92
5.2.2	States, Goals and Rewards	95
5.2.3	Actions and How to Choose Them	100
5.2.4	Algorithm	100
5.2.5	Simulation Results	101
5.2.6	Conclusions	106
5.3	Hybrid Machine Learning Methods for Cognitive Space Communications	109
5.3.1	Machine Learning Overview	110
5.3.2	Neural Networks Overview	111

5.3.3	MORL and Deep Neural Networks: The RLNN Algorithm	112
5.3.4	RLNN Simulation Results	114
5.3.5	Remarks on RLNN Performance	116
5.3.6	Exploiting Reinforced Multi-dimensional Actions: The RLNN2 Algo- rithm	119
5.3.7	RLNN2 Simulation Results	128
5.3.8	Remarks on RLNN2 Performance	133
5.3.9	RLNN2 Performance Trade-off	135
5.3.10	Hardware Implementation Considerations	136
5.4	Chapter Summary	137
6	Conclusion and Future Work	142
6.1	Research Achievements	142
6.2	Future Work	143
A	Appendix: Kalman filter matrices	145
	Bibliography	147

List of Figures

2.1	Illustration of satellite orbit zones between the Earth and the Moon.	12
3.1	Illustration of a communication link between a mobile node and a GEO satellite. The received signal is a combination of the scintillated LOS and multiple reflections of scintillated rays [1].	21
3.2	Schematic of the mobile satellite communication system simulation testbed implemented in MATLAB. Two Rician channels model the effects of ionospheric scintillation and multipath fading.	25
3.3	BER performance of mobile communication link for a Rural Area (RA) 4-path channel model under ionospheric scintillation. Left panel shows results for horizontal polarization and right panel shows results for vertical polarization.	28
3.4	BER performance of mobile communication link for a Hilly Terrain (HT) 6-path channel model under ionospheric scintillation. Left panel shows results for horizontal polarization and right panel shows results for vertical polarization.	29
3.5	Absolute and conditional CCDF's for the percentage of time in one year that the rain rate exceeds the rain rate value at the abscissa for the Worcester area based on the ITU-R P.837-5. Curve fitting for conditional CCDF shows close agreement between the computed conditional CCDF and the theoretical model assumed by the ITU recommendation.	32
3.6	Simulated peak rain rate field area of $100 \text{ km} \times 100 \text{ km}$ with resolution of $0.5 \text{ km} \times 0.5 \text{ km}$ centered on Worcester, MA, USA. Rain rates provided by Rec. ITU-R P.837-5, the EXCELL rain cell model, and Goldhirsh rain field cell distribution model were combined to synthesize the resultant rain field.	34
3.7	Rain cell histogram of the peak rain rate field for an example of the simulation run shown in Fig. 3.6. The resultant peak rain rate contribution in $0.5 \text{ km} \times 0.5 \text{ km}$ resolution follows the log-normal distribution of the rain rate exceedance probability.	35
3.8	Slant path and elevation angle profile for one ISS pass over a GS located at WPI, Worcester, MA, simulated in STK. The pass is 512 seconds long. The minimum slant path distance is achieved during the maximum elevation angle.	36

3.9	Peak rain rate readings in mm/h for the LOS intersection with the synthesized rain layer during the satellite pass considered. For the same synthesized rain field, different passes having different elevation and azimuth angle profiles might result in different rain rate readings.	37
3.10	Segmentation of rain attenuation at time instant k into different peak rain rate $R_i(k)$ and segment LOS range $\Delta l_i(k)$. Knowledge of elevation angle and sub-satellite location allows computation of $\Delta l_i(k)$. Segments without rain contribute with only clear sky slant path attenuation.	38
3.11	Rain attenuation time series examples of outputs for two different fixed elevation angles using the available implementation of ITU-R P.1853 at Ka-band. For varying elevation angles, as is the case of LEO orbits, several time series would be required, but just a few points of each would be used.	39
3.12	Rough transitions between time series examples shown in Fig. 3.11 with fixed elevation angles at non specified time instants. LEO elevation angle shows how smooth these transitions should be.	40
3.13	Resultant LEO rain attenuation time series example showing computed Att_{LEO} values at each time instant k using proposed method “nonlinear weighted two dimensional interpolation of rain attenuation time series for LEO satellites based on elevation angle profile and ITU-R P.1853.” Elevation angle and rain attenuation time series for each elevation sector is shown for reference.	42
3.14	Rain attenuation time series for Ka-band based on ITU-R P.1853. Note that during the entire time interval the channel is facing rain fading.	44
3.15	LMS channel simulator diagram block based on ITU-R P.618-8. After the good or bad state time series generation, the same input parameters are provided to the state attenuation time series synthesizers, including good, bad and their transitions. MATLAB code implementation provided at Reference [2].	46
3.16	Example of an LMS channel synthetic attenuation time series at Ka-band based on ITU-R P.618-8. Shadowing due to mobile terminal LOS blockage by buildings and trees are represented by deep attenuation values, where the good and bad states are Loo distributed.	47
3.17	Block diagram of complete attenuation time series generator algorithm. Output time series is a combination of the individual rain fading time series and the LMS channel time series, generated individually.	48
3.18	LMS channel time series during rain fading for a constant speed mobile ground receiver communicating with a GEO satellite at Ka-band.	48
3.19	Addition of rain fading to the LMS amplitude. Note the change of the mean in the PDF. As expected, both events are independent.	49
4.1	Block diagram of a communication system architecture showing the enhancements added to the receiver, such as the predictor and decision logic. Channel impairments are accounted for by the AWGN channel block.	52
4.2	Theoretical specific rain attenuation for different rain rates at Ka-band and S-band for a carrier frequency at 26 GHz and 2 GHz, respectively, circularly polarized, based on the ITU-R P.838-3 [3]. For circular polarization the specific attenuation versus rain rate does not vary with the elevation angle.	53

4.3	Finite-state machine of the link adaptation and frequency band control between S-band and Ka-band. Triggering events are ruled by the E_b/N_0 conditions shown in Table 4.1.	54
4.4	Simulator diagram block with the operational flow graph between the transmitter and receiver. The channel is driven by the resultant synthesized attenuation weather profile and slant path loss. Receiver contains additional blocks such as Kalman filter and decision logic.	59
4.5	Complete flow graph of simulators used to analyze the Kalman filter prediction while driving the satellite communication adaptations. The rain attenuation profile can be either the 3D rain field generated or the proposed LEO rain synthesizer based on the ITU-R P.1853.	60
4.6	Rain attenuation predicted values k -steps ahead using noisy measurements. The true value cannot be seen due to the large number of samples	61
4.7	Zoomed-in section of Fig. 4.6. It can be seen that the predicted value is close to the true value to be measured k -steps in the future.	62
4.8	Received data rate changes according to the predicted E_b/N_0 based on the maximum allowed BER.	63
4.9	Slant path loss power from a signal being received at GS from a LEO satellite in the ISS orbit, during one of the passes over Worcester, MA. Also the elevation angle from an antenna at the GS shows the minimum attenuation during the maximum elevation angle.	64
4.10	Resultant rain attenuation computed using 3D rain field synthesizer. Marked values were computed using Eqs. (3.15) and (3.16). Due to the segmentation of path distances the output is not smooth. Smoothed version was achieved by applying moving average using LOESS robust regression (solid line). . .	65
4.11	Normalized power amplitude at receiver, for clear sky condition, offset by 10 dB (dashed line). Received amplitude profiles after subtraction of attenuation shown in Figures 4.10 and 3.13.	66
4.12	Comparison of modulation scheme transitions between a receiver with the KF predictions k -steps ahead ON and OFF, for three scenarios: 1) clear sky; 2) rain fading generated by ITU-based proposed method for LEO; 3) rain fading generated by 3D rain field.	69
4.13	Diagram block for the proposed design for search of Q matrix integrated with the IMM filter design. Each matrix combination is tested individually. $Q1$ and $Q2$ are the output matrices chosen by the learning blocks. The ultimate IMM output is the state decision if there is shadowing or not.	72
4.14	IMM outputs of state detection for mobile terminal using a Ka-band link with a GEO during rain fading through an suburban LMS channel. The measured amplitude values and the is shown for comparison. Shadowing detection is achieved at different levels of attenuation due to rain fading. Samples recorded at 10 Hz.	80

4.15	Shadowing detection performance in scenario with rain fading, and no shadowing, for a fixed terminal using a Ka-band link with a GEO satellite. Lower false detection mean time duration is better. Both threshold SNR and mean SNR methods show poor performance.	81
4.16	Shadowing detection performance on suburban (a) and rural (b) LMS channels during clear sky and rain conditions. Lower time duration means better performance. Shadowing detection on LMS channels by IMM filter is invariant to atmospheric conditions when sampled at 10 Hz, and better performance was achieved during rain by spending less time in erroneous shadowing detection condition when compared to current state-of-the-art detectors.	84
5.1	Examples of scenarios where the proposed Intelligent MAC protocol could to be tested. They consist of a links using the Near-Earth Network in S-band and links using the Space Network at both S- and Ka-bands.	90
5.2	High level block diagram of proposed cognitive engine. Inputs are environment measurements and communications system performance metrics. Outputs are actions composed by radio parameters. The learning algorithm reasons over the inputs in order to propose outputs that may or may not be accepted by the decision-making block.	91
5.3	Low level diagram block of proposed cognitive engine showing inter-relationships between the channel state detector, learning algorithm, decision logic, and their parameters.	92
5.4	Diagram block of reinforcement learning elements within the feedback loop for a down-link pair with the receiver at ground and transmitter in space. A_t , S_t , and R_t are the action, state, and reward at time t , respectively.	93
5.5	State change concept for communications system Markov process. Rewards are given for actions that sends the system to states that are above a certain threshold. No rewards are given otherwise. Even if the previous state was rewarded, if a different action results in an state above threshold it must be rewarded.	96
5.6	Time-series for chosen action and multi-objective communication performance for $\varepsilon = 0.5$. Knowledge of good performing actions is taken for granted due to the fixed exploration probability.	104
5.7	Time-series for chosen action and multi-objective communication performance for varying ε . The best known actions are exploited for a while, and eventually new ones are explored.	105
5.8	Normalized histograms for average time spent on a certain multi-objective normalized performance level for a GEO satellite-based link under clear sky conditions. For varying ε scenario, in all missions, more time was spent at higher performance levels when compared to the BF scenario.	108
5.9	The proposed RLNN block diagram, composed by the hybrid multi-objective reinforcement learning assisted by a deep neural network. “Virtual exploration”, driven by “action rejection” probability, prevents time expenditure exploring “bad actions” using radio resources over-the-air.	113

5.10 Deep neural networks used by RL exploration receive the same multi-dimensional input. Ensemble learning trains several NNs in parallel, and averages their outputs into a single multi-objective performance value. For AWGN channel, the mean E_s/N_0 was kept constant.	114
5.11 Example of a 50 seconds time series multi-objective RL using only “virtual exploration” NN with action rejection probability equal to 1 and performance threshold equal to 0.68. During the first 200 packets, the NN collects training data. Clearly, exploration of actions with predicted performance values below the threshold are avoided. Performance predictions had good accuracy with respect to their respective true values.	116
5.12 Boxplots of proposed RLNN algorithm multi-objective performances. On the left panels virtual exploration was turned off (likewise the algorithm proposed in [4] considering the DVB-S2 standard), and on the right panels it was turned on with rejection probability equal to 1. Top panels used fixed exploration probability value equal to 0.5, and bottom panels used the variable exploration probability function. High number of packets concentrated around larger fitness scores is better.	117
5.13 Boxplot of the genetic algorithm multi-objective performance during the same time duration used by the proposed RLNN simulations. The majority of the time is spent in very low performance levels.	119
5.14 Multi-objective reinforcement learning using deep neural networks ensembles block diagram. Virtual environment exploration prevents communications systems from spending time exploring combinations of radio parameters that would result in poor performance, while exploitation networks enables learning of actions best suitable for a dynamically changing environment.	120
5.15 RL work flow for choosing an action to be used in the real-world environment after training both exploration and exploitation NN. The exploration probability aides the choice between exploration and exploitation.	121
5.16 RL algorithm searching for better actions to be exploited over a GEO satellite communications channel during Mission 4. Before any transmission the RL exploitation action decision-maker considers performance values of actions when they were last explored. Exploited actions are represented by bands of constant amplitudes for a certain time duration.	122
5.17 Multi-objective performance for two different actions a_A and a_B , shown in panels (a) and (b), respectively. Each action was kept fixed during the entire time the environment changed, following the SNR profile, also shown in both panels. The performance of action a_A does not seem to change too much over time for each different mission. However, the performance of action a_B changes abruptly in all missions as the environment changes.	123
5.18 Ensemble of NN for action prediction by performance exploitation. Different sets of NN predict a specific action element by averaging their outputs. All NN receive the same multi-objective performance vector, in addition to the measured E_s/N_0	128

5.19 Actions are classified into ‘good’ or ‘bad’ based on their multi-objective performance values predicted by the ensemble of exploration NN using the performance threshold value. Next, a class is selected based on the ‘rejection probability’ value. The chosen action to be used on the environment is then randomly picked up from that class.	129
5.20 Examples of fitness score value time series per packet. Panels (a) to (f) show performances for Missions 1 to 6, respectively. In all missions, the majority of simulation packets experienced performances close to the ideal. Constant score levels represent exploitation, while spikes below and above them represent exploration.	139
5.21 Boxplots of normalized packet count distribution over fitness score values obtained using RLNN2. Panels (a) to (f) represent performances for Missions 1 to 6. More packets concentrated around larger score values is better.	140
5.22 Boxplots of normalized packet count distribution over fitness score error values between proposed RLNN2 and ideal solution. Panels (a) to (f) represent performances for Missions 1 to 6. More packets concentrated around lower error score values is better.	141

List of Tables

3.1	Simulation parameters to be combined into four different scenarios	26
3.2	Rician K-factors for ionospheric scintillation and terrestrial multipath fading filters	27
3.3	Elevation angle intervals	41
4.1	E_b/N_0 adaptation thresholds.	58
4.2	BER for different simulated scenarios	61
4.3	Communication system average performance for 2 different delays during clear sky; 100 runs	66
4.4	Communication system average performance for 2 different delays during rain fading; 100 runs	67
4.5	IMM filter MSE of rain faded channel – 100 runs.	78
5.1	Radio parameter adaptation inter-relationship	97
5.2	Communication mission phases and weights w_i	99
5.3	SDR parameter ranges	103
5.4	Multi-objective Communications Performance Distribution for Mission 1 . .	106
5.5	Multi-objective Communications Performance Distribution for Mission 2 . .	106
5.6	Multi-objective Communications Performance Distribution for Mission 3 . .	107
5.7	Multi-objective Communications Performance Distribution for Mission 4 . .	107
5.8	Multi-objective Communications Performance Distribution for Mission 5 . .	107
5.9	RLNN2 adaptable parameters	130
5.10	Weight values for communications missions	132
5.11	Integral values for average performance and error distribution curves	135

List of Abbreviations

ACM	Adaptive Coding and Modulation
ACTS	Advanced Communications Technology Satellite
AI	Artificial Intelligence
AIAA	American Institute of Aeronautics and Astronautics
APSK	Amplitude Phase-shift Keying
ATM	Asynchronous Transfer Mode
AWGN	Additive White Gaussian Noise
BCH	Bose-Chaudhuri-Hcquengham
BER	Bit Error Rate
BF	Brute Force
CBMM	Center for Brains, Minds and Machines
CCDF	Complementary Cumulative Distribution Function
CDF	Cumulative Distribution Function
CE	Cognitive Engine
COST	Cooperation in Science and Technology
CR	Cognitive Radio
DBPSK	Differential Binary Phase Shift Keying
DQN	Deep-Q Network
DTH	Direct-to-Home
DVB	Digital Video Broadcasting
DVB-S	Digital Video Broadcasting - Satellite
DVB-S2	Digital Video Broadcasting - Satellite - Second Generation

DVB-S2X	Digital Video Broadcasting - Satellite - Second Generation extension
EIRP	Effective Isotropic Radiated Power
ESA	European Space Agency
EXCELL	Exponential Spatial Rainfall Cell Distribution
FEC	Forward Error Correction
FPGA	Field Programmable Gate Array
GEO	Geostationary Earth Orbit
GLONASS	Global Navigation Satellite System
GNC	Guidance, Navigation and Control
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPU	Graphics Processing Unit
GS	Ground Station
HEO	High Earth Orbit
HT	Hilly Terrain
HVEC	High Efficiency Video Encoding
IEEE	Institute of Electrical and Electronics Engineers
IMM	Interactive Multiple Model
IRNSS	Indian Regional Navigation Satellite System
ISS	International Space Station
ITU	International Telecommunication Union
ITU-R	International Telecommunication Union Radiocommunication Sector
KF	Kalman Filter
LDPC	Low-Density Parity Check
LEO	Low-Earth Orbit
LMS	Land Mobile Satellite
LOS	Line of Sight
MAB	Multi-armed Bandit
MAC	Medium Access Protocols

MDP	Markov Decision Process
MEO	Medium Earth Orbit
ML	Machine Learning
MORL	Multi-objective Reinforcement Learning
MPEG-2	Moving Picture Experts Group - Version 2
MPEG-4	Moving Picture Experts Group - Version 4
MSE	Mean-Squared Error
NASA	National Aeronautics and Space Administration
NN	Neural Networks
NN1	Exploration Neural Network
NN2	Exploitation Neural Network
NTIA	National Telecommunications and Information Administration
OSI	Open Systems Interconnection
PDF	Probability Density Function
PSK	Phase-shift Keying
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
QPSK	Quadrature Phase-shift Keying
RA	Rural Area
RL	Reinforcement learning
RLNN	Neural Networks-based Reinforcement Learning
RLNN2	Neural Networks-based Reinforcement Learning - Version 2
RTT	Round Trip Time
Satcom	Satellite Communications
SDR	Software-Defined Radio
SRML	Space Radio Machine Learning
SNR	Signal-to-Noise
STK	Systems Tool Kit
STRS	Space Telecommunications Radio System

TD	Temporal-difference
TDRSS	Tracking and Data Relay Satellite System
UHF	Ultra High Frequency
VDBE	Value-difference Based Exploration
VSAT	Very Small Aperture Terminal
WINDS	Wideband InterNetworking engineering test and Demonstration Satellite
WPI	Worcester Polytechnic Institute
WRC	World Radiocommunication Conference

Chapter 1

Introduction

1.1 Motivation

In light of the transition in advanced wireless communications research from software-defined radio (SDR) systems, now mainly focused on technological development, to cognitive radio (CR), many new questions must now be answered. Also, recent investments in the development of powerful graphics processing units (GPUs) have renewed the interests of the artificial intelligence (AI) research community in improving their current algorithms and experimenting with more complex ones such as deep-Q networks (DQNs) and AlphaGo [5]. Meanwhile, the satellite communications industry is proposing business ventures with satellite fleets comprised of hundreds or even thousands of spacecraft. Such developments have triggered renewed interest with respect to increasing investments in satellite-based communications systems research, aimed at supporting not only the high throughput demands on the terrestrial cellular backbone, but also to enabling novel missions to explore other planets, such as Mars, or to mine asteroids [6]. These missions all have in common the fact that they require high link availability as well as robustness in order to operate in these extremely harsh environments. In addition, these future communication systems must operate autonomously, and be aware of the environment, their own hardware and software capabilities, and the mission they must complete with optimum performance. Given the convergence of future requirements and demands for the satellite communications industry,

the recent developments in AI, and the flexibility provided by SDRs in enabling CRs, a new research field has emerged: space radio-machine learning (SRML).

1.2 Current State-of-the-Art

This section describes the current state-of-the-art of satellite communications, Kalman filters, and cognitive radios.

1.2.1 Satellite Communications

Recently, several technology developments in satellite communications have been made in the fields of adaptive coding and modulation (ACM) schemes; Kalman filters (KFs) and their derivatives, high-throughput satellites; flexible payloads; cognitive and cooperative radio systems; and more. In this section, an overview is provided of the state-of-the-art in satellite communications technologies that relate to or will be potentially impacted by the research presented by this dissertation, especially for systems operating at Ka-band.

An important part of any realistic communications simulator is the communications channel. Previous studies have developed analytical mobile satellite channel models that combine ionospheric scintillation and terrestrial multipath [7–10]. However, these studies have omitted the contributions of the terrain material’s electrical properties on the Rician factor and do not provide a communications performance analysis. Seeking to fill this research gap, in this dissertation, a novel equation for the statistical multipath distribution is derived, which connects the Rician factor to the terrain’s reflection coefficient. Thus, the bit error rate (BER) performance is provided for an aircraft being affected simultaneously by ionospheric scintillation and multipath based on the ground composition it flies over while communicating with a geostationary orbit (GEO) satellite.

Regarding satellite channels in the presence of rain attenuation, there are two options to acquire an attenuation time series: (i) Perform measurement campaigns (or use the data from measurement campaigns), or (ii) Synthesize the channel effects using a channel time series simulator with the advantage of trying several different parameter combinations.

Measurement campaigns play an important role when supporting the development of

channel models and link performance analyses. Several research efforts have been done in order to collect real-world rain attenuation measurements. For instance, several measurement campaigns can be found in the open literature, such as the National Aeronautics and Space Administration's (NASA) Advanced Communications Technology Satellite (ACTS) [11, 12] and the European Space Agency's (ESA) Olympus [13] program, and the Japanese Wide-band InterNetworking engineering test and Demonstration Satellite(WINDS) program [14], which conducted experiments at Ka-band in order to assess the attenuation levels occurring in Digital Video Broadcasting (DVB) systems employed on GEO satellites. Furthermore, the Australian Fedsat [15] program published Ka-band experiment results for low-Earth orbit (LEO) satellite links, as well as a more recent French program using a GEO satellite at Ka-band [16]. However, these campaigns did not publicly release their measurements datasets, including the raw data required for research and performance analysis of adaptation techniques for specific channel conditions. Therefore, an alternate way to perform research experiments is by using a rain fading time series for Ka-band based upon International Telecommunication Union (ITU) recommendations. This dissertation makes use of several ITU Radiocommunication (ITU-R) sector recommendations on synthesizing attenuation time series under different environmental conditions.

ACM is used in cases when the received signal power at the receiver changes due to impairments in the channel such as fading resulting from the weather or the relative motion between the transmitter and the receiver [17]. ACM has been applied to GEO satellite channels operating in the S-band [18] as well as the Ka-band [19], and in this work it is considered as the standard solution to be implemented in LEO satellite links. ACM performance in LEO satellite channels during fading, using an end-to-end system simulation have not been published in the open literature, except for theoretical approaches such as reference [20].

DVB-S2 [21] is the second generation of the original DVB-S, the DVB technical standard for GEO satellite-based digital television broadcast system focused on direct-to-home (DTH) services. It is mainly comprised of advances in the physical layer, using more efficient channel coding, modulation, and error correction techniques, combined with the most recent video compression technology for transmissions compatible with the Moving Picture

Experts Group version 2 (MPEG-2) and MPEG-4 standards. It makes use of a powerful forward error correction (FEC) scheme based on low-density parity check (LDPC) codes concatenated with Bose–Chaudhuri–Hcquengham (BCH) coding, and a mix of four modulation constellations combined with a wide range of code rates controlled by an ACM scheme aimed at mitigation of propagation impairments, allowing adaptation on a frame-by-frame basis. DVB-S2 and the most recent extension, S2X [22], increase the granularity of the modulation and encoding schemes improving performance for mobile applications on land, sea and in the air, while allowing channel bonding, which combines unused portions of the spectrum by up to three transponders into one virtual channel that would have required a higher bandwidth, not offered by the individual transponders. It is expected that with this S2X update, ultra-high definition television will be made more efficient when combined with the new high efficiency video encoding (HVEC) video-coding scheme.

1.2.2 Kalman Filter and Its Derivatives

The Kalman filter (KF) was conceived by Rudolph A. Kalman in 1960 as a recursive solution to the discrete-data linear-filtering problem [23–25], *i.e.*, a filter that is capable of estimating variables (states) from noisy measurements and other inaccuracies of a known approximation of the real system model. The most common applications of the Kalman filter are in the fields of position tracking for guidance, navigation, and control (GNC) of aircraft and spacecraft [26–29], as well as orbit predictions [30], for which the speed and acceleration equations well describe the behavior of the objected being monitored. In addition to GNC, KFs have been widely applied to communications systems for automatic frequency control [31] and global navigation satellite system (GNSS) carrier tracking [32–34]. Since the KF has not been applied to support ACM yet, this research addresses this gap and provides some analysis and simulations of its application when used to assist ACM algorithms in a satellite communications channel.

Furthermore, KFs are used within Interactive Multiple Model (IMM) filters [35, 36], which implement a refined version of the KF that uses a bank of KFs instead, where each KF represents a model of one possible behavior of the variables being tracked. IMM filters have been applied to aircraft tracking [26], in GNSS navigation [37], and for secure

communications systems [38]. Design and performance analysis of IMM filters in satellite communications applications have not been made available in the literature yet. Thus, this dissertation addresses this research gap by providing an algorithm that deals with the lack of knowledge of a system model while showing the application of an IMM filter as a shadowing detector for Land-Mobile Satellite (LMS) channels in the presence of rain at Ka-band.

1.2.3 Cognitive Radios

The next technological leap for SDR is expected to be the implementation of on-board cognition, which can potentially provide the SDR platform with environmental awareness across several Open Systems Interconnection (OSI) layers [39]; real-time knowledge of the current channel conditions; status of its communicating nodes and the network; and assessment of its own available resources. Such information is very important for optimizing the communications link performance. In the past, several optimization algorithms, such as genetic algorithms [40], have been applied to cognitive radios [41]. These optimization techniques do not always converge and might take several hundred or even thousands of iterations, *i.e.*, algorithm runs, until a stable solution is found. Changes in the environmental conditions would require another series of iterations in order to search for a new solution, increasing latency. Also, if these same scenarios repeat, the system is unable to use previous solutions given that it cannot learn. Learning is considered to be the cornerstone of AI. Therefore, CR systems must implement these principles in a computationally feasible way in order to leverage true CR capabilities.

Machine learning (ML) techniques have been studied with respect to their approach to CR, thus addressing the learning issue [42–45]. Both approaches, optimization and ML, have been studied in order to determine how they can assist CR systems with respect to finding the best configuration parameter set, with the majority of research focused on spectrum management, including sensing techniques, for terrestrial links [45–47]. The scenario is the same for satellite links, with the majority of CR research focused on spectrum resource allocation [48–50], and to the best of the author’s knowledge, almost none on radio resource management for point-to-point communication link.

Regarding adaptive medium access protocols (MACs), most research has been done for

asynchronous transfer mode (ATM) over satellite links [51, 52], adaptive FEC, ACM for ATM over satellite links [53], and hybrid models that adapt over time to different aspects depending on the user needs and the channel conditions [48, 49]. Theoretical approaches to adaptive schemes for satellite communications over fading channels using ACM and power adaptation have been pursued [54].

In this dissertation, the scope of the communications performance is beyond individually offering minimum BER, maximum throughput, or power adaptation. During critical space mission phases, communications systems might require proper management of resource allocation while facing conflicting requests to achieve performance levels on specific metrics, limited resource availability, and minimizing impact on other sub-systems sharing the same resources. To the best of the author's knowledge, there has not been any research reported on an adaptive MAC protocols for resource allocation of radio parameters for SDRs in orbit that consider multiple conflicting performance metrics.

In order to enable cognitive radios in space-based communication systems, this dissertation provides a conceptual cognitive engine as the core technology of an intelligent MAC protocol, which leverages an AI system to efficiently manage multiple resources and to autonomously control communications through dynamic channels while learning on-line how to achieve the requested goals based on the hardware and software resources available and current channel conditions.

1.3 Technological Challenges

To the best of the author's knowledge, the research group formed by this dissertation's author and its collaborators seem to be the first to address the following technical challenges in the area of space radio-machine learning for enabling cognitive radios for space communications on dynamically changing channels:

- Channel modeling of the influence of the physical properties of scatterers and reflectors on multipath channel simulators for airborne terminals;
- Synthesis of attenuation time series for different environment conditions at Ka-band for fixed and mobile terminals downlink from LEO and GEO satellites;

- Communications channel state detection for fixed and mobile terminals experiencing rain at Ka-band; and
- Cognitive-radio engine architecture for autonomous operation in space during different mission phases.

1.4 Research Contributions

Given the state-of-the-art and the technological challenges presented above, this dissertation provides the following research contributions:

- Derivation of Rician factor function of the reflection coefficient of terrain for an air-borne terminal communicating with a GEO satellite;
- Algorithm for rain attenuation time series for LEO orbits using ITU-R P.1853 recommendation;
- Prediction algorithms using Kalman filter for QoS improvement of ACM schemes during rain at Ka-band downlink from LEO satellite;
- Model-free prediction algorithm using an IMM filter for shadowing detection during rain for mobile terminals experiencing rain at Ka-band downlink from GEO satellite; and
- Cognitive engine architecture and ML algorithms that enable autonomous on-line learning on stochastic environments during different multi-objective mission phases.

List of Journal Papers

- P. V. R. Ferreira and A. M. Wyglinski, “QoS improvement for LEO-based SATCOM at Ka-band,” *IEEE Transactions on Aerospace and Electronic Systems*, Submitted, October 2016.
- P. V. R. Ferreira, R. Paffenroth, and A. M. Wyglinski, “Interactive Multiple Model Filter for land-mobile satellite communications at Ka-band,” *IEEE Access*, In print preparation, January 2017.

- P. V. R. Ferreira, R. Paffenroth, A. M. Wyglinski, T. M. Hackett, S. G. Bilén, R. Reinhart, and D. Mortensen “Multi-objective reinforcement learning for cognitive satellite communications using deep neural networks ensembles,” In preparation for *IEEE Journal on Selected Areas in Communications*, April 2017.

List of Conference Papers

- P. V. R. Ferreira, R. Metha, and A. M. Wyglinski, “Cognitive radio-based geostationary satellite communications for Ka-band transmissions,” in *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, December 2014.
- P. V. R. Ferreira and A. M. Wyglinski, “Performance analysis of UHF mobile satellite communication system experiencing ionospheric scintillation and terrestrial multipath fading,” in *2015 IEEE 82nd Vehicular Technology Conference (VTC Fall)*, September 2015.
- P. V. R. Ferreira, R. Paffenroth, A. M. Wyglinski, T. M. Hackett, S. G. Bilén, R. Reinhart, and D. Mortensen “Multi-objective reinforcement learning for cognitive radio-based satellite communications,” in *34th AIAA International Communications Satellite Systems Conference*, October 2016.
- T. M. Hackett, S. G. Bilén, P. V. R. Ferreira, A. M. Wyglinski, and R. Reinhart, “Implementation of a parameterized interacting multiple model filter on an FPGA for satellite communications,” in *34th AIAA International Communications Satellite Systems Conference*, October 2016.
- P. V. R. Ferreira, R. Paffenroth, A. M. Wyglinski, T. M. Hackett, S. G. Bilén, R. Reinhart, and D. Mortensen “Multi-objective reinforcement learningbased deep neural networks for cognitive space communications,” *IEEE Cognitive Communications for Aerospace Applications (CCAA) Workshop*, Cleveland, Ohio, June 2017.
- T. M. Hackett, S. G. Bilén, P. V. R. Ferreira, A. M. Wyglinski, and R. Reinhart, “Implementation of a space communications cognitive engine,” *IEEE Cognitive Communications for Aerospace Applications (CCAA) Workshop*, Cleveland, Ohio, June 2017.

1.5 Dissertation Outline

A brief introduction to key satellite communications topics is provided in Chapter 2, including satellite orbits, satellite communication systems architecture, link budget items, and propagation effects. In Chapter 3, a more realistic multipath channel model is described by the derivation of the Rician factor considering the physical properties of the terrain. It also addresses the implementation challenges of rain attenuation time series. Attention is given to the details of attenuation time series for LEO orbits and an algorithm is provided to derive these from the ITU-R P.1853 time series generator. In Chapter 4, novel algorithms for channel state prediction are described, leveraging Kalman filters seeking to improve the QoS of mobile satellite receivers operating at Ka-band when experiencing rain attenuation. In addition, a new model-free Interactive Multiple Model filter algorithm is proposed to detect shadowing in LMS channels during rain attenuation. Lastly, Chapter 5 provides a novel cognitive radio architecture specially designed for space communications, including new AI-based algorithms that allow on-line learning for cognitive autonomous operations including reasoning, decision making, and channel awareness for the node.

Chapter 2

Satellite Communications: A Brief Tutorial

Satellite communications was first proposed by Arthur C. Clarke in 1945 [55, 56]. Sputnik 1, the very first satellite was launched in 1957 into a LEO orbit by the U.S.S.R. (now Russia). In 1962, under a multi-national agreement between several U.S. government agencies and industries and several European industries, the Telstar-1 satellite was launched into a medium Earth orbit (MEO) orbit to experiment with satellite communications over the Atlantic Ocean. Three years later in 1965, Intelsat 1 (known as Early Bird), the first GEO satellite, was launched. Since then, many more GEO satellites have been launched once the technology was proven to be feasible [57]. Nowadays, much more complex satellite systems co-exist in different orbits, including multiple Global Navigation Satellite System (GNSS) systems for guidance and navigation, for instance the Global Positioning System (GPS) [58], Galileo [59], Global Navigation Satellite System (GLONASS) [60], BeiDou [61], and the Indian Regional Navigation Satellite System (IRNSS) [62], all in MEO orbits; Earth observation satellites in LEO orbits; and hundreds of GEO satellites for telecommunications, including TV and radio broadcast, broadband internet service access, weather monitoring, and others [63]. Military satellites are known to operate in these orbits but are not covered in this dissertation.

Communications via satellites has the ultimate goal to provide information access on

a global scale given its wide coverage area, especially to areas not covered by terrestrial network infrastructure in remote and low density population areas such as rural areas, forests, deserts, and polar regions. For mobile operations in the air and at sea, satellite offers the best communications option, especially if the communicating nodes are not in line-of-sight (LOS) with each other or with a repeater in the vicinity that could relay the signal.

2.1 Satellite Orbits

Satellite orbits obey Kepler's Laws of Planetary Motion [64], with orbits having an elliptical trajectory with the Earth at one focus. The orbits are divided into three main zones based on the altitude range: (i) LEO ranges from 200 km to 2,000 km, (ii) MEO ranges from 2,000 km to 36,000, (iii) high Earth orbit (HEO) comprises any orbit higher than 36,000 km such as the Moon, which orbits the Earth at approximately 384,000 km. While orbiting around the Earth, the satellites are in an inclination plane (also known as orbital plane), an imaginary plane that has an inclination with respect to an imaginary plane intersecting the Earth at the equatorial line. Figure 2.1 illustrates these orbit zones and the most common satellite orbits.

Satellites in LEO, such as the International Space Station (ISS), the Hubble Space Telescope, Earth observation satellites such as Jason-3 [65], and smallsat's (small satellites) for science experiments or remote sensing missions such as Terra Bella [66] and Planet [67], all have an orbital period between 80 and 120 minutes.

Satellites in MEO, mainly GNSS constellations for guidance and navigation such as GPS, have an orbital period between 2 and 20 hours. In 2013, O3b Networks launched telecommunications satellites into MEO aimed at commercializing broadband internet access. Its satellite fleet is located at an 8,000-km-altitude orbit with an orbital period of 6 hours.

The boundary between MEO and HEO is where satellites in GEO orbit are located. At 36,000 km altitude, these satellites appear to be “fixed” in the sky for any communicating node on the Earth. Satellites in GEO orbit have an orbital period of approximately 24 hours and are located in the equatorial plane. From this location, beams from GEO satellites have

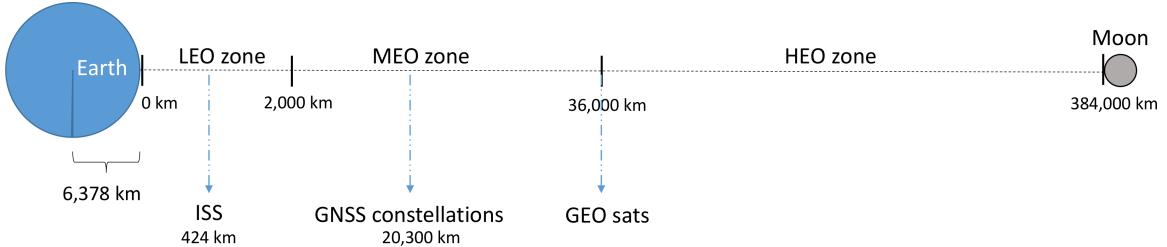


Figure 2.1: Illustration of satellite orbit zones between the Earth and the Moon.

the potential to cover 1/3 of the Earth’s surface, making them ideal for broadcast services and continuous coverage of very wide areas of the globe. Depending on the type of antenna used, it is possible to focus the signal beam on certain regions within the portion of the Earth in view. These individually focused antennas are known as spot beams. A more complex antenna system, known as phased-array antenna, is capable of providing multiple spots using the same antenna. It is also possible to combine different beams focused on the same region by using different polarizations. The beam coverage area is known as the satellite footprint. Concentrating beams over a certain region has the primary advantage of increasing the power at the receiver, allowing for frequency reuse.

Ground stations located at the **equatorial region** have the highest ground station antenna elevation angle when pointing to a GEO satellite, which decreases with increasing latitude, which sometimes make it infeasible in regions close to the poles. In order to cover these other regions there are other types of orbits, for instance the polar orbit is very similar to a LEO orbit with an inclination close to 90 degrees. Another interesting orbit type is known as sun-synchronous, which is also a LEO orbit. A sun-synchronous satellite combines its altitude and inclination such that it passes over a certain region always at the same local time. Derivations of this orbit include noon/midnight when the satellite passes over a specific region at noon/midnight local time, and dawn/dusk which allows the solar panels to always be in view of the sun while allowing on-board sensors to be hidden from the sun or pointing to the night side of the Earth. The last most interesting orbit type is known as Molniya, in which the satellite orbit is highly elliptical, *i.e.*, having high eccentricity, with an orbital period of half a day such that the satellite spends most of the time close to apogee with altitude as high as 40,000 km, and consequently spends much less time close

to perigee. This is an alternative to polar orbits for regions at high latitudes.

2.2 Satellite Communications Systems

In order to communicate with a satellite, an end-to-end infrastructure is required, comprised of the ground segment, which comprises all the equipment for the ground station, also referred to as gateway or terminal, and the space segment, which comprises the spacecraft, in this case the satellite or fleet of satellites.

A **ground station** is the terrestrial node that communicates directly with the satellite in space and has all the necessary equipment to support this operation, such as antennas and power amplifiers, connected to the satellite operator, which controls the satellite by managing its operations and monitoring its telemetry data. There are two different types of ground stations: gateways and end-user ground stations. Gateways are usually a complex with several large antennas that act as hubs to transmit and/or receive large amounts of data to/from satellites. Cable TV companies usually have a ground station with antennas connected to several different satellites from which it acquires different channels and combine them into its proprietary cable system for terrestrial distribution. On the opposite side, satellite TV operators use gateways to send all the TV content up to the satellite for broadcast distribution. Satellite operators usually control several satellites, doing maneuvers and receiving health telemetry data, and may do so through their own ground stations. End-user ground stations can receive-only stations, such as the satellite TV receivers and GPS receivers, or the very small aperture terminal (VSAT) that is a small ground station that can transmit and receive to/from a satellite.

The satellite itself is the unit that is launched into space and positioned in a certain orbit, located in a specified orbital plane. The satellite has three main parts: the **bus, solar panels, and the payloads.** The bus contains all the equipment for house-keeping and control of the unit; is responsible for storing fuel to be used by thrusters during its operational life for orbital positioning corrections and maneuvers; gimbals and wheels for orientation control; thermal control systems; radiation shields; power batteries control; and *etc.* The solar panels are located outside the satellite, and depending on the mission, if they are too

big to be installed on the outer surfaces, they are folded prior to launch and deployed in space. The payload is comprised of the main electronics hardware and software to execute and support the satellite's missions. The basic payload includes the avionics payload that controls the satellite itself and communication hardware for telemetry, command and control of the satellite operations, as well as more specific payloads, for instance, communications hardware for telecommunication missions and/or sensors for Earth observation missions.

Based on the mission, the communications payload divides the satellite into two categories: **bent-pipe satellite**, in which the information simply passes through the satellite acting as a link relay, and a **regenerative satellite**, in which there is a complex and more robust on-board processing payload capable of signal demodulation, data compression, multiplexing, beam forming, and hopping.

Given that there are multiple orbit types with hundreds of satellites co-existing over a certain region, interference is controlled by dividing the **RF spectrum into bands**. The ITU, through the World Radiocommunication Conference (WRC) [68], manages the frequency bands among the satellite operators for different usages. In the United States, the frequency allocation is regulated by the National Telecommunications and Information Administration (NTIA) [69]. The most used bands are L-band from 1.2 GHz to 1.6 GHz, S-band from 2 GHz to 4 GHz, C-band from 4 GHz to 6 GHz, Ku-band from 11 GHz to 14 GHz, and, most recently, Ka-band from 20 GHz to 30 GHz. Due to regulatory issues over the years lower bands received a narrower bandwidths and today are crowded, which resulted in higher frequency bands being allocated with wider bandwidths in order to support the increasing number of users and demand for broadband services. For instance, bands at L-band are usually 15 MHz wide, whereas at Ka-band they can be as wide as 2.2 GHz.

2.3 Link Budget Basics and Propagation Effects

The power irradiated by an isotropic antenna that is assumed to irradiate the same power in all directions from a punctual source expands as a sphere with an increasing radius as the wavefront travels away from the source. As a consequence, the power received at a certain distance from the source **decreases** due to the **power density reduction** in an ideal medium

without energy absorption or reflection. This effect is described by the **Friis equation** that computes the received power as:

$$P_r = \frac{P_t G_t G_r \lambda^2}{(4\pi d)^2}, \quad (2.1)$$

where P_r is the received power in watts, P_t is the transmitted power in watts, d is the distance between the transmitter and the receiver in meters, λ is the wavelength, G_r and G_t are the the receiver and transmitter antenna linear gains, respectively, given by:

$$G = \eta_A \frac{4\pi A}{\lambda^2}, \quad (2.2)$$

where η_A is the antenna aperture efficiency and A is the physical antenna aperture area computed by $A = \frac{\pi D^2}{4}$ where D is the antenna diameter.

In Eq. (2.1), the term $P_t G_t$ is known as the effective isotropic radiated power (EIRP), which is the transmitted signal power concentrated in a certain direction by a practical antenna [57]. The term $\frac{\lambda^2}{(4\pi d)^2}$ is known as path loss. Depending on the environmental conditions through which the signal propagates towards the receiver, a more realistic version of Eq. (2.1) would also include losses in both transmitter and receiver antennas, from the atmosphere, and miscellaneous losses through the transmitter and receiver chain such as cables and connectors. The most important of these additional losses is the atmospheric loss, which accounts for losses due to rain, clouds, fog, snow, ice, gases, sand, as well as tropospheric scintillation and ionospheric scintillation [70, 71]. These atmospheric losses are functions of several parameters such as the terminal antenna elevation angle, signal carrier frequency, and polarization. Other losses could be associated with the presence of objects partially blocking the signal LOS, for instance trees with different foliage density; buildings and other objects with different structural materials composition; or even dust, water, or snow accumulated on the antenna. Depending on the combination of these factors, the following effects can be experienced by the signal:

- **Absorption:** irreversible amplitude reduction of the electric field intensity;
- **Scattering:** signal energy dispersed in a given direction;
- **Reflection:** change in the signal propagation direction due to variations in the index of refraction the medium;

- **Diffraction:** change in the signal propagation direction due to an obstacle (reception behind a building or mountain) based on the Huygens–Fresnel principle; and
- **Multi-path:** destructive or constructive summation of reflected and/or diffracted components that arrive at the receiver.

For carrier frequencies below 30 MHz, the signals get absorbed or reflected by the Ionosphere layers D, E, and F, located at altitudes between 50 km and 1000 km. Thus, it is almost impossible to communicate with a satellite at these frequencies. However, this reflection effect is useful for terrestrial transmissions around the world beyond LOS. Signals with carrier frequencies between **30 MHz and 3 GHz pass through the Ionosphere and are more susceptible to the ionospheric scintillation** (disturbances triggered by space weather phenomena), Faraday polarization rotation, group delay, multi-path fading, tropospheric fading, and refraction. Above 3 GHz signals are mainly affected by attenuation due to atmospheric gases, hydrometeor attenuation, clouds and fog attenuation (only for frequencies higher than 10 GHz), and depolarization by multi-path and hydrometeors (only for frequencies higher than 12 GHz).

Attenuation **due to rain is a function of the carrier frequency**, terminal antenna elevation angle, and amount of rain [3]. At 26 GHz, the specific attenuation varies from 1 dB/km for a low intensity rain (5 mm/hour) to as high as more than 10 dB/km for heavy rain (150 mm/hour).

Water vapor and oxygen are responsible for high signal attenuation levels for frequencies of **10 GHz and above**. For frequencies close to 23 GHz, rain can cause attenuation levels as high as 0.1 dB/km. Oxygen makes communications in the atmosphere impracticable for frequencies close to 60 GHz, at which attenuation levels reach more than 10 dB/km. Figure 5 in Reference [72] shows the specific attenuation due to atmospheric gases, water, and oxygen in dB/km for frequencies up to 350 GHz.

Signal fading causes the **signal intensity to vary due to path-loss changes** over time. For a mobile terminal, the communications channel coherence time T_c is given by:

$$T_c = \sqrt{\frac{9}{16\pi f_d^2}}, \quad (2.3)$$

where f_d is the Doppler frequency. If T_c is much bigger than the bit duration T_b , then the fading is known as slow fading. Otherwise, it is known as fast fading, when rapid signal fluctuations occur resulting in quick variations in the signal phase, angle of arrival, and polarization. Scintillation, a fast fading type of signal attenuation, potentially decreases the performance of a mobile receiver. However, atmospheric induced scintillation can also affect the receiver performance of a fixed terminal. For instance, it plays a key role in defining the GPS signal reception quality since, when in presence of scintillation, the systems loses track of the phase, which results in computing erroneous locations [73]. Ionospheric scintillation, covered in more detail in Section 3.4, is caused by space weather-related phenomena, while tropospheric scintillation is caused by rain at low antenna elevation angles and air turbulence during clear sky conditions [3].

On top of the power losses, noise plays a critical role in any wireless communications system, and satellite communications is no exception. Any body with a physical thermal temperature generates an electrical noise power P_n in watts, over a certain bandwidth B_n in Hz, given by:

$$P_n = bT_sB_n, \quad (2.4)$$

where b is the Boltzmann's constant (1.3910^{-23}J/K), T_s is the system noise temperature, a sum of antenna noise temperature and receiver noise temperature in kelvin, and B_n is the bandwidth in Hz. Noise can be generated by natural sources such as the Sun, atmospheric events (rain, clouds, lightning, gases), distant galaxies (frequencies below 2 GHz) and cosmic background (frequencies above 2 GHz), and also by artificial sources (man-made) such as an electrical equipment operating in the proximity of the receiver antenna, and other communication signals such as an interfering signal. An absorbing medium, such as the atmospheric phenomena, generates noise that is usually accounted for in link budgets as being coupled to the receiver's antenna noise temperature.

Hence, the link performance is measured by the C/N ratio in dB, where C is the received power P_r and N is the noise power P_n , or by the C/N_0 ratio in dB-Hz, where N_0 is the noise power spectral density.

2.4 Chapter Summary

In this chapter, a brief overview of the main satellite communications elements was provided. The three most used satellite orbits, LEO, MEO, and GEO, were described in terms of altitudes and types of satellites and services found in each. An introduction to the basics of satellite systems and their main components (solar panels, bus, and payloads) was given. The principles of link budget, including equations and propagation effects, were covered as well.

Chapter 3

Satcom Channel Modeling and Simulator Implementation

In order to provide realistic satellite communications (Satcom) performance analysis, field measurement campaigns should be done. However, field measurements not only require robust equipment, they are time consuming and require the experimenter to be in place where the equipment is installed and wait until all desired environmental conditions are met. An alternative is to use data previously collected by other measurement campaigns, when these are publicly available, or to generate synthetic time series using simulators. Compared to the state-of-the-art described in Section 1.2.1, the main contributions of this dissertation for satellite communications channel modeling are the following:

- A novel equation derivation for modeling the statistical multipath distribution, making the Rician factor function of the terrain's reflection coefficient;¹
- Verified new Rician factor model through analysis of the combined effects of ionospheric scintillation and terrestrial multipath over different terrain on the receiver BER performance;¹
- Step-by-step implementation description of the 3D rain attenuation field synthesizer described in Reference [74] for LEO orbits. Emphasis is given to the multi-composite

¹Published in the 82nd 2015 IEEE Vehicular Technology Conference [1]

rain rate conversion to the resultant instantaneous path attenuation;^{2,3}

- A novel rain attenuation time series synthesizer implementation for LEO orbits;^{4,5} and
- Algorithm implementation description of the LMS channel time series synthesizer with variable sample rate, based on updated ITU recommendations from 2013 and 2015.^{5,6}

All these aim to make time series generators more realistic such that the experimenter can replicate a certain channel effect under different environmental conditions with the convenience of changing some parameters while keeping others constant.

3.1 A More Realistic Multipath Rician Factor

Based on several public accounts of the operation described in References [79,80], it was observed that communications were not reliable during this time period and that a number of recent scientific studies [81] have identified space weather as one of the main contributors for the communication link disruption during the military operation. These studies also mentioned that the space weather effects were enhanced by the multipath fading caused by the terrain, but did not quantify their combined effect.

In order to disrupt a satellite communication link, the event causing the fading does not need to persist for a long time duration. Rather, it just needs to occur across a certain period of time with a high intensity, which is the typical behavior of ionospheric scintillation. Given that most robust satellite communication systems use some sort of network protocol, if the fading event occurs during an exchange of synchronization messages between a transmitter and a receiver, the link can no longer be sustained and a transmission disruption occurs.

When such a fading event occurs for a receiver moving over terrain such as mountains, flat ground, or over the sea, the multipath contribution of the scintillated signals may

²Code available on GitHub [75]

³Submitted to IEEE Transactions on Aerospace and Electronic Systems [76]

⁴Code available on GitHub [2]

⁵Available on GitHub [77]

⁶In-print IEEE Access [78]

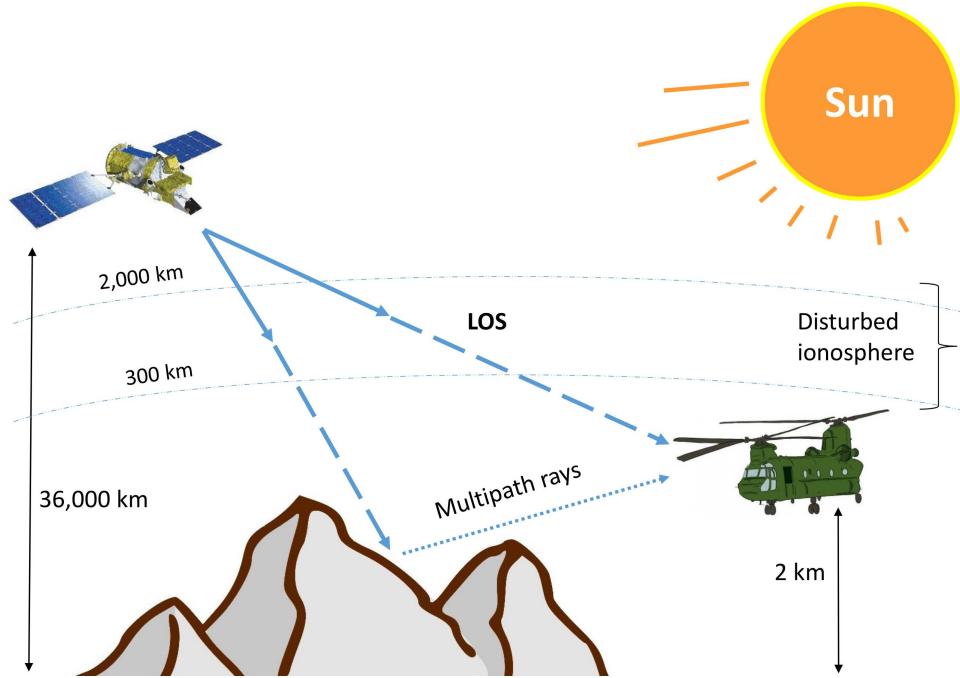


Figure 3.1: Illustration of a communication link between a mobile node and a GEO satellite. The received signal is a combination of the scintillated LOS and multiple reflections of scintillated rays [1].

either be constructive or destructive. If the latter occurs during the same period of the communication protocol when synchronization messages are exchanged, the fading effect is enhanced and a link disruption occurs. The scenario when the multipath fading of scintillated signals occurs is depicted in Figure 3.1.

In mobile satellite communications, there are several factors that contribute to the total signal attenuation. The most significant factor is the free-space loss due to the decrease in power density between a node on the ground and a GEO satellite. The loss is assumed to be recovered by the receiver amplifiers and additional attenuation sources other than the free-space loss are considered. Also, it is assumed that the moving node has a constant speed of 250 km/h (70 m/sec) at a constant altitude of 2 km. Furthermore, two other attenuation sources depicted in Figure 3.1 are considered: Ionospheric scintillation and terrestrial multipath fading. It is pointed out that even though there can be fading on both uplink and downlink, the analysis and simulations are done with regard to the downlink

only.

3.1.1 Ionospheric Scintillation

The Ionosphere is a region in the Earth's atmosphere located between 200 km and 2,000 km in altitude [82, 83]. It is primarily characterized by the presence of ionized gas, *i.e.*, plasma. The ionization within this region occurs by the photo-ionization process as a result of two main events: the daily illumination by the Sun and sporadic Sun disturbance events, such as solar flares or geomagnetic storms [84–86].

The first event repeats every day, with the ionization intensity of the upper layers of the ionosphere varying throughout the day. This event is more intense at latitudes between \pm 40 degrees [83] around the Equator. Just after local sunset, the recombination of electrons and ions begin. However, this process is not uniform throughout a column of the ionosphere with a certain height. Thus, non-uniform density ion regions start to build up during the night, known as ionospheric bubbles, at scales ranging from a few centimeters up to tens of kilometers. This recombination event usually occurs during a daily local time window starting at 20:00 local time and lasting up to local midnight.

Space weather is another factor that can enhance the ionization or cause ionospheric disturbances during any time of day [87]. It also has the potential to extend the disturbance up to higher latitudes. The space weather activity period tends to follow the 11-year solar cycle according to the sunspot number [85]. Depending on the intensity of the solar event, disruptions to satellite communication links can occur if the Earth is in view of the Sun sector where the event occurred [88].

The power of an electromagnetic field traveling through a ionospheric disturbance varies very quickly. This fading effect is called ionospheric scintillation [82], and its intensity is a function of the current ionospheric density gradient faced by the traveling signal.

3.1.2 Multipath Fading

The scenario shown in Figure 3.1 demonstrates the importance of the terrain over which the mobile node is moving. Depending on the terrain's landscape and the electrical properties of its material, the incident rays will be reflected with different intensities and phases.

It was assumed that there is no signal shadowing and the LOS signal component is always present. Also, for the case where the receive antenna is located far from the scatterers, the LOS power can be considered larger than the power of the multipath rays. These reflected rays combine either constructively or destructively with the LOS component at the receive antenna. It is assumed that both LOS and reflected rays are scintillated rays reflected by the terrain profiles considered.

3.1.3 Proposed Channel Model

Several measurement campaigns have assessed the intensity of ionospheric scintillations within the L-band for GPS systems [70, 82, 89]. Based on these measurements, statistical models have been proposed, with the Rician [73, 90] and Butterworth 2nd-order filter [73] models being widely accepted. The recommended approach is to model the scintillation channel using a Butterworth filter since it relies heavily on measurements for defining fading intervals, *i.e.*, the 3-dB autocorrelation time lag, for different frequency bands [90]. However, in the literature there is a lack of such measurements for the scintillation of satellite communications for nodes traveling more than 125 mph (200 km/h) over different terrain profiles. Thus, a Rician channel model is used, which has the probability density function [73]:

$$p(\alpha) = \frac{2\alpha(1+K)}{\Omega} I_0\left(2\alpha\sqrt{\frac{K+K^2}{\Omega}}\right) e^{-\frac{K-\alpha^2(1+K)}{\Omega}}, \quad (3.1)$$

where K is the Rician factor, the variable α is the magnitude of the complex channel response function, it is normalized, $\alpha \geq 0$, and has a second moment, *i.e.*, mean-square for the fading amplitude $\Omega \equiv E[\alpha^2] = 1$.

The scintillation index S_4 is defined by the square-root of the normalized variance of the signal intensity I over a given interval of time, usually is 1 minute with a measurement sample rate of 1 Hz, defined by:

$$S_4 = \sqrt{\frac{E[I^2] - E[I]^2}{E[I]^2}}. \quad (3.2)$$

The Rician parameter that models the ionospheric effects, K_{ion} is related to S_4 by [73]:

$$K_{\text{ion}} = \frac{\sqrt{1-S_4^2}}{1-\sqrt{1-S_4^2}}, 0 \leq S_4 \leq 1. \quad (3.3)$$

The multipath model has a Rician distribution with different parameters from the Rician distribution of the scintillation model. A K-factor function of the reflection coefficient Γ was derived, which, in turn, is a function of the carrier frequency f , incidence angle θ_i , reflection angle θ_0 , and the electrical properties of the terrain's material, such as relative permittivity ε_r and conductivity σ . The approach taken is to define a Rician K-factor function of the reflection coefficient using the classical two-ray propagation model [91], assuming that $\theta_i = \theta_0 = \theta$, where θ is the satellite elevation angle.

The reflection coefficient Γ is given by [92]:

$$\Gamma = \frac{C \sin \theta - \sqrt{(\varepsilon_r - j\chi) - (\cos \theta)^2}}{C \sin \theta + \sqrt{(\varepsilon_r - j\chi) - (\cos \theta)^2}}, \quad (3.4)$$

where $C = \varepsilon_r - j\chi$ for vertical polarization and $C = 1$ for horizontal polarization. Furthermore, χ is given by:

$$\chi = \frac{\sigma}{\omega \varepsilon_0} = \frac{\sigma}{2\pi f \varepsilon_0} = \frac{1.8 \times 10^{10} \sigma}{f}, \quad (3.5)$$

with $\varepsilon_0 = 8.854 \times 10^{-12}$ F/m. The phase difference between the two reflected paths is given by [91]:

$$\Delta\phi = \frac{2\pi}{\lambda} \left(\sqrt{d^2 + (h_t + h_r)^2} - \sqrt{d^2 + (h_t - h_r)^2} \right), \quad (3.6)$$

where λ is the wavelength, d is the distance between the transmitter and receiver antennas, h_t and h_r are the height of the transmitter and receiver antennas, respectively.

The resultant received power P_r is given by the sum of the LOS received power plus the received multipath power, resulting in:

$$P_r = P_t \left(\frac{\lambda}{4\pi d} \right)^2 G_t G_r [1 + |\Gamma|^2 + 2|\Gamma| \cos(\angle \Gamma - \angle \Delta\phi)], \quad (3.7)$$

which is function of transmitter power P_t and the reflection coefficient [92], where G_t and G_r are the transmitter and receiver antenna gains, respectively.

Since the K-factor is defined as the ratio of the LOS power to the multipath power, the former cancels out, resulting in:

$$K = \frac{1}{|\Gamma|^2 + 2|\Gamma| \cos(\angle \Gamma - \angle \Delta\phi)}. \quad (3.8)$$

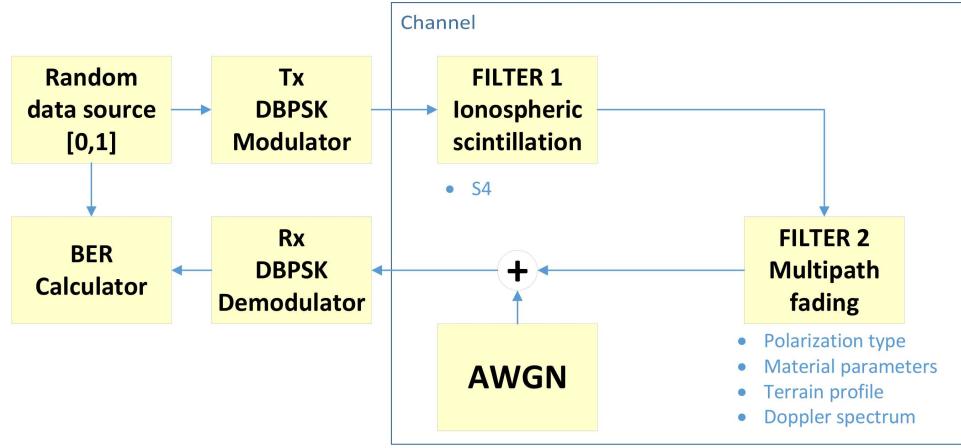


Figure 3.2: Schematic of the mobile satellite communication system simulation testbed implemented in MATLAB. Two Rician channels model the effects of ionospheric scintillation and multipath fading.

It is worth noting that, for the geometry considered in this case, the elevation angle is assumed constant for the area where the mobile node is moving around. For longer distances, one might consider θ to be function of time, especially if the satellite is in LEO.

The nature of this scenario, consisting of a mobile node, suggests that the propagation delay is long [93, 94], in which case a “Gaussian I” type for the Doppler power spectrum is used, suitable for multipath components with long delays in ultra high frequency (UHF) communications [95], which is also proposed as a model for the aeronautical satellite channel [96, 97]. For this Doppler spectrum, the propagation delay τ has an interval $0.5 \mu\text{s} \leq \tau \leq 2 \mu\text{s}$ with a Doppler spread given by $\sigma_{\text{DS}} = 0.45 f_{\text{max}}$, where $f_{\text{max}} = \frac{vf}{c}$ is the maximum Doppler frequency [93, 94], where v is the node linear speed in m/s, f is the carrier frequency in Hz, and c is the speed of light in m/s. For the Gaussian I Doppler power spectrum, the standard deviation σ_N normalized by f_{max} is $\sigma_N = \frac{\sigma_{\text{DS}}}{f_{\text{max}}} = 0.45$.

3.1.4 Simulation Results

The simulation testbed used in this work was implemented in MATLAB [98] and is composed of a transmitter, a channel, and a receiver. Figure 3.2 shows the block diagram of the simulation testbed. Based on the discussion about the channel parameters in the

Table 3.1: Simulation parameters to be combined into four different scenarios

S_4	Frequency	Polarization	Terrain profile	Material
Medium $S_4 = 0.3$	UHF (250 MHz)	Horizontal	Rural Area (RA)	Poor Ground
High $S_4 = 0.9$		Vertical	Hilly Terrain (HT)	

previous section, the implementation of the two main channel impairments, scintillation and multipath, uses Rician channel models, each with different parameters, and the Additive White Gaussian Noise (AWGN) channel representing the noise at the receiver.

For this experiment simulation, random bits are modulated using a Differential Binary Phase Shift Keying (DBPSK) modulation scheme and sent through the channel to the receiver. The ionospheric channel input is the S_4 index. The multipath channel inputs are: polarization type, material parameters, terrain profile, and Doppler power spectrum density distribution. The simulations were performed for all the combinations of parameters shown in Table 3.1.

The values used for the electrical material properties can be found in Table 4.1 in [91]. The \mathcal{L} -path channel models the propagation delay in seconds and the path power in dB for each of the terrain profiles described in Table 3.1. The European Cooperation in Science and Technology (COST) COST 207 [94] recommendations are followed and can be found in Table 7.2 in [93].

The noise is added to the signal using the AWGN channel, which had its E_b/N_0 parameters varied from 0 dB to 20 dB at 2 dB step for the BER curve acquisition. Then, 20,000 symbols were transmitted per frame with a total of 100 runs for each E_b/N_0 value, such that the minimum value for bit error for each E_b/N_0 was guaranteed to be 100. The maximum Doppler shift considered was 60 Hz for a node moving at 70 m/sec and a carrier frequency of 250 MHz. Then, the Doppler spread is $\sigma = 0.45f_{\max} = 27$ Hz. Since $\sigma \ll f_{\max}$, the channel is expected to be slow fading. Two S_4 indexes were considered: $S_4 = 0.3$ for medium scintillation and $S_4 = 0.9$ for high scintillation. Even though the considered satellite uses circular polarization, simulations were performed for two different polarizations: horizontal and vertical. Thus, the following results are bounds on BER performance, not

Table 3.2: Rician K-factors for ionospheric scintillation and terrestrial multipath fading filters

Filter 1		Filter 2	
Parameter	K-factor	Parameters	K-factor
Medium $S_4 = 0.3$	20.7104	Horizontal pol.	6.2943
High $S_4 = 0.9$	0.7727	Vertical pol.	16.6698

precise estimates. Two terrain profiles were considered for the path gains and delay profiles: 4-path Rural Area (RA) and 6-path Hilly Terrain (HT) channel models [93].

For the reflection coefficient of the K-factor for Filter 2 in Figure 3.2, which uses the two-ray model, the antenna heights considered as shown in Figure 3.1 are 36,000 km and 2 km for the satellite and mobile node, respectively. The range between the two antennas is 36,497 km and was computed using the STK software by AGI [99] for the Takur Ghar region ($33^{\circ}20'35''$ N, $69^{\circ}12'52''$ E) to the UFO-10 satellite (77.5° E).

The material parameters are defined in terms of their material electrical properties. It was considered a Poor Ground material type with relative permittivity $\epsilon_r = 4$ and conductivity $\sigma = 0.001$ [91]. Regarding the terrain profiles, the values used for the path gain and propagation delay were taken from the COST 207 [94] recommendation. The K-factors for the scenarios considered are described in Table 3.2. Since the terrain material, frequency, and antenna distances were constant for this experiment, the K-factor changes with the polarization type only, but is function of all these parameters.

After running the simulations for the combinations of channel parameters in Table 3.1, the results are split in terms of terrain profile and polarization. For a performance reference, the BER for the link using only the AWGN channel is given. Next, results were obtained by adding Filter 1 in order to analyze the effect of simulated ionospheric scintillation. The Filter 2 was connected in series with Filter 1, allowing for the analysis of terrestrial multipath fading of ionospheric scintillated signals. The results for RA and HT terrain profiles are shown in Figure 3.3 and Figure 3.4, respectively. Both figures show distinct left and right

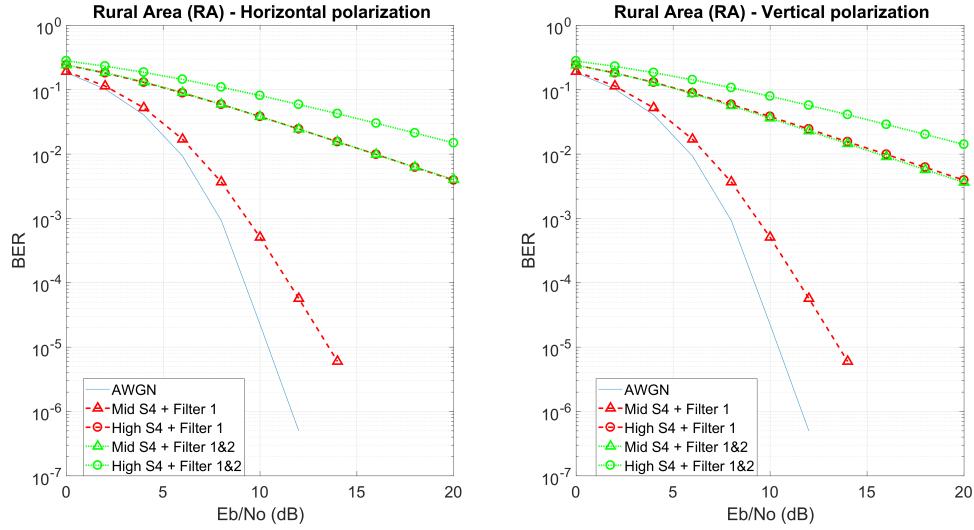


Figure 3.3: BER performance of mobile communication link for a Rural Area (RA) 4-path channel model under ionospheric scintillation. Left panel shows results for horizontal polarization and right panel shows results for vertical polarization.

panels for the simulations using horizontal and vertical polarization, respectively.

From Figure 3.3, it is clear that the BER performance difference between polarizations is minimal given the scenario assumptions as well as transmitter and receiver configurations. Furthermore, the BER performances for a scenario with high S_4 using only Filter 1 and the scenario with medium S_4 using Filters 1 and 2 are similar. Both panels in Figure 3.3 show a great decrease in BER performance for medium S_4 while comparing a scenario with only scintillation and another with multipath of scintillated signals (dashed and dotted lines with triangle markers, respectively). This decrease is more than two orders of magnitude for $E_b/N_0 = 12$ dB. For high S_4 , the BER performance is worse as expected. However, the decrease in performance in this last case is smaller, less than one order of magnitude over the entire range of E_b/N_0 simulated.

From Figure 3.4, minimal BER performance differences are also observed between the two polarizations. As expected, the overall BER performance for the two polarizations in the HT terrain profile are worse than the performances in the RA profile. The main performance difference between the profiles are for the multipath fading of ionospheric scintillated signals, from 3×10^{-2} to 1.5×10^{-2} for medium S_4 and from 2×10^{-2} to 3×10^{-3} for high S_4 , both

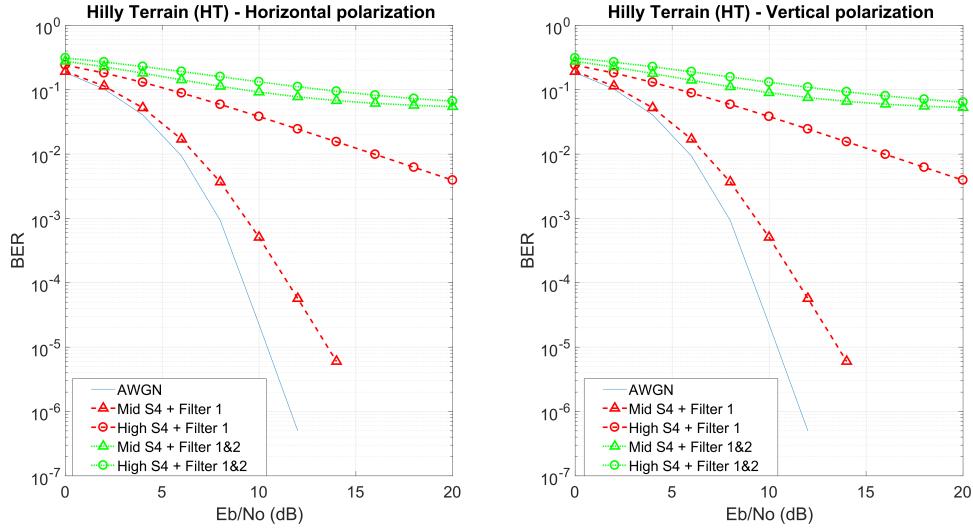


Figure 3.4: BER performance of mobile communication link for a Hilly Terrain (HT) 6-path channel model under ionospheric scintillation. Left panel shows results for horizontal polarization and right panel shows results for vertical polarization.

for $E_b/N_0 = 20$ dB.

3.1.5 Conclusions

The BER performance of a mobile satellite communication system between a GEO satellite and a moving node under multipath fading of scintillated signals was analyzed while traveling through different terrain profiles under different scintillation indexes. In order to implement the channel, a model composed of two Rician channels connected in series was proposed. Furthermore, the K-factor in terms of the terrain's reflection coefficient was derived for the Rician channel implementing the multipath fading.

Simulations for RA and HT profiles were performed. The performance for horizontal and vertical polarization were also analyzed. As expected, the performance for the HT profile was found to be worse than the RA profile. The root cause assumed in [81] alone does not seem to justify the possible message delivery failure based on the results shown by Fig. 3.4, which assumes the scenario parameters and conditions close to the mountain region. Depending on the polarization type used, the BER performance can be even worse depending on the E_b/N_0 being used. Thus, there is a significant BER performance de-

crease when the communicating node changes from a scenario in which only ionospheric scintillation is present, such as a node flying at high altitudes, to another in which there is multipath fading of ionospheric scintillated signals, with the performance decrease reaching some orders of magnitude depending on the E_b/N_0 being used.

3.2 Step-by-Step Implementation of a 3D Rain Field Model for LEO Orbit Attenuation Time Series Synthesizers

It is difficult to study the behavior of rain fading since it varies at each location on Earth, as well as with the azimuth, elevation angle, and time of year. Due to the current lack of publicly available rain fading time series data at Ka-band, an alternative approach is to use rain cell and rain field models that take into account historic statistical data of rain rate measurements [100] in order to generate synthetic time series of rain attenuation profiles for a certain location [74].

The 3D synthetic rain field model simulates rain rates in mm/h for a specific geographical area, and converts them into attenuation values based on the varying LOS geometry at each time instant. The rain is characterized by rain cells distributed over a field. A rain cell is assumed to be the region surrounding a local maximum rain rate limited by a minimum rain rate, known as rain rate threshold [74], which is considered a designer parameter. For a specific area, when the actual rain rate is higher than this threshold, it is assumed that it is raining.

The exponential spatial rainfall cell distribution (EXCELL) model [101] used in this work assumes rain cells having an exponential profile with a central peak. Thus, providing the rain rate in mm/h distributed in an horizontal plane, the rain cell model is given by:

$$R(r) = R_M \exp\left(\frac{r}{\rho_0}\right), \quad (3.9)$$

where r is the radial distance from the rain cell center, R_M is the peak rain rate at the center, and ρ_0 is the characteristic distance from the cell center where the rain rate reduces to e^{-1} of R_M . The EXCELL model is used by the rain field model proposed by Goldhirsh [102]. Knowing the rain rate variation over a specific area, rain attenuation can be computed for

a given path distance, elevation angle, and rain height. The following is a brief step-by-step description of the algorithm used in this work (Eqs. (3.10)–(3.14) are derived in [102]):

1. Determine the absolute complementary cumulative distribution function (CCDF) for a specific location using the MATLAB code provided by Rec. ITU-R P.837-5 [103]. The code suggested by Rec. ITU-R P.837-6 [100] can only convert between higher integration times down to 1-minute integration times. Notice that the rainfall rates exceeded by a percentage of the year for each location on Earth, given by the source code on Rec. ITU-R P.837-5, are already in the 1-minute integration time scale. The absolute CCDF is shown in Figure 3.5.
2. Determine the conditional CCDF by dividing the absolute CCDF by the exceeded percentage value (found in the absolute CCDF corresponding to the assumed rain rate threshold). A rain rate threshold R_q of 0.5 mm/h is assumed, which translates into rain exceedance of 4.84 % of the year in the absolute CCDF for the region considered. The conditional CCDF is also shown in Figure 3.5.
3. Determine the modeled rain rate distribution parameters: the probability of rain on the path P_0 , the percentage probability of rain in an average year R^* , and the constant κ in the model proposed in [102], by fitting the conditional CCDF to $P(R_q)$, the probability that rain rate is greater than R_q , given by:

$$P(R_q) = P_0 \left[\ln \left(\frac{R^*}{R_q} \right) \right]^\kappa. \quad (3.10)$$

Assuming that R^* needs to be bigger than the maximum measured rain rate [102], using $R^* = 250$ mm/h, the fitting was performed using MATLAB curve fitting tool, which resulted in $P_0 = 2.291 \times 10^{-5}$ and $\kappa = 5.956$.

4. Determine the amount of rain cells, within the $A_0 = 100 \text{ km} \times 100 \text{ km}$ area for pairs of peak rain rates within an interval of $\delta = 5 \text{ mm/h}$, from the minimum peak rain rate $R_M = 2.5 \text{ mm/h}$ up to the maximum being $R^* = 250 \text{ mm/h}$ according to the expression:

$$\text{NUM}(R_Q) = \left(\frac{N(R_M) + N(R_M + \delta)}{2} \right) \delta A_0, \quad (3.11)$$

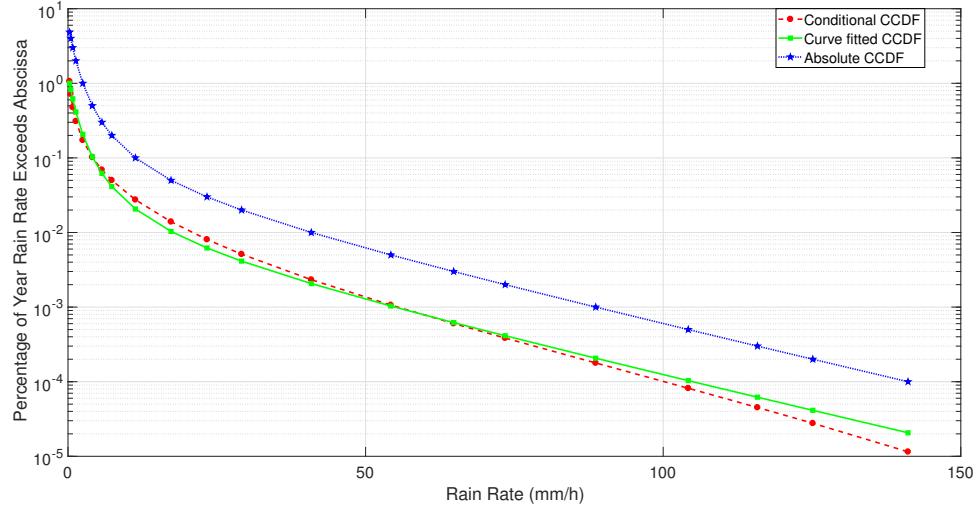


Figure 3.5: Absolute and conditional CCDF's for the percentage of time in one year that the rain rate exceeds the rain rate value at the abscissa for the Worcester area based on the ITU-R P.837-5. Curve fitting for conditional CCDF shows close agreement between the computed conditional CCDF and the theoretical model assumed by the ITU recommendation.

where each $N(R_M)$ is the density of rain cells in (number of cells)/km²/(mm/h), given by:

$$N(R_M) = \left(\frac{P_0}{R_M 2\pi \rho_0^2} \right) \kappa(\kappa - 1)(\kappa - 2) \left[\ln\left(\frac{R^*}{R_M}\right) \right]^{\kappa-3}, \quad (3.12)$$

and ρ_0 is given by:

$$\rho_0(R_M) = \frac{10 - 1.5\log_{10}R_M}{\ln\left(\frac{R_M}{0.5}\right)}. \quad (3.13)$$

5. For each peak rain rate interval, randomly assign a peak rain rate value from the respective interval using a uniform distribution. Then, randomly distribute these rain cells throughout the considered area with a uniform distribution.
6. Divide the total area into smaller boxes of 0.5 km × 0.5 km area and determine the total rain rate at the center of each box, which is given by the contribution of all cells that lies within that particular box. After identifying all cells belonging to one box,

compute their individual contribution by:

$$R(U, V) = R_M \exp \left(-\frac{\sqrt{(x-U)^2 + (y-V)^2}}{\rho_0} \right), \quad (3.14)$$

where R_M is the randomly assigned peak rain rate centered at (x, y) within the box centered at (U, V) , and ρ_0 is computed using Eq. (3.13). Then, the total rain rate contribution for each box is the sum of all the individual contributions of the cells within that box.

The resultant contribution of all peak rain rates within the same box results in the peak rain rate $R(U, V)$ for the (U, V) location point within the area considered. One example of a simulation output of the implementation of the 3D rain field synthesizer described above is shown in Figure 3.6, centered on Worcester, MA, USA. This field contains the $R(U, V)$ grid point values. Since each $R(U, V)$ value is centered at a box with area dimensions of $0.5 \text{ km} \times 0.5 \text{ km}$, the resultant attenuation value will not be smooth, as shown by the attenuation time series. However, in order to represent a natural transition of rain rates between the grid points, a smoothing approach, such as spline interpolation, was used. The rain rate histogram for this field is shown in Figure 3.7. For the simulations, a total of 44 rain cells randomly distributed across the entire Worcester, MA, USA was considered, and the surrounding areas were used.

3.2.1 Orbit Simulator, Ground Path Acquisition and Rain layer Intersection

In order to generate realistic satellite passings, the license-free version of STK software by AGI [99] was used. The LEO satellite considered in this work is the ISS, a science laboratory that orbits the Earth at an average altitude of 420 km and a linear speed of 27,600 km/h [84]. The passings over one month were simulated, assuming a ground station (GS) were located on the rooftop of Atwater Kent Laboratories ($42^{\circ}16'30.8''$ N, $71^{\circ}48'25.2''$ W), Worcester Polytechnic Institute (WPI), Worcester, MA, USA.

For the analysis in this work, a pass that lasted 512 seconds is considered, with an elevation angle range of (5.02° – 88.593°) and a slant path range of (422.26 km–1870.24 km), as shown in Figure 3.8.

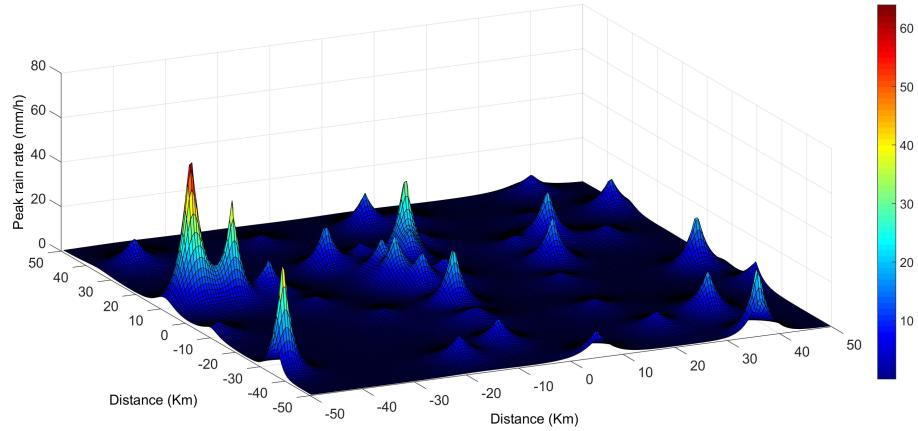


Figure 3.6: Simulated peak rain rate field area of $100 \text{ km} \times 100 \text{ km}$ with resolution of $0.5 \text{ km} \times 0.5 \text{ km}$ centered on Worcester, MA, USA. Rain rates provided by Rec. ITU-R P.837-5, the EXCELL rain cell model, and Goldhirsh rain field cell distribution model were combined to synthesize the resultant rain field.

According to Rec. ITU-R P.839-4 [104], the rain height for an antenna at the chosen location is 4.53 km. Using the total slant path distance and elevation angle profiles provided by STK, the distance between the GS and the point that intersects the rain layer for each elevation angle is computed. This resultant distance experiences rain and is used to compute the total rain attenuation for a specific attenuation and total rain attenuation along the path, described below by Eqs. (3.15) and (3.16) at each time instant. The rain rates are obtained by the simulated synthetic rain field, for each elevation and azimuth angle that represents the satellite pass movement in sight of the GS. Figure 3.9 shows the rain rate reading during the considered satellite pass at the LOS intersection with the synthesized rain layer, accordingly to the elevation angle profile shown in Figure 3.8.

To compute the rain attenuation at each second, the range from the ground antenna up to the point intersecting the rain layer is specified. However, along this path the rain rate may vary. Thus, for each segment with a different rain rate and range, its attenuation contributions are computed, as shown by Figure 3.10. For those segments that are not covered by rain, only the slant path loss is accounted for and summed to the total attenuation for that time instant. This entire process repeats for each time instant, in this case,

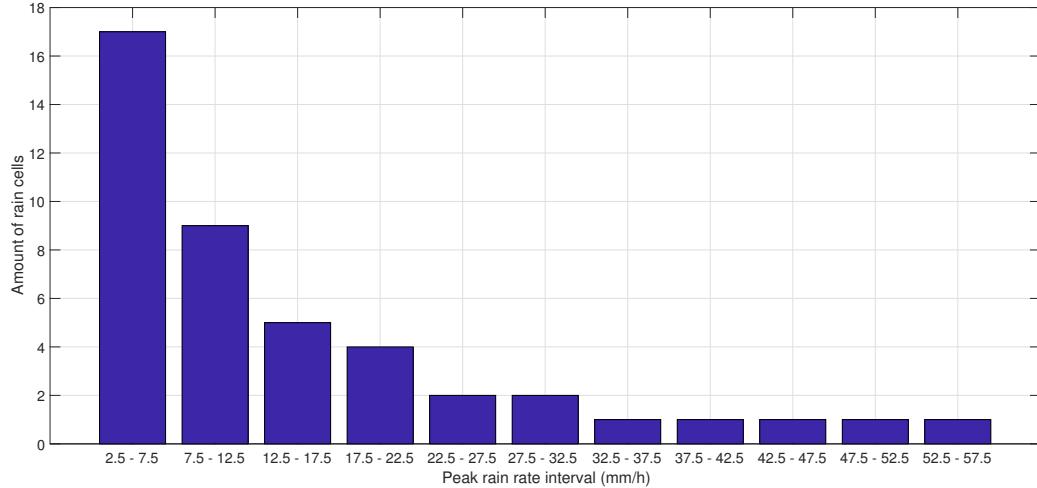


Figure 3.7: Rain cell histogram of the peak rain rate field for an example of the simulation run shown in Fig. 3.6. The resultant peak rain rate contribution in $0.5 \text{ km} \times 0.5 \text{ km}$ resolution follows the log-normal distribution of the rain rate exceedance probability.

each second, due to the satellite motion, tracked by variations of the elevation and azimuth angles that results in changes on the LOS intersection point at the rain field, as well as the distances involved.

Note that the rain rate values shown in Figure 3.9 are the ones read at the farthest-most distance from the antenna, located at d_n , having a value of R_n in Fig. 3.10, for each time instant. Thus, at each time instant, as mentioned above, there is a set of rain rate values for each segment when there is rain between the antenna and the point in which the LOS intersects an imaginary plane at the rain height. This work does not consider elevation angles below 5° since that would cause the LOS to never cross this imaginary plane.

According to Rec. ITU-R P.838-3 [3], the attenuation due to the rain depends on the LOS range being affected by rain $L_r(k)$, carrier frequency, and polarization type. For each time instant k , $L_r(k)$ is segmented into i segments, each with different values for range $\Delta l_i(k)$ and peak rain rate $R_i(k)$. In the simulations the values used for $R_i(k)$ are the grid points $R(U, V)$ values computed for the 3D rain field. Each segment has its peak rain rate value converted into specific attenuation value $\gamma_{R_i(k)}$ (dB/km) by:

$$\gamma_{R_i(k)} = \nu \cdot (R_i(k))^\alpha, \quad (3.15)$$

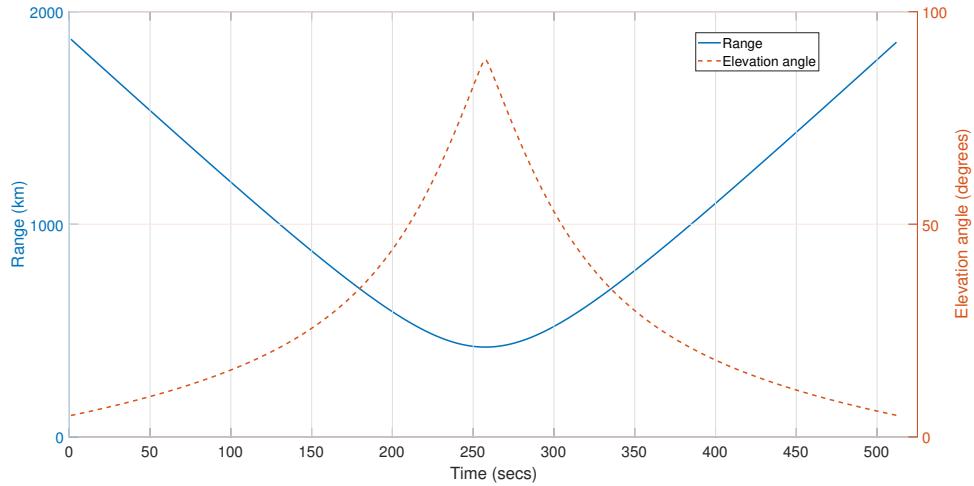


Figure 3.8: Slant path and elevation angle profile for one ISS pass over a GS located at WPI, Worcester, MA, simulated in STK. The pass is 512 seconds long. The minimum slant path distance is achieved during the maximum elevation angle.

where ν and α are dependent on frequency, polarization, and elevation angle, computed using tables provided by [3]. The rain attenuation contribution of all segments in dB at time instant k is computed by:

$$A_{\text{rain}}(k) = \sum_{i=1}^n \gamma_{R_i(k)} \cdot \Delta l_i(k), \quad (3.16)$$

which is the sum of the attenuation over each segment. Computing $\Delta l_i(k)$ requires some attention since, given the peak rain rates distributed across the 3D rain field, one needs to account for the LEO motion in terms of its elevation and azimuth angles, as well as for the rain height, at each time instant k . Figure 3.10 illustrates the rain rate segmentation during a generic time instant k .

For the particular case of circular polarization, which is assumed in this work, it turns out that it is not a function of elevation angle. For the Ka-band, we used $\nu = 9.2249 \cdot 10^{-5}$ and $\alpha = 1.0028$. The relationship between γ_R and R is shown in Figure 4.2 for Ka-band and S-band. More details on the resultant time series are provided in Section 4.1.7.

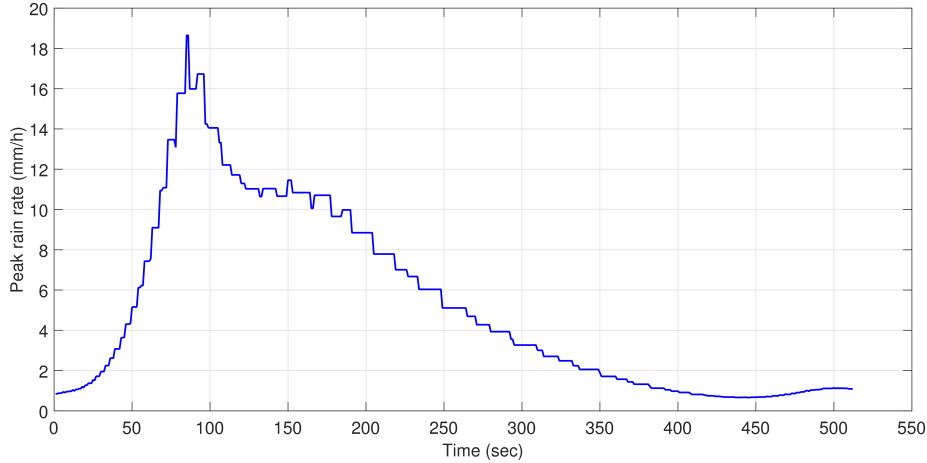


Figure 3.9: Peak rain rate readings in mm/h for the LOS intersection with the synthesized rain layer during the satellite pass considered. For the same synthesized rain field, different passes having different elevation and azimuth angle profiles might result in different rain rate readings.

3.3 Proposed Rain Fading Time Series Synthesizer for LEO Orbit Based on ITU-R P.1853-1

Using the rain attenuation time series synthesizer implementation described above, a technique to synthesize rain fading time series for LEO orbits in any orbital inclination is proposed. This approach advances the applicability of ITU-R P.1853 [105] for scenarios where rain fading affects communications links between a fixed or mobile GS and a LEO satellite.

3.3.1 Overview of ITU-R P.1853

Released in 2013, the ITU-R P.1853 [105] only generates rain attenuation time series for fixed elevation angles that can be used only with GEO satellites. Its current version requires three inputs: a CCDF, the probability of rain on the path P_0 , and the attenuation levels exceeded in dB for the percentages of time used to compute the CCDF. Following the procedures described in ITU-R P.1853, the ITU-R P.837 [100] is used to compute both the CCDF of the rain rate and P_0 for a specific location. The time percentages used to

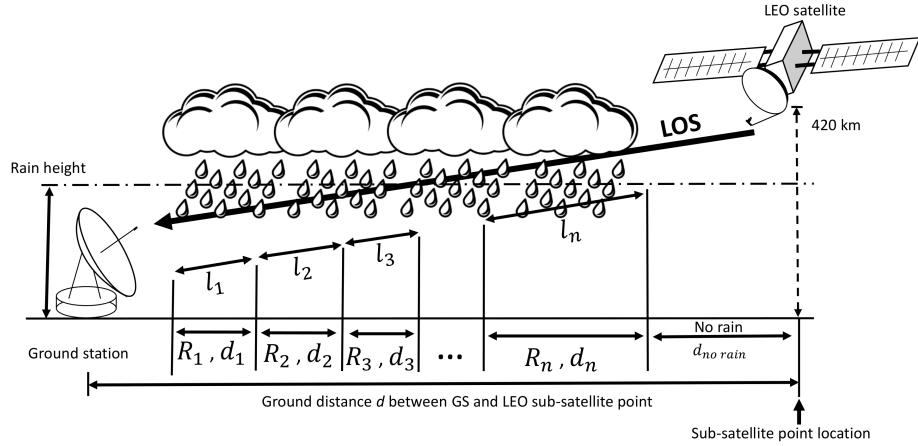


Figure 3.10: Segmentation of rain attenuation at time instant k into different peak rain rate $R_i(k)$ and segment LOS range $\Delta l_i(k)$. Knowledge of elevation angle and sub-satellite location allows computation of $\Delta l_i(k)$. Segments without rain contribute with only clear sky slant path attenuation.

compute these are assumed to be equal to:

$$[0.01 \ 0.02 \ 0.03 \ 0.05 \ 0.1 \ 0.2 \ 0.3 \ 0.5 \ 1 \ 2 \ 3 \ 5 \ 10]. \quad (3.17)$$

The rain rates in the CCDF are converted into attenuation levels in dB using the ITU-R P.618-12 [71] based on parameters from the ITU P.839 [104] and ITU-R P.838-3 [3], such as rain height and circular polarization for a carrier frequency $f_c = 26$ GHz.

The final step consists of feeding all of these values to the MATLAB implementation, also provided by the ITU Study Group 3 [106], which is limited to synthesizing time series only for GEO orbits.

3.3.2 Proposed Rain Fading Time Series Synthesizer for LEO Orbit

The main synthesizer design challenge was the generation of a rain attenuation time series that is computed for a varying elevation angle at each time instant. In order to achieve this, the conversion of a collection of synthesized GEO time series into one LEO time series was proposed. As shown in Figure 3.11, for different fixed elevation angles there is a different attenuation value for the same current channel conditions.

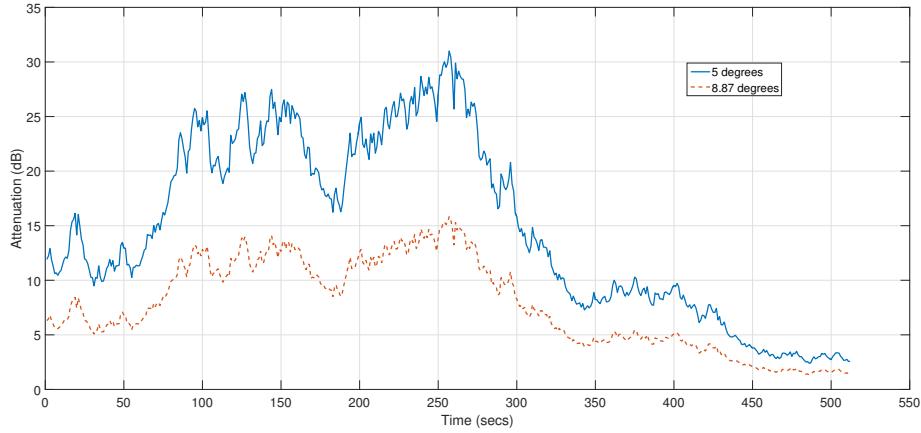


Figure 3.11: Rain attenuation time series examples of outputs for two different fixed elevation angles using the available implementation of ITU-R P.1853 at Ka-band. For varying elevation angles, as is the case of LEO orbits, several time series would be required, but just a few points of each would be used.

However, for LEO satellite orbits, the elevation angles continuously change over time following a non-linear relationship with the time, for each pass. Also, the attenuation time series for GEO is different for each elevation angle. Thus, the attenuation levels should transition between these attenuation time series that have fixed elevation angles. The goal is to generate a time series such that the resultant amplitude changes across different fixed elevation angle time series, based on the elevation angle of a LEO pass.

One approach would be to generate an amount of attenuation time series that would be equal to the number of data points in the elevation angle time series. However that would be computationally costly since only one attenuation data point would be used out of each attenuation time series generated for each fixed elevation angle. Another approach would be to generate an attenuation time series generated for a fixed number of elevation angles and to obtain the resultant time series by roughly hopping between them. The problem with this method is that it would not represent the smooth slant path loss for a LEO satellite, as shown in Figure 4.9, due to the abrupt value changes. Another issue with this approach would be to determine at what time instant the output should hop from one fixed elevation angle time series to another. Figure 3.12 highlights this issue by illustrating these rough

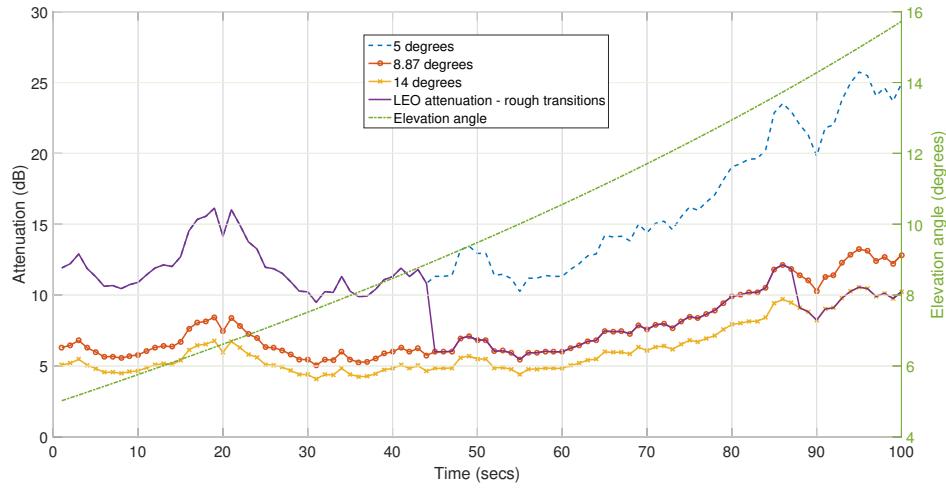


Figure 3.12: Rough transitions between time series examples shown in Fig. 3.11 with fixed elevation angles at non specified time instants. LEO elevation angle shows how smooth these transitions should be.

changes for time series shown in Figure 3.11.

To solve this interpolation problem, it was proposed to use the elevation angle profile obtained from an orbital simulator such as STK [99] to aid the interpolation of attenuation levels between time series with fixed elevation angles. This proposed method is called “nonlinear weighted two dimensional interpolation of rain attenuation time series for LEO satellites based on elevation angle profile using ITU-R P.1853”.

It is assumed that six elevation angle intervals between the minimum and maximum values are obtained for a certain satellite pass, as shown in Table 3.3. Thus, in the proposed approach the goal of generating a LEO time series containing 512 points was achieved using only six attenuation time series. Note that there is a trade-off between accuracy of the generated LEO time series and the number of GEO time series generated. Accuracy performance analysis, such as statistical comparisons between the outputs of the proposed method and measurement time series, will be possible as soon as measurement data become publicly available.

An elevation angle profile is considered for a single pass ranging between 5° and 83.41° . Each interval time series was generated using the first elevation angle value of that interval.

Table 3.3: Elevation angle intervals

Interval	Elevation angle (in degrees)
1	5 – 8.869
2	8.869 – 14
3	14 – 21.37
4	21.37 – 33.48
5	33.48 – 56.48
6	56.48 – 83.41

Within an elevation angle interval, at each discrete time instant k , one LEO attenuation value $\text{Att}_{\text{LEO}}(k)$ is computed by:

$$\text{Att}_{\text{LEO}}(k) = [\text{Att}_{\theta_1}(k) \cdot t_1(k)] + [\text{Att}_{\theta_2}(k) \cdot t_2(k)], \quad (3.18)$$

where $\text{Att}_{\theta_1}(k)$ and $\text{Att}_{\theta_2}(k)$ are the attenuation values from the time series computed using the fixed elevation angles θ_1 and θ_2 , and the weights $W_1(k)$ and $W_2(k)$ are given by:

$$t_1(k) = \frac{|\theta_1(k) - \theta_{\text{ref}}(k)|}{\theta_{\text{diff}}} \quad (3.19)$$

and

$$t_2(k) = \frac{|\theta_2(k) - \theta_{\text{ref}}(k)|}{\theta_{\text{diff}}}, \quad (3.20)$$

with

$$\theta_{\text{diff}} = |\theta_1 - \theta_2| \quad (3.21)$$

being constant during one elevation interval between $[\theta_1, \theta_2]$. The function $\theta_{\text{ref}}(k)$ is the satellite elevation angle at instant k that drives the non-linearity for the weights. The implementation code of this proposed method is available online via GitHub [2]. Figure 3.13 shows a smoother and more natural transition between the fixed elevation angle time series relative to those transitions shown in Figure 3.12 as a result of the proposed method. Also, the transitions occur at the correct time since they are driven by the elevation angle time

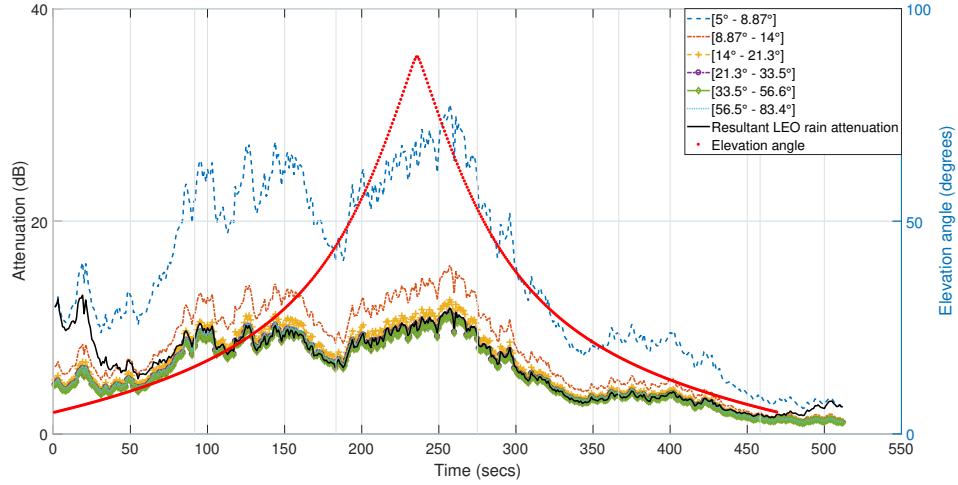


Figure 3.13: Resultant LEO rain attenuation time series example showing computed Att_{LEO} values at each time instant k using proposed method “nonlinear weighted two dimensional interpolation of rain attenuation time series for LEO satellites based on elevation angle profile and ITU-R P.1853.” Elevation angle and rain attenuation time series for each elevation sector is shown for reference.

series profile. It is worth noting that, for the total slant path loss, the slant path distance loss for the LEO satellite needs to be added to the rain attenuation time series.

3.3.3 Conclusions

Due to the current lack of measurement data availability with respect to rain fading at Ka-band for LEO satellites, two different methods to synthesize rain attenuation profiles were used, one using a 3D rain field and another that was proposed by the author, based on the ITU-R P.1853. Future measurement campaigns of rain fading at Ka-band from LEO satellites will allow these synthesizers to be validated, or calibrated, in order to allow for the convenience of testing communication algorithms while varying several channel attenuation parameters without the need for performing in-field measurements.

3.4 LMS Channel Experiencing Rain Fading at Ka-Band from a GEO Satellite Downlink

As previously mentioned, due to the lack of available measurements, especially of rain fading impairments at Ka-band, a time series had to be synthesized based on the recently updated ITU recommendations. In order to generate the synthetic rain attenuation levels, a method proposed by ITU-R P.1853 [105] was employed, in which three inputs were required: CCDF; the probability of rain on the path, P_0 ; and the attenuation levels exceeded in decibels for the percentages of time used to compute the CCDF. In this work, the following time percentages were used:

$$[0.01, 0.02, 0.03, 0.05, 0.1, 0.2, 0.3, 0.5, 1, 2, 3, 5, 10]. \quad (3.22)$$

Using the ITU-R P.837 [100] and focusing on the location of Worcester, MA, USA ($42^{\circ}16'30.8''$ N, $71^{\circ}48'25.2''$ W), a CCDF for rain rate and P_0 were computed. Then, the rain rate values from the CCDF were converted into attenuation levels (dB) based on the ITU P.618-12 [71] specifications using parameters given by ITU P.839 [104], such as the rain height, as well as parameters given by ITU-R P.838-3 [3], such as the circular polarization for a carrier frequency f_c at 26 GHz. Finally, using the method proposed by ITU-R P.1853 [105], a rain attenuation time series was generated, which assumed an elevation angle of 34° , at a sampling rate of 1 Hz across a time period of 84,600 seconds. A snapshot of 512 seconds of a generated times series is shown in Figure 3.14, which is the same time series used in other scenarios, further described in later sections.

3.4.1 LMSS Channel Simulation at Ka-Band

Given the ground station mobility, the Earth-space LMS service channel time series was generated based on ITU-R P.681-8 [107] communicating across the Ka-band with a GEO satellite. For a more realistic propagation simulation, both statistical and stochastic models for mixed propagation conditions, such as rural, wooded, urban, and suburban areas, were also employed. This model computes the cumulative distribution function (CDF) using a semi-Markov two-state model represented by a non-shadowed (good) state and a shadowed

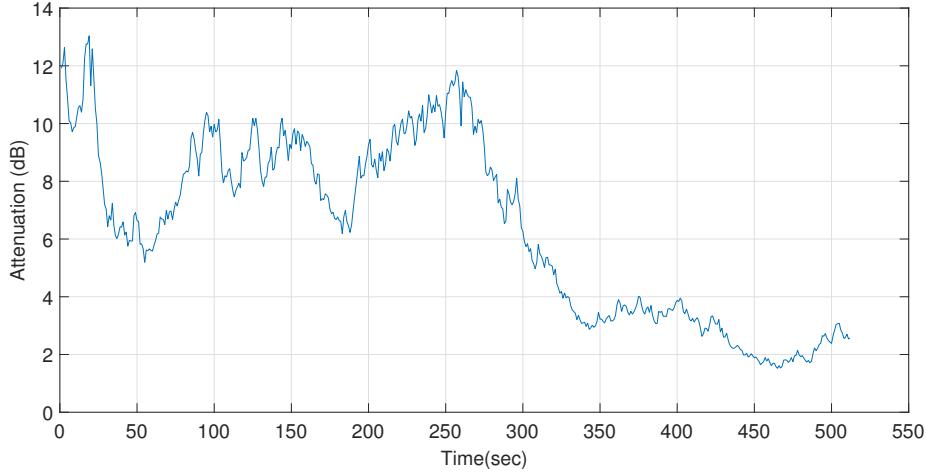


Figure 3.14: Rain attenuation time series for Ka-band based on ITU-R P.1853. Note that during the entire time interval the channel is facing rain fading.

(bad) state, where the state duration follows a log-normal distribution and the signal within each state follows a Loo distribution. The log-normal shadowing affects only the direct component, while the diffuse multipath components have a constant average power [108].

It is worth noting that ITU-R P.681-8 recommends the usage of parameters measured specifically for a suburban area at 11.7 GHz rated to be used by any frequency between 10 GHz and 30 GHz, thus covering the lower portion of the Ka-band. The ground vehicle speed v_m was set to 33 km/h, while the elevation angle was set to 34° . Furthermore, the angle between the vehicle heading vector projected on the ground and the satellite azimuth vector projected on the ground was specified to be equal to 0° . The sampling frequency f_s was assumed to be equal to 10,000 samples per second, which is a function of the carrier frequency f_c , as well as of the maximum speed of the vehicle v_{\max} , given by [109]:

$$v_{\max} \leq \frac{c f_s}{2 f_c}, \quad (3.23)$$

The synthesizer diagram block, shown in Figure 3.15, was implemented in MATLAB (source code available from Reference [2]). Instead of the recommended Jakes model, a 10th-order low-pass Butterworth filter with normalized 3-dB cutoff frequency equal to

$$f_{\text{cutoff}} = \frac{f_{\text{Doppler}}}{f_s/2}, \quad (3.24)$$

was used to reproduce the fast variations due to multipath since it is more realistic for an LMS channel than the Jakes model [110]. Note that $f_{\text{Doppler}} = \frac{v_m f_c}{c}$ is the maximum Doppler frequency. The time series was generated as follows. Using the input parameters provided by the ITU-R P.681-8, parameters are generated for the two states as well as for the transitions between them (for implementation purposes, transitions are considered states too). Then, a complex time series for both direct and multipath components were generated. Since the direct signal suffers from slow fading, it is filtered by a low-pass filter with transfer function:

$$T(S_{\text{signal}}) = \frac{\sqrt{1 - \rho^2}}{1 - \rho S_{\text{signal}}^{-1}}, \quad (3.25)$$

where

$$\rho = \exp\left(\frac{-v_m (1/f_s)}{L_{\text{corr}}}\right) \quad (3.26)$$

and L_{corr} is the correlation distance given by [107]. The phase of the direct signal is defined by the maximum Doppler frequency, whereas the multipath series is filtered using the Butterworth filter. Note that the proposed synthesizer implementation differs from that of Reference [110] since the latter uses a maximum transition rate (shadowing slope) of 5 dB/m whereas the former uses the recommended linear interpolation of the parameters during all transitions between good (bad) and bad (good) states. This leads to a more natural transition of the mean and standard deviations of the Gaussian random variables used to generate the raw series for both direct and multipath signals.

The proposed implementation generates data at a sampling rate of f_s , resulting in a total of 5.12 million samples. However, to comply with the rain attenuation time series, the LMS synthesizer output is down-sampled to 512 samples, as shown in Figure 3.16.

For the scenario with a mobile LMS channel during a rain fading event, the previous two time series were added together. A step-by-step block diagram for the algorithm of the complete fading time series generator is provided in Figure 3.17. The resultant attenuation time series is shown in Figure 3.18. Several aspects that differ between the proposed analysis and those found in Reference [111] include the following: (i) The power attenuation over both direct and multipath signals for both shadowed and non-shadowed scenarios is assumed; and (ii) The proposed approach uses synthesized time series based on the updated

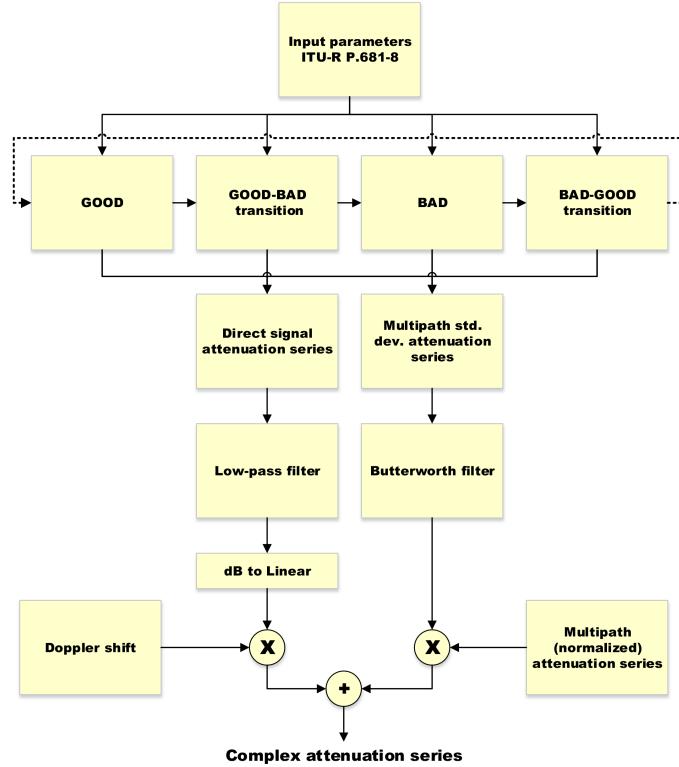


Figure 3.15: LMS channel simulator diagram block based on ITU-R P.618-8. After the good or bad state time series generation, the same input parameters are provided to the state attenuation time series synthesizers, including good, bad and their transitions. MATLAB code implementation provided at Reference [2].

ITU recommendations, with the goal of analyzing the attenuation values on a per-second basis rather than an annual shortage percentage. However, both studies assume that the rain possesses minimal influence on the probability density function (PDF) with respect to shadowing events in clear sky scenarios. Consequently, the shadowing resulting from both direct and multipath rays, as well as the rain attenuation from these same direct and multipath rays, can be considered to be two independent stochastic processes.

When comparing the CDF of the LMS channel time series (shown in Figure 3.16) with the CDF of the same LMS time series but with the effects of rain attenuation at Ka-band included (shown in Figure 3.14), one can observe that the resultant channel possesses a log-

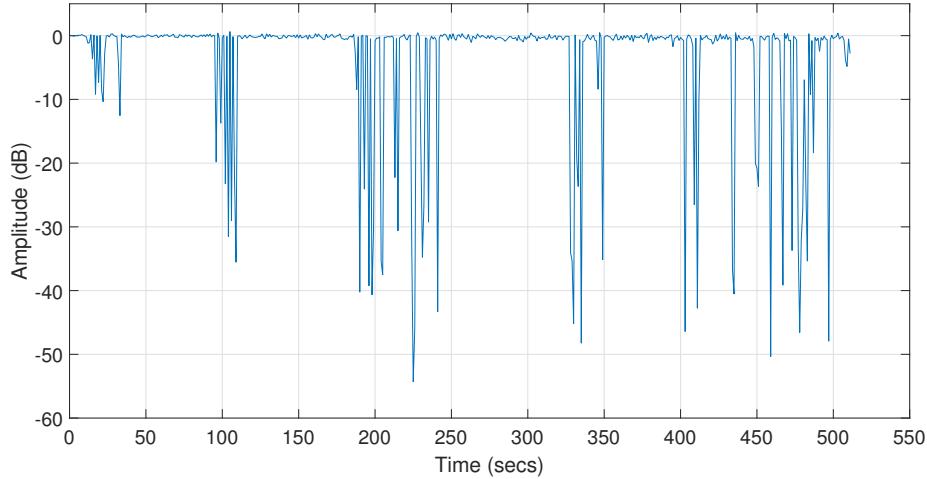


Figure 3.16: Example of an LMS channel synthetic attenuation time series at Ka-band based on ITU-R P.618-8. Shadowing due to mobile terminal LOS blockage by buildings and trees are represented by deep attenuation values, where the good and bad states are Loo distributed.

normal PDF since only the mean of the LMS PDF changes, which is observed in Figure 3.19. Thus, it can be deduced that an LMS channel experiencing from rain fading possesses a smaller mean.

3.4.2 Conclusions

A detailed description of the implementation of an LMS channel synthesizer was given together with the addition of rain attenuation time series affecting the downlink from a GEO satellite at Ka-band. The modularity of the implementation allows generation of time series for different environments, such as suburban or rural, as well as for different sample rates.

3.5 Chapter Summary

In this chapter, two new satellite channel models were proposed aiming to fill a few research gaps in the literature. A new Rician factor function of the reflection coefficient was derived, allowing signal multipath effects to be more realistic while accounting for the

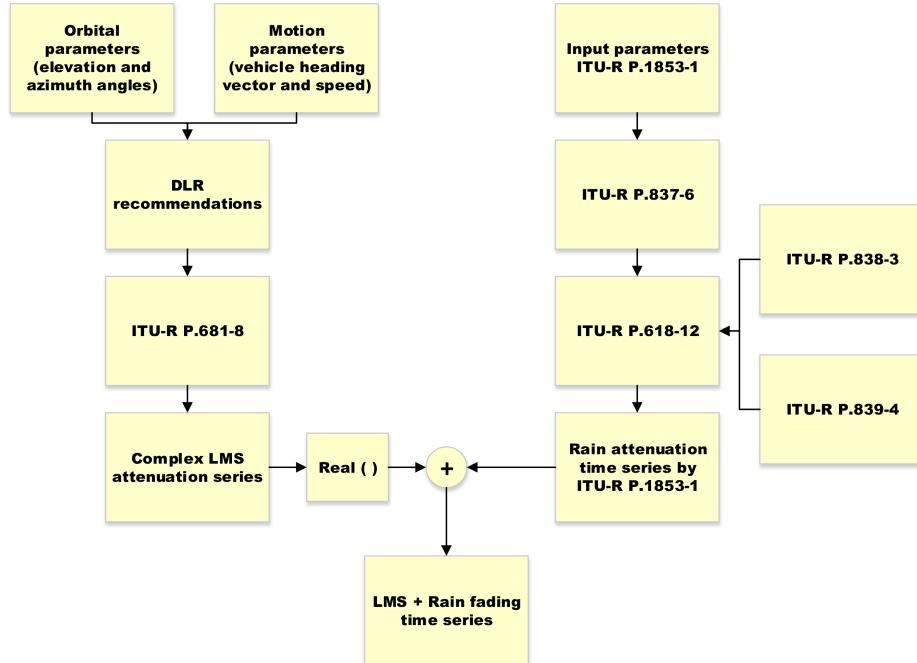


Figure 3.17: Block diagram of complete attenuation time series generator algorithm. Output time series is a combination of the individual rain fading time series and the LMS channel time series, generated individually.

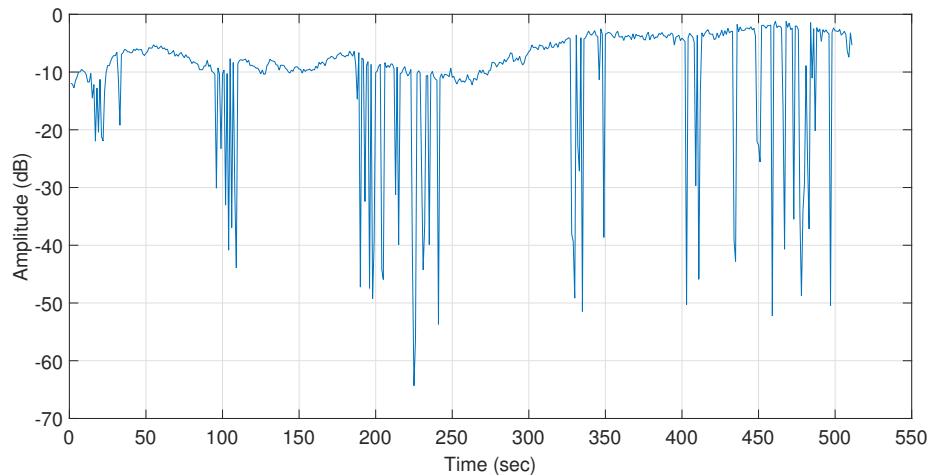


Figure 3.18: LMS channel time series during rain fading for a constant speed mobile ground receiver communicating with a GEO satellite at Ka-band.

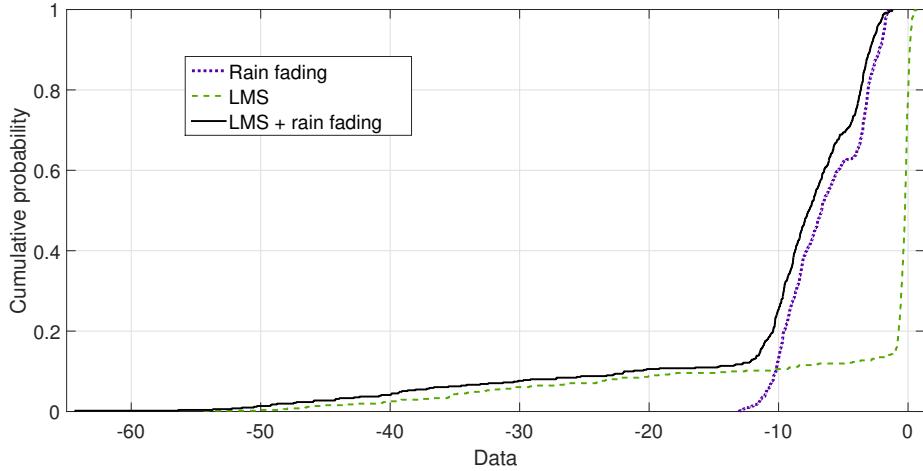


Figure 3.19: Addition of rain fading to the LMS amplitude. Note the change of the mean in the PDF. As expected, both events are independent.

material composition of the reflective surfaces, in this case the terrain, for mobile terminals. A communications system performance was analyzed while ionospheric scintillated signals from a GEO satellite experienced multipath on their way to an helicopter.

Furthermore, a 3D rain field synthesizer was described in details, with an emphasis on the acquisition of the rain attenuation time series, resultant from the contribution of the different rain rate segments, and how to compute them, was given for LEO orbits. This implementation is available at [75]. A novel method for generating more realistic rain attenuation time series for LEO orbits was proposed, available at [2] bundled with its equations and an example of a time series used in this work for the simulations.

A detailed description of the implementation of an LMS channel synthesizer being affected by rain attenuation for a downlink from a GEO satellite at Ka-band was provided. It is worth noting that the channel attenuation time series synthesized using the methods proposed and described in details by this chapter were used to test the algorithms proposed in Chapter 4.

Chapter 4

Kalman Filter and Interactive Multiple Model Filter for Satellite Communications

In this chapter, two KF applications are presented. Taking advantage of the channel attenuation time series generation methods covered in detail in Chapter 3, first the end-to-end satellite communication system simulator is described in detail, from the orbit generation up to the ACM algorithm within the receiver. Our main contribution relies on the addition of a Kalman filter at the receiver for signal-to-noise ratio (SNR) time-series predictions, which considerably decreased the amount of ACM switching without using any hysteresis for a satellite in LEO orbit. This achievement is important since it represents the potential for more robust systems, such as the DVB-S2X, to enable adaptations to be performed at the exact SNR level, resulting in an increase of throughput while meeting the BER requirements, *i.e.*, the QoS delivered can be improved with the proposed technique. First, the performance of KF predictions of SNR time series is described for rain attenuation condition while assisting the ACM at a downlink from a satellite in GEO orbit at Ka-band.¹ Next, QoS improvements achieved by the KF are presented for a downlink from a satellite in LEO

¹Published in the 2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP) [112]

orbit while being attenuated by rain at Ka-band.²

Furthermore, a more complex version of the KF is explored, which uses a bank of KF in parallel, called the IMM filter. This tool is proposed to be part of a receiver as well. It is demonstrated that its prediction advantages are two-fold: first, for networks suffering variable latency, it is possible to adjust the prediction interval with a smaller error than simply using a KF alone. For instance, recent reports such as Reference [113] have successfully gained a chance to demonstrate the impact of such variable network delays on satellite-based networks, even for the ones using LEO satellites. Second, another application of IMM filters is in assisting network relay mechanisms in selecting alternative routes during shadowing events. Our proposed method seems to be immune to the presence of rain at an LMS channel operating at a Ka-band downlink from a GEO satellite.^{3,4}

4.1 QoS Improvment in Satellite Communication Link Adaptations Using Kalman Filter

In order to assess the performance of a communications system operating in different scenarios, a communications system consisting of one transmitter, one receiver, and the channel was implemented in MATLAB [98]. Since the adaptations are performed on the downlink, the satellite acts as the transmitter while the GS is assumed to be the receiver that performs filtering and predictions on the received signals, as well as decisions upon adaptation schemes. Figure 4.1 shows the diagram blocks of enhancements required by the receiver-side of a communications system to perform predictions and adaptations. Note that the satellite channel dynamics, such as slant path loss and rain attenuation, are accounted for in the AWGN channel block for simulation purposes.

The system is designed such that the receiver continuously measures the signal attenuation, filters out sensor noise, and predicts its future value depending on the amount of delayed samples. Then, it decides whether modulation adaptation scheme is needed or not, and reports its decision back to the transmitter. An ideal feedback channel is assumed to

²Submitted to IEEE Transactions on Aerospace and Electronic Systems [76]

³In-print IEEE Access [78]

⁴Published in 34th AIAA International Communications Satellite Systems Conference [114]

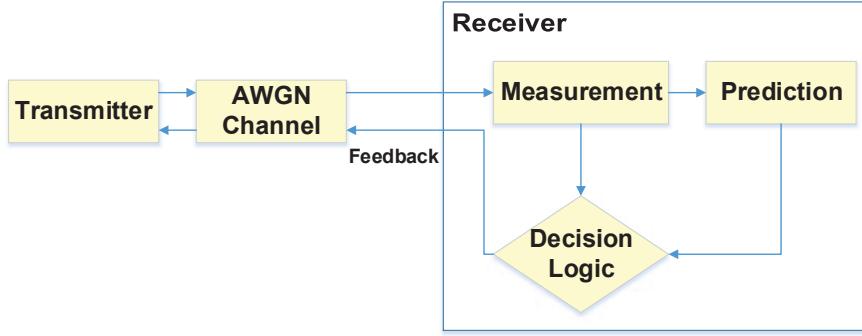


Figure 4.1: Block diagram of a communication system architecture showing the enhancements added to the receiver, such as the predictor and decision logic. Channel impairments are accounted for by the AWGN channel block.

be operating also at Ka-band. Although these actions will have an immediate impact on the higher-layer protocols, the analysis of such impacts are not the focus of this research.

For adaptation, the allowed modulation schemes consist of 4- and 16-QAM uncoded modulation schemes for GEO orbit simulations and 4-, 16-, and 64-QAM Hamming-encoded schemes at a rate of 4/7 for LEO orbit simulations. Perfect synchronism between transmitter and receiver is assumed. Considering a band-limited channel with constant noise power over the bandwidth, the channel is modeled as AWGN [57].

At the receiver input, the SNR is filtered and predicted k samples ahead, assuming k equal to the known number of delayed samples. Then, predicted values are converted into E_b/N_0 . Based on these predictions, the decision logic decides upon the need to reconfigure the radios according to the link performance requirements established by the network manager. Each of these receiver subsystems is described in detail on the following subsections.

4.1.1 Channel Adaptation Concept

Ideally, all the SDR parameters should be reconfigurable. Without loss of generality, this research work considers that only the modulation scheme can be changed. Note that it is also possible to reconfigure parameters such as encoding schemes, power levels, and waveforms. In this case, two frequency-independent downlink channels are considered. They

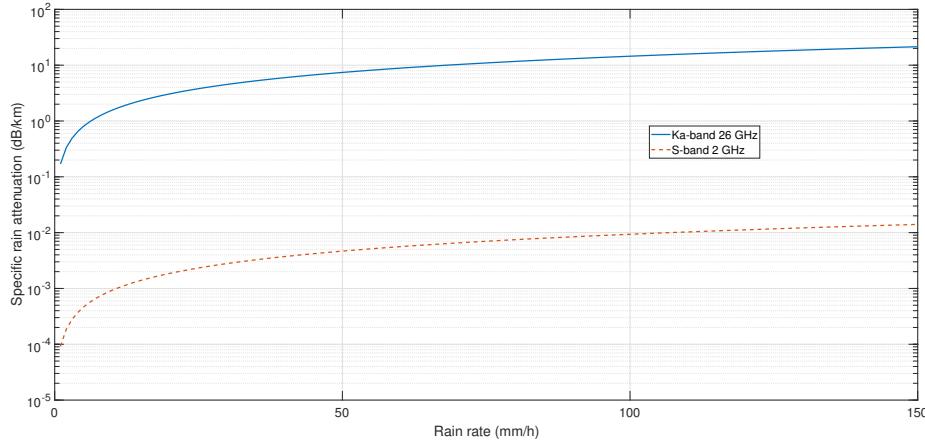


Figure 4.2: Theoretical specific rain attenuation for different rain rates at Ka-band and S-band for a carrier frequency at 26 GHz and 2 GHz, respectively, circularly polarized, based on the ITU-R P.838-3 [3]. For circular polarization the specific attenuation versus rain rate does not vary with the elevation angle.

are referred to as the *control channel* and the *data channel*. The data channel operates at the Ka-band and is responsible for the high data rate transmission, and it is where the adaptation takes place. The control channel is intended to operate at the S-band, where the rain fading is not as severe as in the Ka-band (see Figure 4.2). Its transmissions are primarily performed when a connection starts up for the first time, such as during a hand-shake protocol, as well as every time the Ka-band link enters an outage state due to deep attenuation or when the Ka-band transmitter is turned off.

Such an approach guarantees control over the satellite during the outage periods at Ka-band and allows battery power to be saved if the Ka-band transponder is turned off. While transmitting at S-band, the system might use a robust low-rate modulation scheme, such as DBPSK, for radio control purposes, as well as to probe the channel when the main Ka-band transmitter is deactivated. The communications performance analysis at S-band is outside the scope of this work and, similar to the feedback channel, it is assumed to be ideal. Instead, more focus is given to the adaptation performance at the Ka-band downlink data channel. As soon as the channel conditions achieve favorable link quality levels, the transmitter is instructed to start using the data channel and to turn off the control channel.

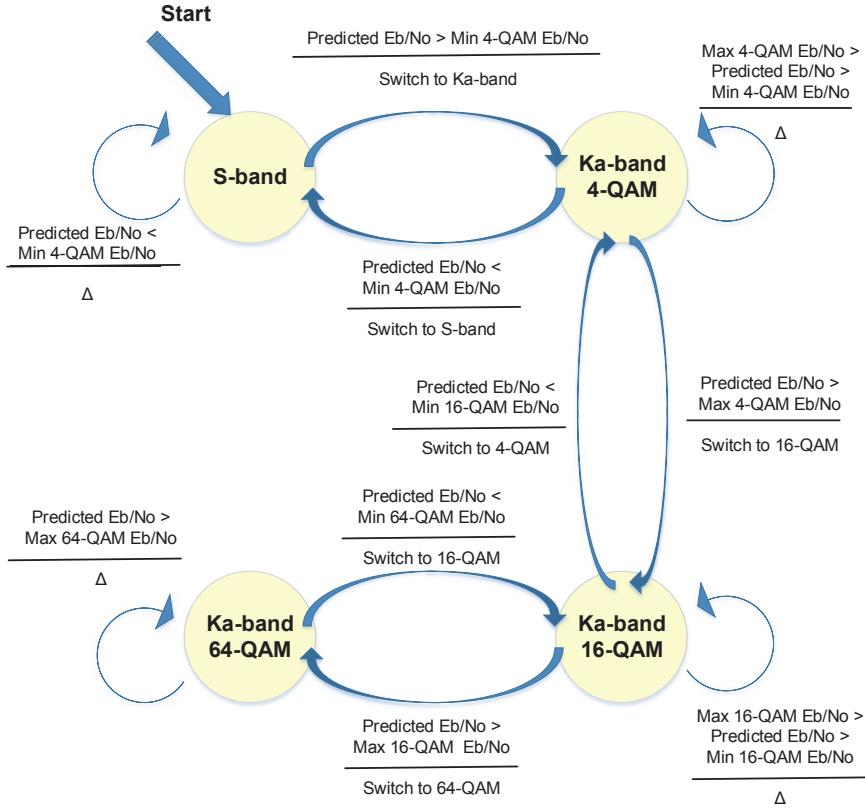


Figure 4.3: Finite-state machine of the link adaptation and frequency band control between S-band and Ka-band. Triggering events are ruled by the E_b/N_0 conditions shown in Table 4.1.

Figure 4.3 shows the finite-state machine illustrating this concept. Additional details about the modulation scheme adaptation and E_b/N_0 are given in the following subsections.

4.1.2 Prediction Using Kalman Filters

The predictor proposed for this work has a fundamental role of identifying the attenuation level a few samples ahead in time based on the noisy receiver measurements, thus allowing the transmitter to know what to reconfigure ahead of time. Although the round-trip delay for LEO satellites is usually only 1.4 ms, as opposed to 250 ms when dealing with GEO satellites [57], the additional delays induced by the processing equipment at the trans-

mitter and receiver should be considered, which is a major contributor to the processing delay of the packet buffering. Another delay source can be the protocol frame variable size, which is a function of the ACM scheme being used. The delay can potentially be increased if the link needs to be relayed over an additional link going through one or more satellites, causing the total delay to be on the order of a few seconds in duration [113]. Note that, in a scenario of full-duplex communications mode, some sort of attenuation prediction would be required to be embedded within each node such that both can reconfigure their radios at the correct time. In this research, just the downlink, *i.e.*, only the GS receiver, has the predictor implemented.

The main focus of this work is to improve QoS in adaptive satellite-based communications, *i.e.*, the ACM performance in terms of adaptation stability for applications that are sensitive to data rate. Regarding applications that require a stable data rate, adaptation systems that blindly change their modulation schemes, such as those that just rely on raw measurements, might end up degrading the end-user performance instead of improving the communications link. On the other hand, if hysteresis is used, the SNR threshold levels bringing the modulation scheme up or down are different, inhibiting the system to successfully use resources for a certain time period. Thus, predictions have to be made such that the decision logic does not inherit the measurement noise or fast fading oscillations around the ACM thresholds, which would cause quick adaptations in series, while being aware of the E_b/N_0 variations caused by the adaptation itself. The predictor output should be robust enough to recognize and to filter out noise in the boundaries of the adaptation levels between different modulation schemes, such that hysteresis approaches are not required, allowing the correct adaptations to be performed immediately after the signal crosses the theoretical E_b/N_0 threshold.

Usually, the measurements are contaminated by noise from several sources, such as the channel and both transmitter and receiver antennas and amplifiers, which result in noisy measurements at the predictor input. In order to tackle the problems of noisy measurements causing oscillations on the adaptations, while allowing for predictions to be done, the usage of a KF is proposed, and performance analyses are proposed for three different scenarios mentioned above: clear sky, rain fading using the 3D rain field, and rain fading using the

proposed LEO attenuation time series synthesizer in Chapter 3.

Therefore, a scheme is proposed to predict the attenuation level k -steps ahead using the well-known linear KF equations without control [23–25]. The most challenging part of using a KF is to model the system precisely. In the present approach, a simple mode, constant velocity, is used and all the model uncertainties are accounted for in the noise variance matrices. The search for these noise matrices is an open question regarding the system being modeled, and the interested reader can refer to [27, 29, 115, 116]. However, a solution to this noise variance matrices search problem is proposed in Section 4.2. In this work, the only measurement used is the SNR value at the receiver. However, during the filtering process, a second state, the SNR slope rate, which cannot be measured, is estimated and predicted as well. This is the reason why the matrices in the model have two dimensions (see Appendix A for detailed matrix descriptions).

The KF, fully explained in Reference [117], estimates a signal contaminated by two noise sources, the process noise $\omega(k)$ and the measurement noise $v(k)$, using the state space equations without control, namely:

$$X(k+1) = F \cdot X(k) + \omega(k), \quad (4.1)$$

which is responsible for modeling the state evolution with F being the transition matrix implementing the constant velocity model that allows for the projection of the current estimated state, and:

$$z(k) = H \cdot X(k) + v(k) \quad (4.2)$$

describes the noisy outputs of the measurement sensor. The observable state $z(k)$ is the measured received power amplitude, thus, measurement matrix $H = 1$. The process noise $\omega(k)$ and the measurement noise $v(k)$ are both zero-mean Gaussian-distributed with covariance matrices Q and O , respectively. O is known since it can be measured on the receiver, differently from Q , which is assumed known for the QoS analysis but requires a search method that has been proposed and described in Section 4.2.

The KF approach used in this work can be summarized by the following equations, which can be further divided into two sets of equations. The first set is responsible for

predictions of \hat{X}_j , given by:

$$\hat{X}_j(k+1 | k) = F \cdot \hat{X}_{j0}(k | k), \quad (4.3)$$

with the error covariance matrix Z_j given by:

$$Z_j(k+1 | k) = F \cdot Z_{j0}(k | k) \cdot F^T + Q. \quad (4.4)$$

The second set performs updates of the Kalman gain J_j :

$$J_j(k+1) = Z_j(k+1 | k) \cdot H_j^T \cdot (S_j(k+1))^{-1}, \quad (4.5)$$

where the residual covariance matrix S is computed by:

$$S_j(k+1) = H \cdot Z_j(k+1 | k) \cdot H_j^T + O_j. \quad (4.6)$$

\hat{X}_j and Z_j are updated using the expressions:

$$\hat{X}_j(k+1 | k+1) = \hat{X}_j(k+1 | k) + J_j(k+1) \cdot e_j(k+1), \quad (4.7)$$

$$\begin{aligned} Z_j(k+1 | k+1) = & Z_j(k+1 | k) \\ & - J_j(k+1) \cdot S_j(k+1) \cdot J_j^T(k+1), \end{aligned} \quad (4.8)$$

where the residual e is computed using the expression:

$$e_j(k+1) = z(k+1) - H \cdot \hat{X}_j(k+1 | k). \quad (4.9)$$

4.1.3 Decision Logic

The predictor outputs, *i.e.*, SNR values that are expected to affect the channel k -steps ahead, are used by the decision logic in E_b/N_0 format. If there is a need for adaptation of the modulation scheme or frequency band, in case of system outage at Ka-band, the receiver sends commands containing new parameters to the transmitter. Decisions are made considering the performance thresholds set by the network manager, left as designer parameters. The target BER is monitored through predictions of E_b/N_0 values. Thus, based on the maximum BER allowed, E_b/N_0 thresholds for the available modulation schemes are set. For instance, BER thresholds used in the simulations are detailed in Table 4.1. Without loss of generality, the maximum BER assumed was 10^{-3} .

Table 4.1: E_b/N_0 adaptation thresholds.

Mod. scheme	Freq. band	GEO	LEO
4-QAM	Ka-band	7.6	5.9
16-QAM	Ka-band	11.6	9.45
64-QAM	Ka-band	N/A	13.55

4.1.4 Complete Simulator Algorithm

To summarize, the overall flow graph of the simulator algorithm is illustrated by the block diagram in Figure 4.4. The left feedback channel represents the control channel at S-band, whereas the right feedback channel is the data channel for the Ka-band, and both are assumed to be ideal.

4.1.5 Simulator Setup

The satellite communication simulator is composed of a satellite orbit simulator, an attenuation profile synthesizer, the satellite communication link simulator, and the proposed improvements in the GS receiver, in which the Kalman filtering and adaptation decisions are performed. The attenuation simulator receives the satellite orbit profile from the orbit simulator and outputs the attenuation profile. For clear sky conditions, just the slant path loss is considered. Next, the communication link is simulated using the attenuation profile as the channel profile. At the end of each transmission, as mentioned above, the predicted E_b/N_0 is used to drive the adaptation for the next step. A general concept of the simulator's flow graph is depicted in Figure 4.5.

4.1.6 Simulation Results for GEO Orbit

Due to the lack of measurement data for analysis of rain fading at Ka-band for GEO satellites and based on fading slopes provided by measurements from the literature [14, 118–120], the rain attenuation behavior was simulated. Later, this same synthetic signal is considered as the measured E_b/N_0 at the receiver, which represents the rain attenua-

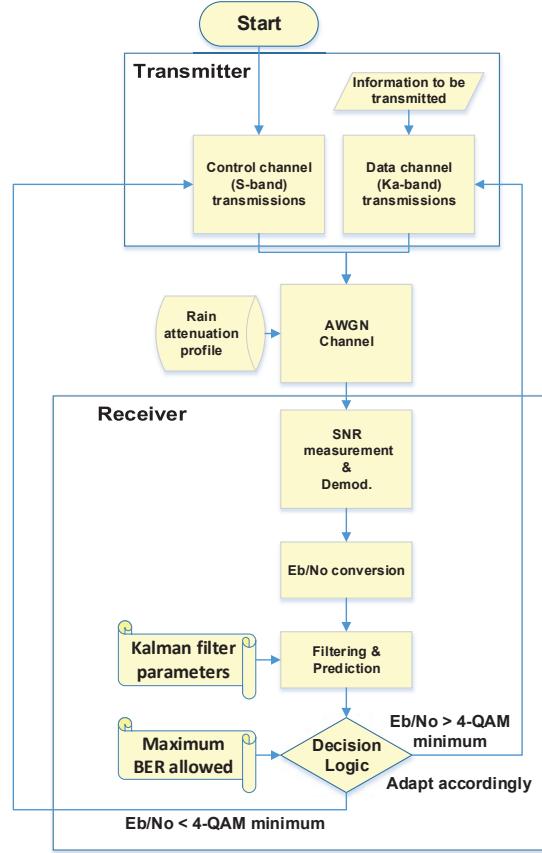


Figure 4.4: Simulator diagram block with the operational flow graph between the transmitter and receiver. The channel is driven by the resultant synthesized attenuation weather profile and slant path loss. Receiver contains additional blocks such as Kalman filter and decision logic.

tion behavior. This signal is 15 minutes long and the sampling frequency is 1 Hz. Even though there are ITU recommendations based on statistical models for GEO orbits, and based on the purpose of testing the proof-of-concept of the proposed prediction algorithm, this method was easier to implement. More complex ITU implementations for time series synthesizers and proposed equations for extending these synthesizers are provided in Section 4.2. Figure 4.6 shows the synthetic attenuation signal representing the E_b/N_0 measured by the receiver during rain and the predicted value plotted at the instant it was predicted to happen. Figure 4.7 shows a zoomed-in portion of Figure 4.6.

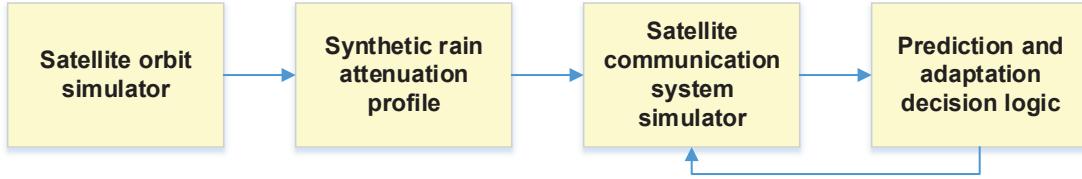


Figure 4.5: Complete flow graph of simulators used to analyze the Kalman filter prediction while driving the satellite communication adaptations. The rain attenuation profile can be either the 3D rain field generated or the proposed LEO rain synthesizer based on the ITU-R P.1853.

The number of symbols sent remained constant throughout all the simulations at 30,000 symbols/sec. Our design considers link loss every time the E_b/N_0 is lower than 7.6 dB, *i.e.*, transmissions made with the BER higher than allowed represent a zero throughput for the customer. The predictor values are initialized as follows (refer to Appendix A). The initial values of the state x , the state variances z_v and z_c , the process variance q , and the measurement noise O are unknown. It was chosen to be $x_0 = 0$. Since the attenuation value and the rate are uncorrelated $z_c = 0$ and $z_v = 10,000$ since it will be corrected with time. The value of q was varied from $q = 1$ (*e.g.*, assuming an inaccurate model) up to a low value as $q = 10^{-10}$ (*e.g.*, assuming a very accurate model). The noise was assumed to be $O = 1$. With the exception of the q values and the noise in the added portion Δx , the initial choice of these does not have a considerable effect in the overall system performance. The prediction window had the length of 10 samples and the prediction horizon considered 5 steps ahead, which represents 5 samples ahead, based on the sample rate used. In the simulations, one sample per second was used. The system's delay was set to be equal to the prediction horizon, *i.e.*, 5 seconds.

First, the system without the prediction and adaptive features was simulated during the emulated rain event, the same shown in Figure 4.10, and collected the BER values. Next, two additional scenarios were simulated: 1) both prediction and adaptive features on and 2) with the predictor off and the adaptive feature on. Table 4.2 summarizes the BER

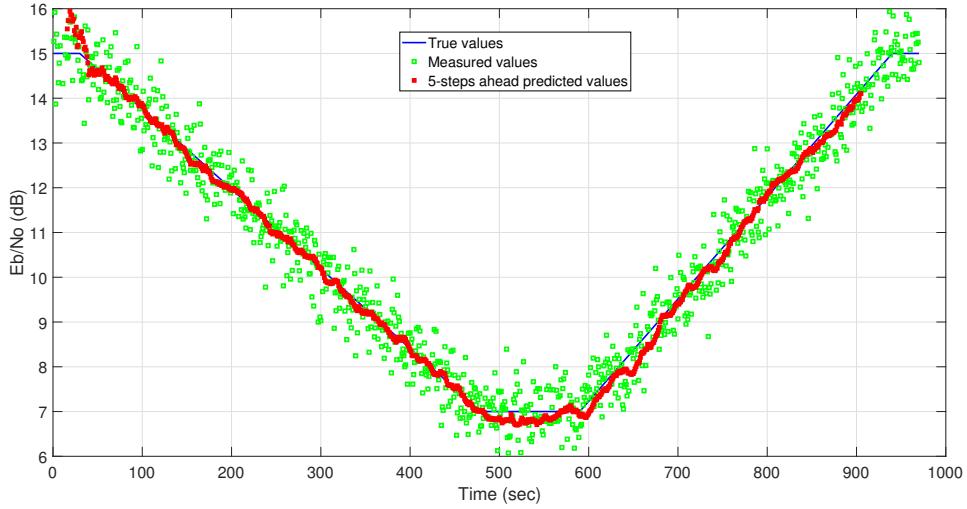


Figure 4.6: Rain attenuation predicted values k -steps ahead using noisy measurements. The true value cannot be seen due to the large number of samples

Table 4.2: BER for different simulated scenarios

Mod Scheme	Total bits	Error bits	BER
4-QAM (No adapt)	44,160,000	1,452	3.2880×10^{-5}
16-QAM (No adapt)	88,320,000	194,805	2.2×10^{-3}
Adapt on and Pred on	63,480,000	4,254	6.7013×10^{-5}
Adapt on and Pred off	88,320,000	194,805	2.2×10^{-3}

performance for these four scenarios.

According to Table 4.2, without adaptation and prediction, the BER for 16-QAM was higher than 4-QAM because before and after link outages the transmissions were made when the attenuation levels were high. However, the interesting part of these results is when the adaptation and prediction schemes were used, the BER performance was closer to that of only 4-QAM but an additional of 19M bits were transmitted. This is due to the fact that the predictor could “see” the increase in attenuation ahead of time and decreased the data rate by switching to a more robust modulation scheme. After the receiver reported

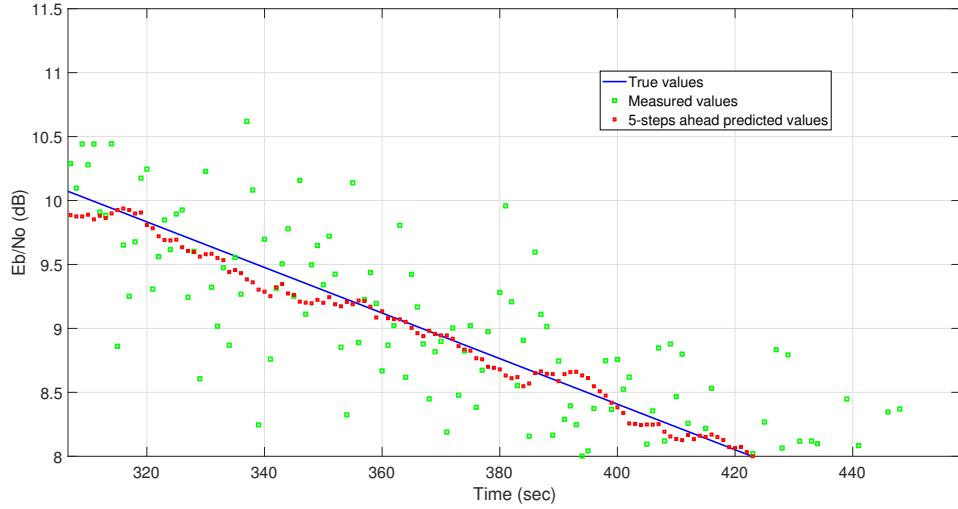


Figure 4.7: Zoomed-in section of Fig. 4.6. It can be seen that the predicted value is close to the true value to be measured k -steps in the future.

that the BER was above the required threshold, the transmitter started transmitting again according to the instructions received from the receiver and improved the data rate when the attenuation level allowed to do so.

Also, when using the adaptation without the prediction the BER was high. This is due to the fact that the modulation switching was being triggered at the wrong time instants leading to a decrease on the system's overall performance. Figure 4.8 shows the system performance for the third scenario, showing the change in the data rates, or goodput, for the customer over time. The instantaneous BER is also shown, where during the majority of transmission time the system tried to keep the BER below the threshold of 10^{-3} .

4.1.7 Simulation Results for LEO Orbit

The satellite communication simulator was implemented in MATLAB. Throughout this work, the area considered is a 100 km square centered on Atwater Kent Laboratories, located in Worcester, MA, USA. The attenuation due to free-space loss for a LEO ISS orbit is shown in Figure 4.9. This profile, assumed to be a clear sky condition, is normalized and inverted in order to be used as the received power amplitude due to the satellite motion.

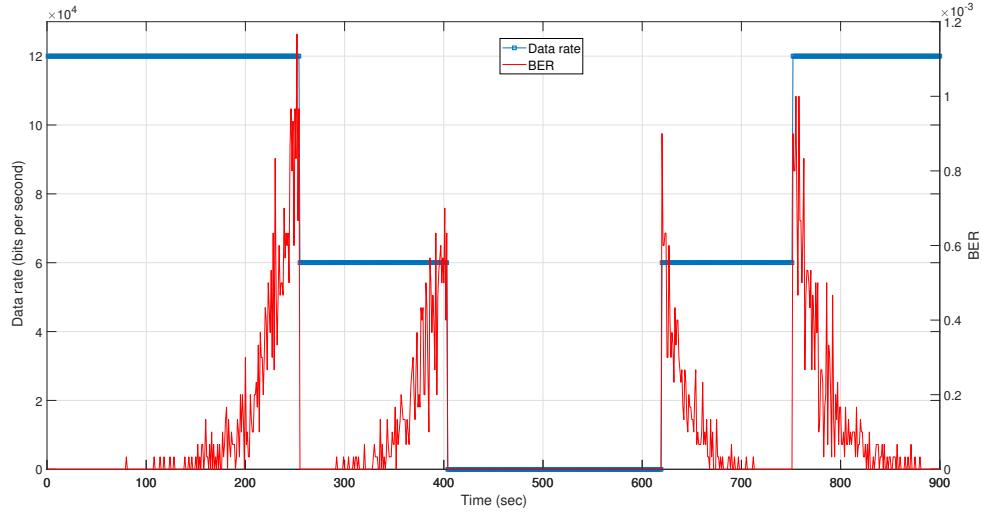


Figure 4.8: Received data rate changes according to the predicted E_b/N_0 based on the maximum allowed BER.

The transmitted signal is then affected by the attenuation profile (which is different for each of the three scenarios considered in this work) plus the channel noise. At the receiver, the attenuation profile corrupted by channel noise is corrupted by sensor noise with Gaussian distribution and variance O . The original attenuation profile was offset by 10 dB above zero dB such that the entire available range of E_b/N_0 thresholds could be achieved. The normalized offset version is shown in Figure 4.11 as the dotted curve.

As mentioned in Section 3.2, for the 3D rain field attenuation synthesizer, at each time instant t the total contribution of each segment with different rain rate is accounted for, up to the LOS intersection point with the rain height plane. In this work, the LOS range for each segment $\Delta l_i(t)$ was assumed to be 1 km long. This resulted in an “staircase”-like attenuation profile, computed by Eqs. (3.15) and (3.16). $A_{\text{rain}}(t)$ is shown by the marked line in Figure 4.10. In order to make it more natural, since rain attenuation does not have a discretized behavior, it was smoothed by applying a moving average with LOESS robust regression. This smoothed version is also shown in Figure 4.10. The resultant smoothed 3D rain field-based profile was subtracted from the solid line in Figure 4.9, resulting in the final rain attenuation profile, which is shown as a dotted line in Figure 4.11, and then used

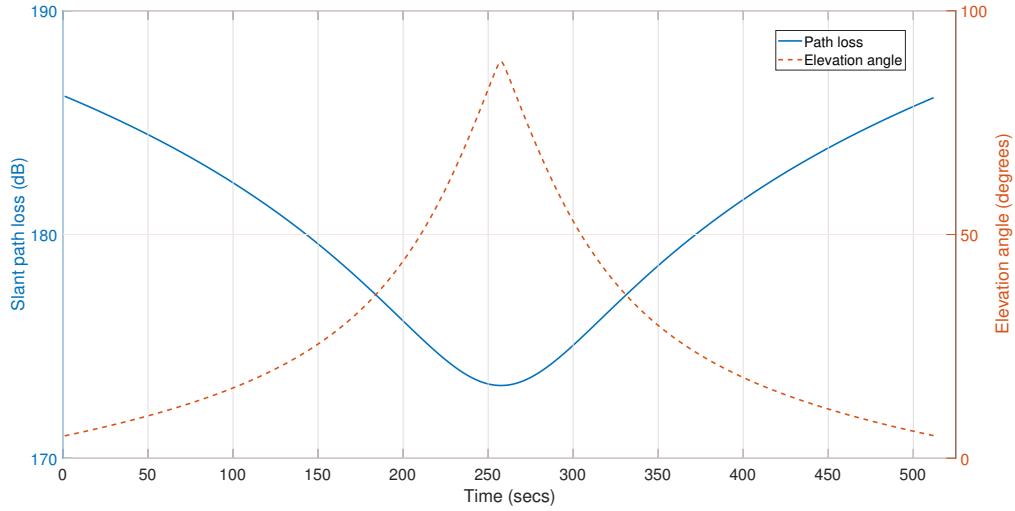


Figure 4.9: Slant path loss power from a signal being received at GS from a LEO satellite in the ISS orbit, during one of the passes over Worcester, MA. Also the elevation angle from an antenna at the GS shows the minimum attenuation during the maximum elevation angle.

by the simulator for the 3D rain field scenario.

Regarding the attenuation profile provided by the proposed ITU-based method, the resultant LEO attenuation profile shown in Figure 3.13 was subtracted from the normalized offset slant path loss shown in dashed line by Figure 4.11. This yields another total attenuation profile for the rain fading scenario, which is shown as a solid line in Figure 4.11.

For the communication system simulator implementation, the synthetic attenuation profiles were fed into the AWGN channel as the SNR values, such that both fading and noise effects could be achieved at once. For all modulation schemes used, the average transmitted power was kept constant, such that the channel effect was driven purely by the SNR profile.

A frame payload of 180,000 bits/sec was transmitted. The predictor values were initialized in the following order: The true initial values of the state x_0 , the state variances p_v and p_c , the process variance q , and the measurement noise O are unknown. Thus, it was assumed $p_c = 0$, $p_v = 1,000$, and $x_0 = 0$, since they will be computed and corrected over time. The process noise variable q is assumed constant during the simulations and equal to

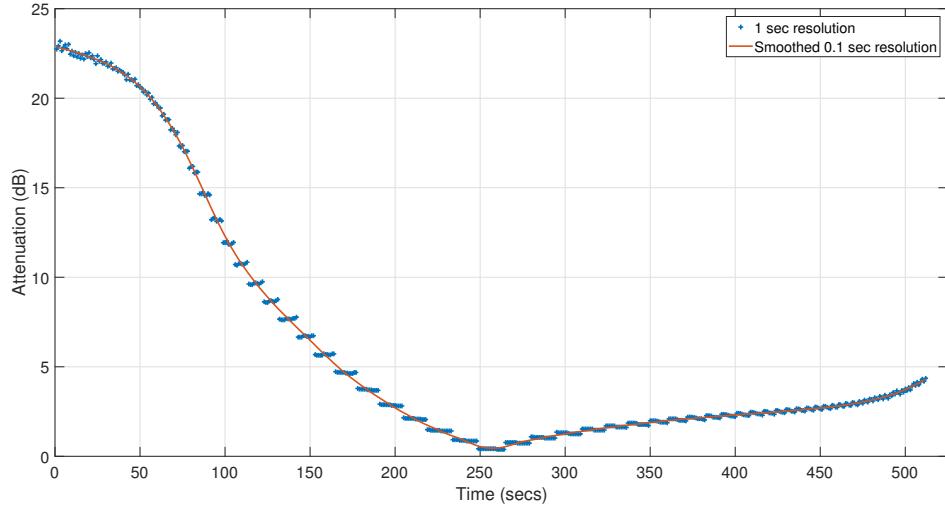


Figure 4.10: Resultant rain attenuation computed using 3D rain field synthesizer. Marked values were computed using Eqs. (3.15) and (3.16). Due to the segmentation of path distances the output is not smooth. Smoothed version was achieved by applying moving average using LOESS robust regression (solid line).

10^{-4} . Simulations were run for measurement noise variances of $O = 1$ and $O = 0.5$ [23].

In terms of delay, it was assumed there was a delay of 5 and 10 samples. Thus, E_b/N_0 was predicted for 6 and 11 steps ahead, which is when the output of the decision logic would be used based on the last delayed measurement. All combinations of scenarios and parameters (24 in total) were simulated 100 times and the mean of results, such as the BER after the demodulator, the BER of decoded data, and the number of transitions between modulation schemes, were collected and presented in Tables 4.3 and 4.4 for clear sky and rain fading scenarios. The number of modulation scheme transitions is shown in Figure 4.12. For the ideal case of no noise present either in the channel or at the measurement sensor, as well as no delay, the total number of modulation transitions expected for a LEO orbit with clear sky channel, considering the three available modulation schemes would be four, the BER after demodulation was 2.74×10^{-4} , and BER after decoding was 1.57×10^{-6} .

Based on the numerical results shown in Tables 4.3 and 4.4, for the large majority of scenarios simulated, the mean BER values are almost the same for the cases with and

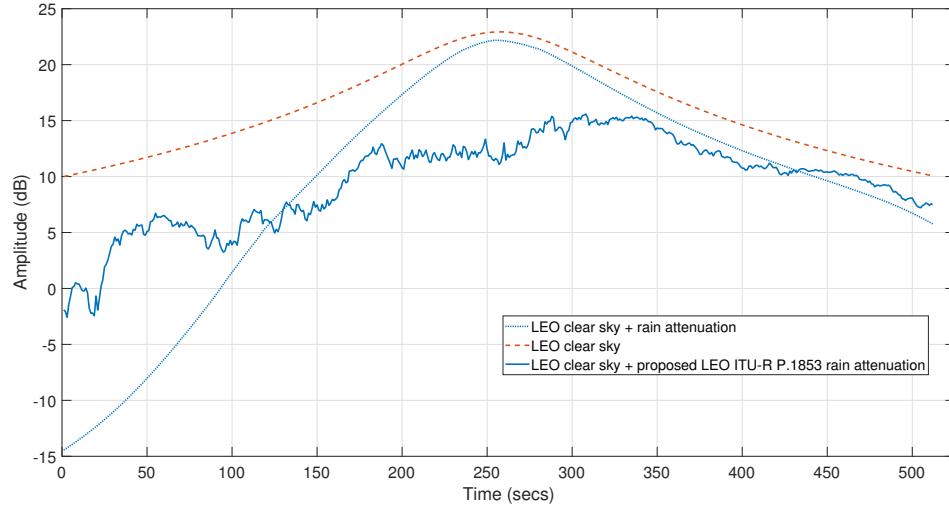


Figure 4.11: Normalized power amplitude at receiver, for clear sky condition, offset by 10 dB (dashed line). Received amplitude profiles after subtraction of attenuation shown in Figures 4.10 and 3.13.

Table 4.3: Communication system average performance for 2 different delays during clear sky; 100 runs

Clear sky						
		$O=0.5$		$O=1$		
Delay (samples)		ON	OFF	ON	OFF	
5	BER	$8.6 \cdot 10^{-4}$	$6.04 \cdot 10^{-4}$	$1.11 \cdot 10^{-2}$	$4.12 \cdot 10^{-4}$	
	BER uncoded	$4.31 \cdot 10^{-4}$	$2.3 \cdot 10^{-4}$	$5.81 \cdot 10^{-4}$	$7.66 \cdot 10^{-6}$	
10	BER	$9.97 \cdot 10^{-4}$	$3.5 \cdot 10^{-4}$	$1.1 \cdot 10^{-3}$	$4.53 \cdot 10^{-4}$	
	BER uncoded	$5.06 \cdot 10^{-4}$	$4.14 \cdot 10^{-6}$	$5.31 \cdot 10^{-4}$	$1.02 \cdot 10^{-5}$	

without the KF. This allows link performance analysis of improvements achieved when adding KF to receivers operating in such a wide range of different scenarios.

Table 4.4: Communication system average performance for 2 different delays during rain fading; 100 runs

Delay (samples)	Rain 3D field				Rain proposed ITU LEO			
	$O=0.5$		$O=1$		$O=0.5$		$O=1$	
	ON	OFF	ON	OFF	ON	OFF	ON	OFF
5	BER	0.0172	0.0172	0.0173	0.0173	0.0678	0.0678	0.0679
	BER uncoded	0.0173	0.0173	0.0242	0.0173	0.0679	0.0678	0.0679
10	BER	0.0173	0.0173	0.0242	0.0173	0.0679	0.0678	0.0697
	BER uncoded	0.0003	$1.9 \cdot 10^{-5}$	0.0095	$5.4 \cdot 10^{-5}$	0.0001	0.0001	0.0073

Regarding the QoS performance analysis for systems using adaptation, the number of modulation scheme transitions was recorded. This number alone does not convey QoS information since it only represents the number of times the system had to react to a varying E_b/N_0 measured and/or predicted due to relative motion between transmitter and receiver, and/or other channel impairments such as weather-induced ones. It was noticed that there was a big difference in the number of transitions in the modulation schemes for when the KF was used as opposed to when it was not.

When there was no KF, the number of transitions ranged from 3.5 times (delay = 5, $O = 1$, rain fading based on proposed synthesizer) up to 9 times (delay = 10, $O = 0.5$, rain fading based on proposed synthesizer) more than systems with the KF included, while keeping the average overall performance in terms of throughput or BER. For scenarios when the measurement noise variance was $O = 1$ the system experienced many more transitions than for the scenarios with $O = 0.5$, independent of the delay amount, the presence of the KF at the receiver kept the transition numbers down, close to the ones in the scenarios with $O = 0.5$.

It is worth noting that the improvement, in terms of reduction in the amount of adaptation transition, was similarly observed for the four combinations of delay and sensor noise variance, for all three different channel fading profiles as shown in Figure 4.12. This reinforces the potential for the applicability of more advanced ACM techniques that may take advantage of the KF, even though knowledge of the system model was abstracted to the noise covariance matrices, yielding adequate QoS performance without precise system modeling.

This evident reduction in the amount of adaptation transitions for the same BER performance due to the use of KF in receivers dealing with both clear sky and rain fading conditions, while experiencing different measurement noise variances, in a LEO satellite downlink at Ka-band, is believed to be novel to the best of the authors' knowledge.

Other observations involving the link simulations include: there was no difference between the BER computed after the demodulator and the BER after decoding, for a delay of 5 samples, with the exception of the case $O = 1$ for clear sky. These increased due to the increase in the measurement sensor noise. Interestingly, for all the cases of rain fading, the

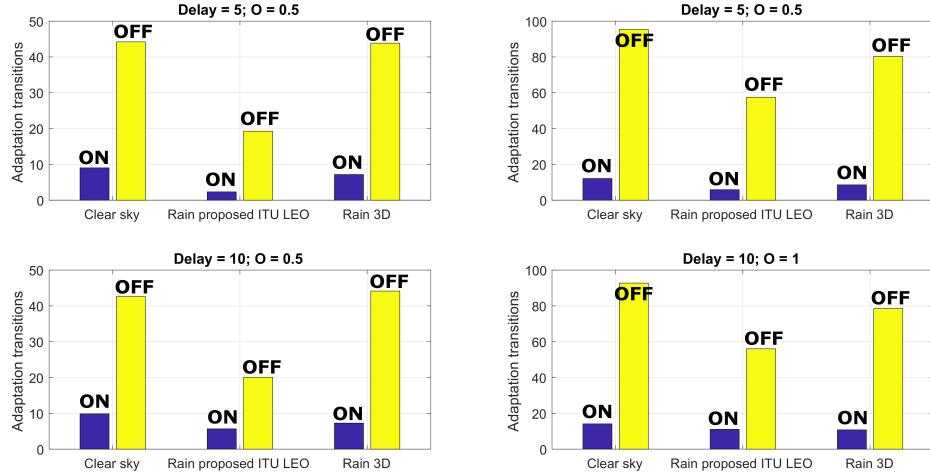


Figure 4.12: Comparison of modulation scheme transitions between a receiver with the KF predictions k-steps ahead ON and OFF, for three scenarios: 1) clear sky; 2) rain fading generated by ITU-based proposed method for LEO; 3) rain fading generated by 3D rain field.

effect of encoding was evident, which resulted in a much smaller BER after decoding than the BER computed right after the demodulator, as expected.

4.1.8 Conclusion

Using a simple model KF, SNR time-series predictions were shown to improve the BER performance of a receiver while using ACM. In addition, the KF also improved the QoS of a satellite receiver through a channel experiencing rain fading at Ka-band on a downlink from a LEO satellite. It was found that using KF to drive link adaptations improved the adaptation stability by more than nine times when compared with a system without a KF, for three different scenarios including the one using the proposed rain fading synthesizer for LEO orbits, while minimizing the impact of measurements with different noise variance values.

Our results show that, from the BER point of view, the effects of adding and removing the KF were similar, independent of the scenario, and the method of rain attenuation synthesis. Also, the pattern observed in the number of adaptation transitions, shown in

Figure 4.12, among the three different channel profiles, is due to the fading level, characteristic of the method and statistics used to synthesize each attenuation profile. Even though a performance comparison between these three channels is not recommended, they do demonstrate the QoS benefits in terms of adaptation performance achieved in systems using a KF when dealing with variable delayed measurements from noisy sensors with different variance levels in a downlink from a LEO satellite operating at Ka-band.

This might be of interest to future high throughput communications systems researchers and designers that need to achieve an overall data throughput target, and a maximum BER target, while improving the QoS for services that provide applications sensitive to data rate through different channel conditions. This work showed that these QoS improvements could be achieved in all three different scenarios and, most interesting, for two simulated rain fading scenarios based on two different sets of ITU recommendations.

4.2 Shadowing Detection Using Interactive Multiple Model Filter

An IMM filter was proposed to operate in a dynamic changing channel, such as a downlink from a satellite in LEO orbit while experiencing rain at Ka-band. Its main application is on assisting node relay selection mechanisms during signal shadowing for mobile terminals experiencing LMS channels such as those described in Section 3.4.

In order to perform shadowing detection in the presence of rain, a signal state classification is proposed to be done by a filter that is capable of state estimation in the presence of noisy measurements such as the KF. In a KF, Eq. (4.1) is responsible for modeling the state evolution and Eq. (4.2) describes the noisy outputs of the measurement sensor. In this case, the transition matrix F in Eq. (4.1) is expressed as

$$F = [1 \ dt], \quad (4.10)$$

which is a time-invariant constant-velocity model that allows for the projection of the current estimated state, in this case the measured power k -steps ahead, controlled by dt assumed to be equal to 1. The observable state $z(k)$ is the measured received power amplitude,

thus, $H = 1$. The process noise $\omega(k)$ and the measurement noise $v(k)$ are both zero-mean Gaussian-distributed with covariance matrices Q and O , respectively.

The first step in setting up a KF is to have an accurate system model that describes the majority of different system behaviors. For instance, several specific behaviors can be modeled when tracking a target position for navigation purposes, such as the speed and acceleration of an aircraft [26–29]. In these cases, the model fully describes the system dynamics, with few uncertainties left to the noise functions $\omega(k)$ and $v(k)$.

For the atmospheric environment model, the goal was to include as many local weather variables in the model as possible, such as the influence of the current tropospheric temperature, wind speed, amount of cloud over the area, and all the time evolutionary behaviors for each of these variables. However, such modeling is known to be very complex, thus developing both a state space model and a transition matrix that accounts for the effects of all these variables and their evolution over time is considered to be outside the scope of this work. Nevertheless, based on the assumption that these unknowns have the same Gaussian distribution as the noise [28, 37, 38, 121], the proposed approach uses a very simple state space model that accounts for the uncertainties in the process noise $\omega(k)$, which ends up estimating parameters such as covariance matrices, described in the following subsection. The KF approach used in this work is summarized by Eqs. (4.3)–(4.9).

Since in this case the conditions for an optimal KF are not satisfied (if known, a realistic model of atmospheric impairments affecting the communications channel would be non-linear), an extended version known as an IMM filter has been proposed in order to perform recursive estimation when continuous uncertainties are Gaussian-distributed, such as AWGN, and discrete uncertainties such as finite system states are assumed [28, 37, 38, 121]. The IMM filter is composed of a finite number of KFs, and each filter is designed to represent a different system behavior, or state.

Each KF input \hat{X}_{j0} is computed by [121]:

$$\hat{X}_{j0}(k \mid k) = \sum_{i=1}^N \hat{X}_i(k \mid k) \mu_{i,j}(k), \quad (4.11)$$

for $i, j = 1, 2, \dots, N$ used to identify the KFs within the IMM filter (the subscript $j0$ refers

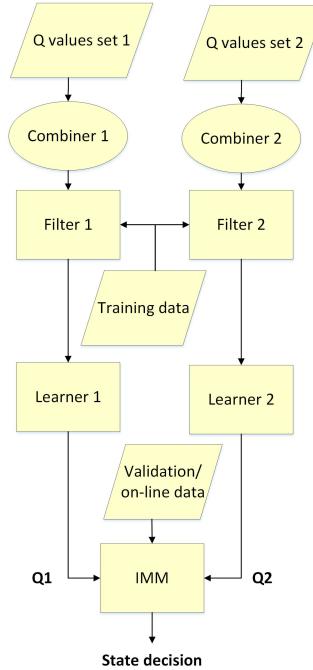


Figure 4.13: Diagram block for the proposed design for search of Q matrix integrated with the IMM filter design. Each matrix combination is tested individually. Q_1 and Q_2 are the output matrices chosen by the learning blocks. The ultimate IMM output is the state decision if there is shadowing or not.

to IMM mixed values only). The mixing probabilities $\mu_{i,j}$ at instant k are computed by:

$$\mu_{i,j}(k) = \frac{\pi(i, j)\mu_i(k)}{\hat{\mu}_j(k+1 | k)}, \quad (4.12)$$

and mixes the original KF inputs \hat{X}_i with each other. The predicted model probability $\hat{\mu}_j$ from instant k to $k+1$ is computed by:

$$\hat{\mu}_j(k+1 | k) = \sum_{i=1}^N \pi(i, j)\mu_i(k), \quad (4.13)$$

where $\pi(i, j)$ is the Markovian state transition probability matrix [38].

The next step is to use the standard KF equations in each parallel filter, with its respective mixed inputs computed by Eqs. (4.11) and (4.14). Each KF in parallel has the same update and prediction equations presented in Eqs. (??)–(??), with the only difference being their respective Q matrices, which in this case are computed automatically, and the

covariance matrices Z_{j0} computed by:

$$\begin{aligned} Z_{j0}(k \mid k) = & \sum_{i=1}^N \left[Z_i(k \mid k) + (\hat{x}_{j0}(k \mid k) \right. \\ & \left. - \hat{x}_i(k \mid k)) \cdot (\hat{x}_{j0}(k \mid k) - \hat{x}_i(k \mid k))^T \right] \mu_{ij}(k). \end{aligned} \quad (4.14)$$

Finally, the individual KF estimates are combined into one IMM filter estimate using the expression:

$$\hat{X}(k+1 \mid k+1) = \sum_{j=1}^N \hat{X}_j(k+1 \mid k+1) \mu_j(k+1), \quad (4.15)$$

where the model probabilities μ_j are updated by using the expression:

$$\mu_j(k+1) = \frac{\hat{\mu}_j(k+1 \mid k) \cdot LL_j(k+1)}{\sum_{j=1}^N \hat{\mu}_j(k+1 \mid k) LL_j(k+1)}. \quad (4.16)$$

The likelihood function LL_j is given by:

$$LL_j(k+1) = \frac{\exp \left[-\frac{1}{2} e_j^T(k+1) \cdot S_j^{-1} \cdot e_j(k+1) \right]}{(|2\pi S_j(k+1)|)}, \quad (4.17)$$

for $j = 1, \dots, N$. The process repeats itself for the next iteration, when a new set of measurements is acquired from the sensors.

In order to initiate the proposed algorithm the following assumptions are made: $X(0)$ is equal to the received power level under clear sky conditions; $Z = [10000]$ since the covariances are expected to converge to a constant value; initial state probabilities $\mu = [0.5; 0.5]$; and the Markovian state transition probabilities are assumed to be:

$$\pi = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}. \quad (4.18)$$

In this case, π was chosen to represent the behavior of the system staying at a certain state with a higher probability than the probability of transitioning to another state. This behavior was considered based on the slow-fading characteristic of the rain attenuation effect and on the deep-fading characteristic of shadowing in LMS channels. In the following subsection, a more detailed explanation regarding the choice of the filter number present on the IMM filter is provided in the following subsection.

During online operations, after choosing the Q matrices, at each iteration one SNR measurement is input and two output probability values, computed by Eq. (4.16), from each KF inside the IMM filter, are compared against a threshold. Figure 4.13 presents the state decision output for a threshold value equal to 0.5.

4.2.1 Defining Number of Filters in IMM

When designing an IMM filter, one of the key requirements is to define how many KF should be used. Each KF in an IMM filter is often used to model one system state [28, 29, 121], and the IMM filter computes probabilities to most accurately identify which model reflects the system current behavior. Thus, one must know beforehand which system behaviors will be tracked by the IMM filter, such that a KF can be designed for each of them. Two filters model the detection of shadowing in LMS channels, being able to distinguish deep fading from slow fading, such as due to rain attenuation. The primary idea here is to have one filter that quickly follows the noisy SNR measurements steep transitions caused by shadowing, and the other filter to follow the slower transitions caused by rain fading.

4.2.2 Automatic Search for Q

In the previous section, the problem of modeling the system was left to defining values for the O and Q matrices, which represent the measurement and noise process noise variances based on the assumption that these noise sources are Gaussian. Since O is an independently generated value of the system relative to the measurements, O is assumed to account for only the noise resulting from the sensor. Thus, O is assumed to be known, and can be measured offline using a known input.

Considering the primary goal of allowing the system to estimate measurements from noisy inputs, the cost function to be minimized is the mean-squared error (MSE). In order to achieve this goal, an approach that tests and builds the different Q matrices for each parallel KF is proposed as shown in Figure 4.13, which illustrates their diagram blocks.

The algorithm is initiated with different sets of possible variance values for each parallel KF. In order to tailor each parallel filter to a different behavior, or mode, one needs to provide different ranges of values for each set, through which the search algorithm will look

for and build the Q matrices. One of the goals is to demonstrate and analyze the IMM performance using a simple model that distinguishes between attenuation slopes, with slower changes being a characteristic of clear sky or rain conditions, and very steep slopes being a signature of shadowing events. Thus, only two different modes, *i.e.*, two different Q matrices, are necessary: (i) A mode with small values indicating slower attenuation slopes, and (ii) A mode with large values indicating steeper attenuation slopes. This idea is standard for IMM filter designers, in which there is one model and two filters representing the system state being in that model or not. In the literature [29, 36, 121], values used to distinguish states in various model approximations, such as aircraft tracking, are two-to-three orders of magnitude apart. Further details are described in Section 4.2.3.

Given that both filter models run in parallel within the IMM, the output of each iteration is a combination of both operations depending on their respective probabilities to distinguish between no fading and/or slow fading, and deep fading modes. As a result, the technical challenge associated with this operation is identifying the correct mode, or at least the most probable one at each instant.

While designing these filters, it was noted that large values in a Q matrix make the IMM filter rely more on its noisy inputs, thus its output follows the noisy inputs more closely. Furthermore, this choice of Q causes the output to quickly change its values in an attempt to follow the rapidly changing input, such as when noise or deep fading is present. On the other hand, for small values of Q the output of the IMM slowly follows the changes in the inputs. Thus, the result is a smoothed output with respect to its input. Each KF attempts to track the signal within the noise, thus when the IMM mixes their output probabilities, it makes easier to distinguish deep fading from noise and slow fading, allowing for the detection of shadowing events even in the presence of rain.

It is worth noting that, when building Q matrices by combining the values available in a set, for the Kalman filtering equations to be computed properly one needs to make sure that Q be symmetric and all of its eigenvalues must be positive, *i.e.*, Q be positive definite since it needs to be invertible. In the following section, the proposed IMM approach performance is discussed for each of the scenarios mentioned in the previous section.

For each set of values, Q matrices are constructed by evaluating all possible combinations

of these values within the Q matrix. The process occurs in parallel and iteratively for all KF until all the possible Q matrices are evaluated. In Figure 4.13, this parallel process is depicted by the “Combiner 1” and “Combiner 2” blocks.

The outputs of each “Combiner” are used by the individual KF using only Eqs. (??)–(??). The filters are operated using a training dataset and the MSE of the outputs from each individual filter is recorded. Then, each individual “Learner” independently decides which Q matrix met a certain requirement, in this case, the minimum MSE. The result of each “Learner” is then fed into the main IMM, for validation and online estimations. Comments about its performance evaluation are provided in Section 4.2.3.

This proposed methodology allows the system to update its Q matrices in order to operate under changing environments for achieving a better performance. The determination of how frequently Q should be re-evaluated is currently under investigation by the authors, and is outside the scope of this work.

4.2.3 Simulation Results

In this section, the proposed IMM filter performance is analyzed for five different channel conditions: (i) fixed receiver under rain, (ii) LMS rural under clear sky, (iii) LMS rural under rain, (iv) LMS suburban under clear sky, and (v) LMS suburban under rain. The input to the IMM filter is the time series of synthesized noisy SNR measurements at the receiver. For simulation purposes, the noise measurement process is assumed to be Gaussian with zero mean and standard deviation O with values equal to 0.1, 0.5, and 1. The search for the Q matrices Q_1 and Q_2 was performed building matrices using all combinations of values from two independent sets. The set for Filter 1 is $[0, 10^{-10}, 10^{-3}]$, and for Filter 2 is $[10^{-10}, 10^{-1}, 1, 10]$. The careful reader might note that included in the ranges are 0 and 10^{-10} , which are very close to each other; however, results show that there is a difference in performance when choosing one over the other. Note that the sets used contain different number of values, since there is no requirement regarding the maximum number of elements of each set. However, an increase in the number of elements increases the total number of combinations that must be tested during training of the KF. The IMM proposed approach was tested using the MSE of the SNR estimations, the same metric used for the automatic

search of Q matrices during training.

Considering the time series length, the IMM filter performance was evaluated using three different training dataset sizes, with the largest one being half of the time series duration. After the Q_1 and Q_2 matrices were chosen, the testing dataset was used to validate the performance of the IMM filter. Table 4.5 shows the validation MSE statistics for 100 simulation runs for three different O values for a scenario with a fixed terminal experiencing rain fading only, and for the three different training dataset sizes. Despite the smallest mean error values for the measurement standard deviation $O = 1$, the mean error for $O = 0.1$ and $O = 0.5$ can be considered similar, with $O = 0.1$ presenting the smallest standard deviation error values, as expected. Also, for all different values of O , the mean error linearly decreased with the increase of the training dataset size, as expected. Since O is a parameter directly related to the hardware being used, the discussion of better approaches to improve the IMM performance function of O values is out of the scope of this work. Thus, $O = 0.1$ assumed for the remaining simulations in this chapter. For the five scenarios mentioned above, the Q values for the IMM inner Filters 1 and 2, were computed, respectively, as:

$$Q_1 = \begin{bmatrix} 10^{-3} & 0 \\ 0 & 10^{-3} \end{bmatrix}, \quad (4.19)$$

$$Q_2 = \begin{bmatrix} 10 & 10^{-10} \\ 10^{-10} & 10^{-10} \end{bmatrix}. \quad (4.20)$$

For Q_1 , the algorithm preferred the value of 10^{-10} over 0 for the variance value of the rate. This shows that a very small change of values had an impact on the decision of the final Q matrix, indicating the level of sensibility that the automatic Q search method provides.

Tracking the inner filter probabilities values μ indicates which filter has a higher probability of having a model that best describes the current environment behavior, defined as the channel states. Using a threshold of 0.5 the IMM inner Filters 1 and 2 represent the states 0 and 1, no-shadowing and shadowing states, respectively. For instance, during the deep-fading events, IMM filter outputs higher than the threshold probability values for Filter 2, leads to the detection of State 1 indicating shadowing is being experienced by the

Table 4.5: IMM filter MSE of rain faded channel – 100 runs.

$O = 0.1$			
Training dataset size	50 samples	150 samples	250 samples
Mean	33.99	26.66	17.22
Std. dev	0.9	0.79	0.56
$O = 0.5$			
Training dataset size	50 samples	150 samples	250 samples
Mean	31.87	25.95	19.41
Std. dev	1.95	1.78	1.58
$O = 1$			
Training dataset size	50 samples	150 samples	250 samples
Mean	10.02	8.05	5.85
Std. dev	1.64	1.39	1.22

signal. The complete detector is described by Algorithm 1. Figure 4.14 illustrates a snippet of the state decision time series output together with the measured SNR at the receiver. Several shadowing events were detected at different attenuation levels, in this case solely due to rain fading. Notice that the detection is not ideal and a few samples caused false alarm detection. It is worth noting that the dominance of one of the filters occurs based upon the current channel shadowing conditions and how each filter was designed, independently of the fading. This is implied within the order of magnitude difference among the values available for the initial set that each Q matrix is constructed from.

In order to assess the performance of the proposed shadowing detector through different scenarios and to compare the proposed approach with some common state-of-the-art detection methods, simulations were conducted for five different scenarios, described above. Each scenario was sampled at two different rates, 1-Hz and 10-Hz, and divided into 100

Algorithm 1 Proposed shadowing detector operational routine

Require: Initial parameters

```

1: Define training window size  $n$ 
2: Select number of states to be detected  $s$ 
3: Choose  $s$  sets of  $Q$ -values of any size within different ranges each
4: Define  $s - 1$  thresholds values for state detection
5: Build all possible  $c_s$  combinations of matrices for each of the  $s$  sets
6: Choose optimization metric
7: for  $t=1$ :end of operation do
8:   if  $t > n$  then
9:     Receive new measurement  $z$ 
10:    for  $k = 1 : 1 : s$  do
11:      for  $j = 1 : 1 : c_s$  do
12:        Train individual Kalman filters
13:        Learn  $Q$ -matrices for optimized chosen metric
14:        Run IMM filter
15:        Get detected state value
16:      end for
17:    end for
18:   else
19:     Update training window
20:   end if
21: end for

```

different slices of 512 seconds duration each, resulting in more than 14 hours of synthesized attenuation time series, containing only rain, shadowing, or a combination of these two impairment sources. For each scenario, 5 detection methods were compared: SNR threshold, mean SNR, and the proposed IMM with three different training dataset sizes (30, 150, and 250 samples).

Since the shadowing events were simulated, the ideal detection time series are known

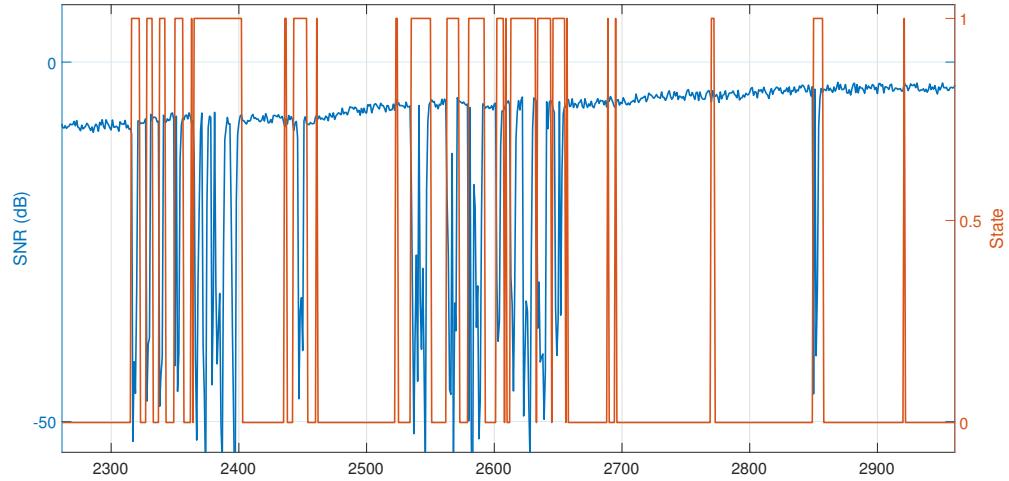


Figure 4.14: IMMM outputs of state detection for mobile terminal using a Ka-band link with a GEO during rain fading through an suburban LMS channel. The measured amplitude values and the is shown for comparison. Shadowing detection is achieved at different levels of attenuation due to rain fading. Samples recorded at 10 Hz.

beforehand and were used as a baseline for comparison against the detector outputs. Statistical shadowing detection performance was assessed on three major categories, over the time slices: mean time duration of wrong detection, standard deviation of wrong detection, and percentage of total time of wrong detection. Figure 4.15 shows the detection performance in a scenario with a fixed terminal experiencing only rain fading, without shadowing, allowing evaluation of false detection performance. Both threshold SNR and mean SNR detection methods showed poor performance with more than 60% of the time making false detection at a sample rate of 10 Hz. However, the IMM filter showed a very good performance with close to zero false detection. These results motivated the aforementioned research to be extended to several different LMS channels, under different atmospheric conditions.

Regarding the sample rate, a comparison between 1-Hz and 10-Hz results is presented in order to show to the reader the impact of choosing a different sample rate value to track a certain channel phenomena, as well as to illustrate its potential to be used within similar scenarios but with different purposes of state tracking. The remaining performance

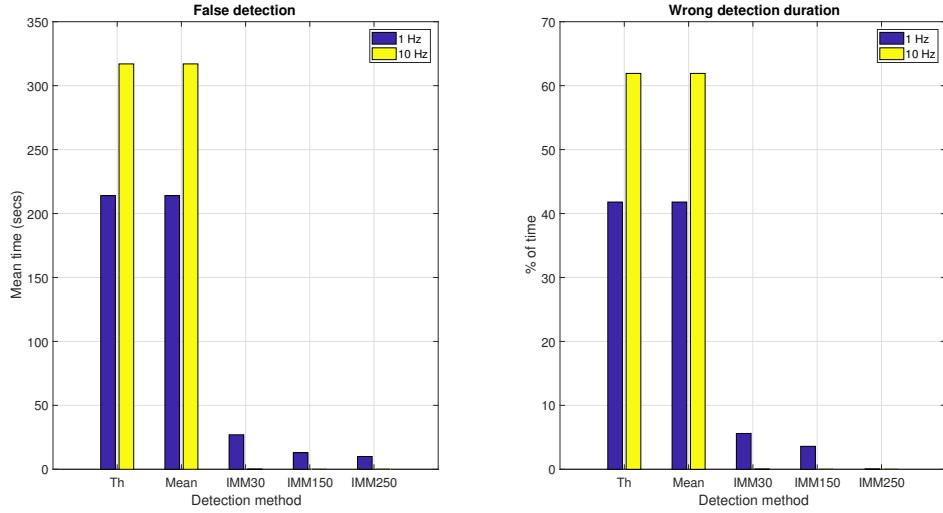


Figure 4.15: Shadowing detection performance in scenario with rain fading, and no shadowing, for a fixed terminal using a Ka-band link with a GEO satellite. Lower false detection mean time duration is better. Both threshold SNR and mean SNR methods show poor performance.

evaluation discussion considers only the 10-Hz sample rate, since it is the most appropriate to capture the shadowing effects based on the minimum time-series sample rate for a mobile terminal, as mentioned in Reference [109].

On suburban scenarios, shadowing detection was evaluated under clear sky and rain conditions. The results, illustrated in Figure 4.16, show that the IMM filter behavior in terms of mean time duration of detection error and total percentage of time doing wrong detection is similar and immune to the presence of rain. However, the performance of the other methods, SNR threshold and mean SNR, will vary substantially depending on the channel atmospheric conditions. These possess a relative good performance during clear sky conditions, even somewhat better than the proposed IMM filter, but an extremely poor performance during rain with a difference of more than 32% in this case for the percentage of total time duration making wrong decisions.

Simulations were performed for rural scenarios as well, for both clear sky and rain conditions. The results shown in Figure 4.16 also confirm that the proposed IMM filter

method has a better performance for shadowing detection during rain on LMS channels in rural scenarios, where a minimum of 8.5% improvement in percentage of time duration of wrong detection was achieved by the IMM filter over the SNR threshold method. As in the suburban scenario, the IMM filter performance during clear sky and rain conditions was similar. The SNR threshold and mean SNR showed a better performance during clear sky conditions with very low mean time duration of detection error and percentage of time duration.

4.2.4 Conclusions

Using synthesized attenuation time series for five different GEO satellite channels experiencing clear sky and rain conditions, simulation results presented the advantages of using the IMM filter for shadowing detection for fixed and mobile terminals during rain conditions. Preliminary results exposed a major flaw in current state-of-the-art shadowing detection methods while operating during rain, presenting very poor performance in terms of false alarm for shadowing detection when no shadowing event was actually present. However, these methods still have better detection performance than the proposed IMM filter when there is no slow attenuation caused by rain.

Considerable detection performance improvements could be achieved by the proposed IMM filter approach when comparing it against current state-of-the-art methods. This is the case for a mobile terminal using an LMS channel experiencing rain fading in a GEO satellite link operating at Ka-band.

Future high throughput mobile satellite communication systems using GEO satellites at Ka-band will be able to rely on better adaptive designs based on detailed and more accurate channel state information, as those provided by filter-based detectors such as the proposed IMM method. These systems will be aware of the current communication channel conditions while dealing with channel uncertainties, such as precise space and atmospheric weather models. In addition, multiple states can be configured in order to represent specific channel conditions, with the potential to improve the performance of adaptive radios and increase the overall network cooperative aspect.

4.3 Chapter Summary

This chapter provided prediction algorithms for satellite-based receivers. Simulation results showed that a simple model KF is capable of improving the QoS of receivers through incredible decrease of the number of adaptation switches during clear sky and rain conditions at Ka-band. Also, with the proposed model-free approach of the IMM filter, shadowing can be more accurately detected during rain for mobile terminals, which has the potential to be used in future network node relay selection mechanisms in cognitive networks.

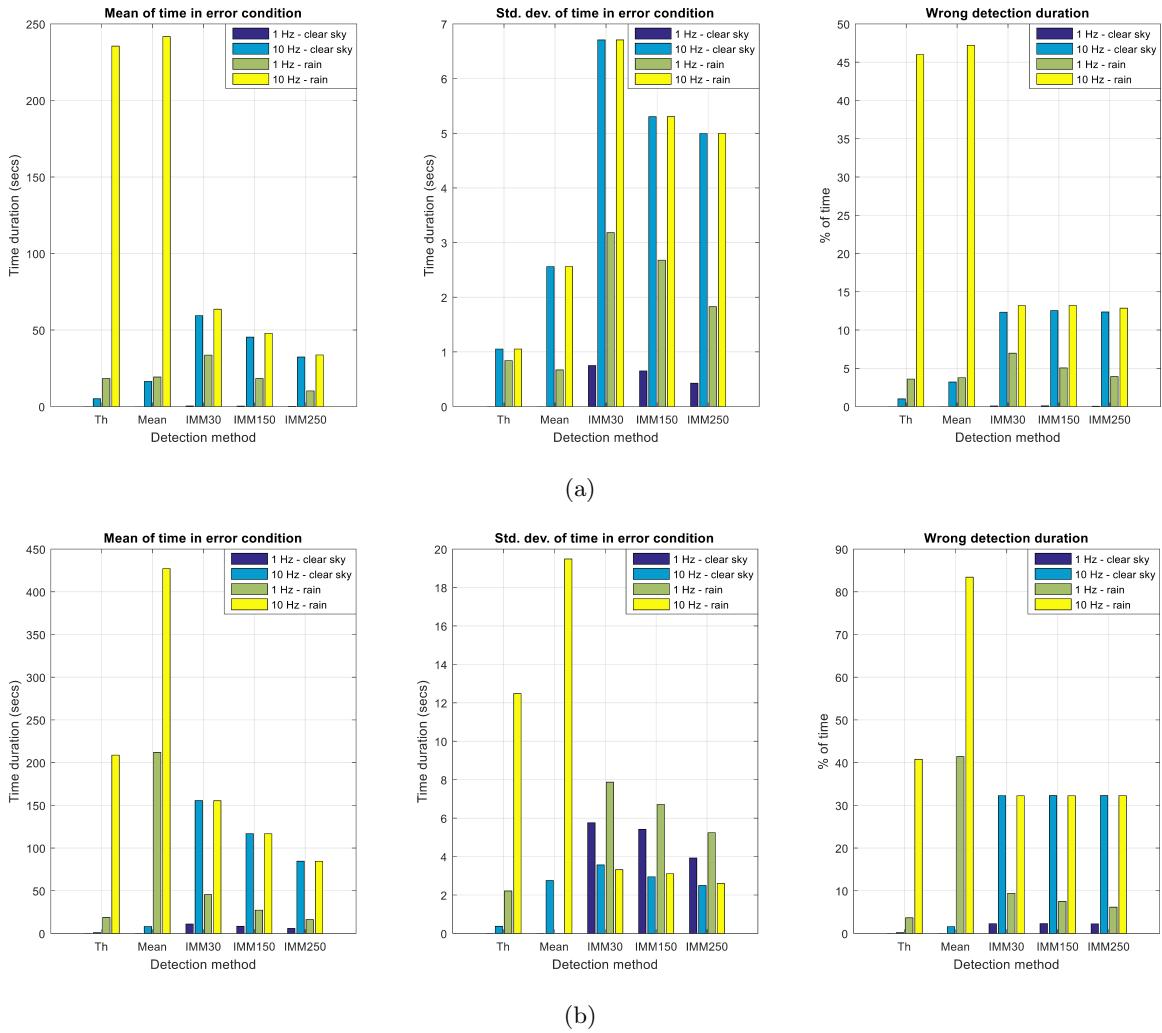


Figure 4.16: Shadowing detection performance on suburban (a) and rural (b) LMS channels during clear sky and rain conditions. Lower time duration means better performance. Shadowing detection on LMS channels by IMM filter is invariant to atmospheric conditions when sampled at 10 Hz, and better performance was achieved during rain by spending less time in erroneous shadowing detection condition when compared to current state-of-the-art detectors.

Chapter 5

Cognitive-Radio Engine Architecture for Autonomous Space Communications

This chapter describes the cognitive radio (CR) principles and core architectures^{1,2,3} required to enable learning and decision-making on SDRs for future autonomous space-based communications systems. In Section 5.1, the cognitive engine (CE) architecture proposed to NASA John H. Glenn Research Center (GRC) Space Communications and Navigation (SCaN) Testbed project is described. Next, in Section 5.2 a new comprehension about the Reinforcement Learning role in adaptive wireless communications is provided with a focus on satellite-based communications, which forms the groundwork for the description of the autonomous MAC protocol containing the physical (PHY) layer parameter scheduler. This scheduler leverages a novel hybrid machine learning algorithm, described in Section 5.3 as the primary component of the proposed cognitive engine. Numerical simulations and results analysis are provided within the end of each subsection. Concluding remarks close the discussion in Section 5.4.

¹Published at the 34th AIAA International Communications Satellite Systems Conference, October 2016 [4]

²Submitted to IEEE Cognitive Communications for Aerospace Applications Workshop [122]

³In preparation [123]

5.1 Intelligent MAC Protocol for SDR-based Satellite Communications

In 2012, NASA GRC installed the SCaN Testbed [124–126] on-board the ISS and continues to operate it from a mission control center at GRC. The SCaN Testbed is an experimental communication system comprised of three different SDRs [127, 128], and is the first space-based testbed of its kind available to selected researchers to propose implementation solutions to address issues related to SDR-based communications systems to and from space. The radios operate at S-band and Ka-band with NASA’s satellite relay infrastructure, *i.e.*, Tracking and Data Relay Satellite System (TDRSS), and S-band direct-to-ground stations on Earth. The communication links to and from the SCaN Testbed provide real-world satellite dynamics between user spacecraft and relay satellites and user spacecraft directly with ground stations. These dynamics include time-varying Doppler changes, thermal variations, differences in waveform characteristics, real-time interference, significant range variation, ionospheric effects and scintillation, and other propagation impairments.

The radios onboard the testbed are **flight-grade systems**, fully compliant with NASA’s SDR architecture, the Space Telecommunications Radio System (STRS) [129, 130]. The STRS Architecture provides abstraction interfaces between radio software and proprietary hardware (general-purpose processors and field-programmable gate array processors), allowing third-party software waveforms and applications such as the CR system software to interact and run on the radio. STRS also makes a library of waveforms available to developers to provide various modulation, coding, framing, and data rate options (the so-called radio “knobs”) available to the decision-making algorithms.

Solutions such as adaptive communications based on cognitive decision making needs to be researched, not only to solve communications issues on Earth, but also to allow the development of systems that will enable space exploration in the near future [131]. SDRs will become a **larger part of the communications infrastructure** as exploration continues over the next decade. These flexible and reconfigurable systems are generally more complex than traditional fixed-operation radios and CR systems offer a solution to reduce the complexity and risk associated with these new systems. The work presented in this chapter is part of

this effort, providing CR algorithm research and development for SDR systems in space.

5.1.1 Applications of CE for NASA⁴

Applying cognitive applications to space poses a number of challenges due to the **risk-adverse** culture of space missions coupled with the **resource-constrained** environment of space platforms. These two driving conditions will heavily influence where and when CR systems are used within mission spacecraft and throughout the communication system.

Three areas have emerged as candidate application areas for CR systems. The different areas are **node-to-node** communications, **system-wide** intelligence, and intelligent internet-working. The first application entails the radio-to-radio link between mission spacecraft and ground terminal (either through relay or direct-to-ground). Cognitive decision making may improve (increase) throughput across a communication link by consuming otherwise unused designed link margin or mitigating impairments. Algorithms that sense performance and understand the entire link capacity could adjust waveform settings to maximize user data and symbol rate by minimizing coding or other overhead. Taking advantage of significant range changes during a ground station pass or operating at reduced data rates at low elevation angles (normally outside the traditional link design) or through weather events offers additional opportunities for additional science data return. Signal recognition among nodes may alleviate missed opportunities due to configuration errors or mitigate unexpected interference.

The second application of CR systems is **system-wide intelligence** where CR systems make operational decisions normally performed by operators or data-intensive aspects not currently done. For example, CR systems could be applied to **relay and ground station scheduling, asset utilization** (proper asset loading and accommodating mission priority), **optimum link configuration and access times**, infrastructure fault monitoring, and failure prediction, among others. Many of the applications will help reduce operation oversight (and cost) and help reduce operational complexity due to the large number of possible configurations. Large data analysis opens a new area to discover performance and operational benefits from all aspects of data collected including: link performance, platform environ-

⁴Provided by NASA GRC SCaN Testbed engineers Dale J. Mortensen and Richard Reinhart.

ment (*e.g.*, radiation, thermal, and mechanical/vibration), asset availability, and system performance.

Finally, as communications infrastructure becomes more network-based using commercial and international standard protocols, CR systems may benefit **the control and data functions** of the communications network. Optimizing data throughput according to QoS metrics such as bit error rate, loss packet rate, routing decisions, store-and-forward protocols, and publish-and-subscribe techniques may benefit from cognitive control. Allowing algorithms to learn network behavior, especially small networks with repeatable data flows, may yield throughput and reliability benefits.

One notable aspect regarding CR systems for space is the need for **verification or ground testing of all operational conditions** before launch. To minimize risk on orbit, missions generally test each mode of operation prior to flight. This helps provide confidence in the on-orbit operation. Having CR systems make unplanned and potentially unpredictable changes to the flight systems on-orbit will take considerable research and technology demonstrations such as those described in this dissertation.

5.1.2 Project Overview

NASA GRC, in Cleveland, Ohio, accepted a proposal entitled “Intelligent MAC protocol for SDR-based satellite communications” with the goal to develop a cognitive engine algorithm for future space-based communications systems, with the unique opportunity of experimenting with SCaN Testbed SDR’s.

The R&D team is composed of members from WPI and The Pennsylvania State University, in collaboration with the SCaN team at NASA GRC. Commenced in November 2014, the project is expected to be completed in the May 2017 time frame.

During this collaboration, the team has designed and developed a proof-of-concept of an **intelligent MAC protocol** to maximize data link performance and to improve the robustness of space communications systems, by using SDRs for low margin data links operating on dynamically changing channels, especially at low elevation angles. This protocol’s core is made up of a CE that must be capable of dealing with conflicting multi-objective issues during radio-resource allocation. Adaptation of radio parameters might be required while

mitigating channel impairments according to the current channel conditions and/or when meeting new performance requests. Key capabilities to perform cognition are prediction and learning techniques, assisted by third-party databases whenever these are available.

This CE project is aligned with NASA's **Communication and Navigation Systems Roadmap Technology Area 5** [132], which focuses on cognitive radios in space that sense their environment, autonomously determine when there is a problem, attempt to fix it, and learn as they operate. This experiment is also one step in the direction of **removing communication as a constraint for future missions** and their critical phases, and emergency communications to enable safe and efficient human exploration and autonomous robotic space explorations. The CE algorithm, as well as simulation and future space-based experiment results will be used to assess the adaptive MAC protocol performance for the on-orbit SDRs, which will help in the design of future MAC protocols and mitigation techniques that can make use of the radio flexibility in terms of intelligent radio parameter reconfiguration.

5.1.3 Scenario Description

There are four different communication channels between the **fixed ground stations** at GRC/White Sands, the moving **ScaN Testbed SDR on board the ISS**, and the **TDRSS satellite at a GEO orbit**, as illustrated by Figure 5.1. Radio parameter reconfiguration might be done during periods of signal fading, especially those happening during low elevation angles of the SCaN Testbed antenna, while tracking the TDRS satellite, or low elevation angles from a GRC ground station antenna, while tracking the ScaN Testbed.

A higher level overview of the proposed CE diagram blocks is shown in Figure 5.2. This basic architecture consists of a **gathering observation data**, *i.e.*, telemetry data reported from the transmitter or measured at the receiver. A **predictor** uses past information to **predict radio parameters** values and builds learning in an efficient way in order to keep under control storage memory growth. The decision logic receives the predictions and, based on the link performance requirements and resource availability, decides upon the need for adaptation. Future applications and additional sources of information might include third-party databases providing atmospheric/space weather data.

During radio adaptations, the CE is expected **learn the environment behavior** by building

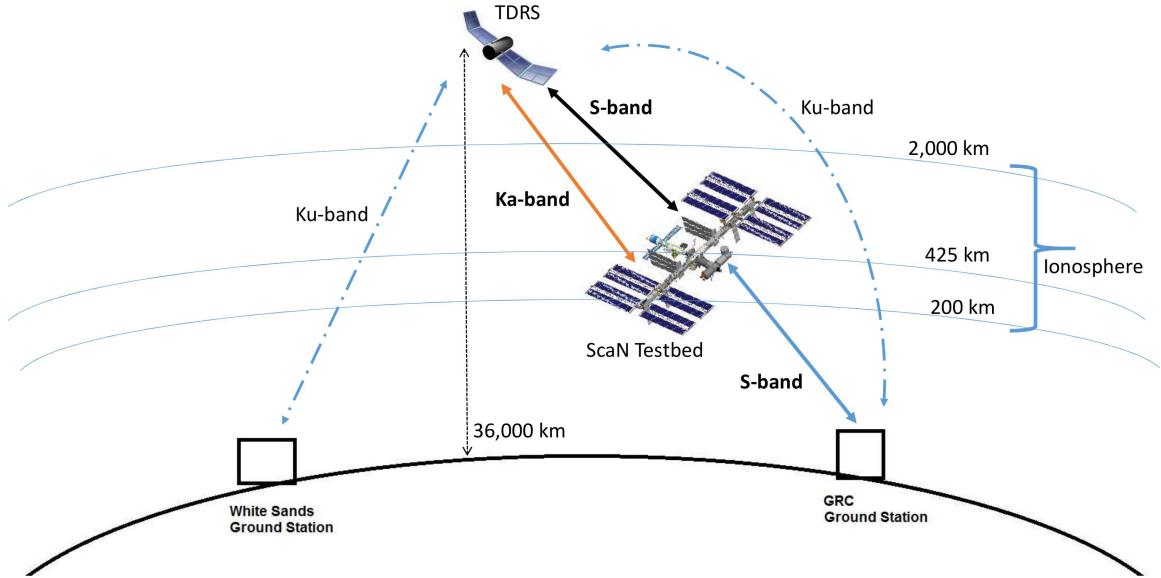


Figure 5.1: Examples of scenarios where the proposed Intelligent MAC protocol could be tested. They consist of links using the Near-Earth Network in S-band and links using the Space Network at both S- and Ka-bands.

a model that maps observations into actions, *i.e.*, radio parameter sets, while the channel dynamically changes while at a certain channel state. Figure 5.3 illustrates the proposed CE architecture diagram block at a lower level perspective from that shown by Figure 5.2.

An initial proof-of-concept ML algorithm implementation of the proposed design shown in Figure 5.2 is a modified version of the classic Reinforcement Learning (RL) algorithm, described in Section 5.2.1. It is specially designed to deal with multi-objective resource allocation in space communications systems. A more complex and autonomous version of this algorithm is proposed in Section 5.3.

5.2 Multi-Objective Reinforcement Learning (MORL) for Cognitive Radio-based Satellite Communications

In this section, a basic CE design based on reinforcement learning (RL) is described specifically to enable a satellite-based CR to learn, reason, and make decisions over multiple available resources and multiple goals based on its past experiences. Regarding a satellite

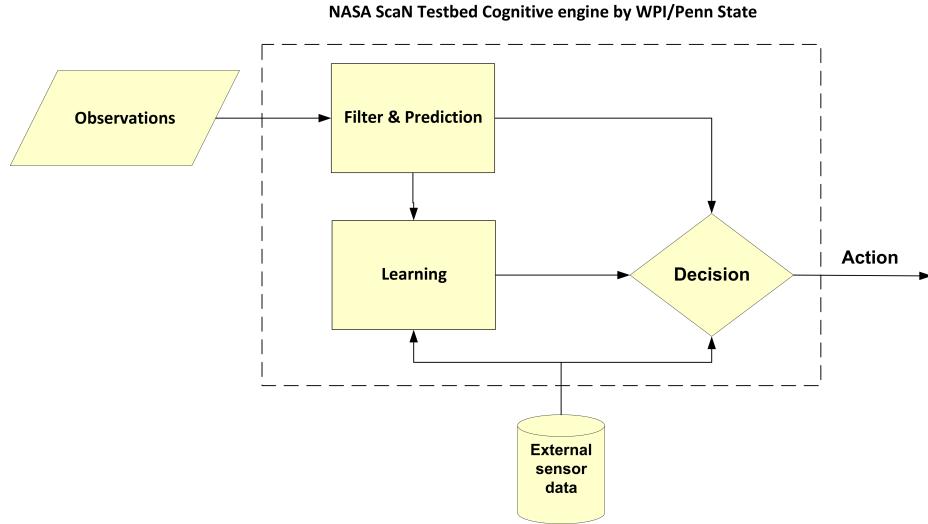


Figure 5.2: High level block diagram of proposed cognitive engine. Inputs are environment measurements and communications system performance metrics. Outputs are actions composed by radio parameters. The learning algorithm reasons over the inputs in order to propose outputs that may or may not be accepted by the decision-making block.

communications channel, some additional factors play a significant role that usually are not considered in terrestrial links including orbital dynamics, such as spacecraft trajectory, velocity, and antenna elevation angle profile, space weather, spacecraft mission, and payload status.

All of these factors magnify the complexity of the decision-making process being performed by the CR since the dimensionality of the multi-objective function, radio resources, and end-to-end link conditions are much more diverse than on a point-to-point terrestrial link. Not only does the resource management complexity for satellite-based CR increase, but also the learning, reasoning, and decision-making algorithms that must be redesigned in order to cope with all these new challenges while operating in a dynamically changing and complex environment.

Based on the increasing demand for high-throughput satellite-based communications systems, the need for CR systems operating on board satellites possessing different orbits at the same time, with nodes potentially spread throughout the Earth experiencing very different channel conditions, is of high importance and represents as a research gap.

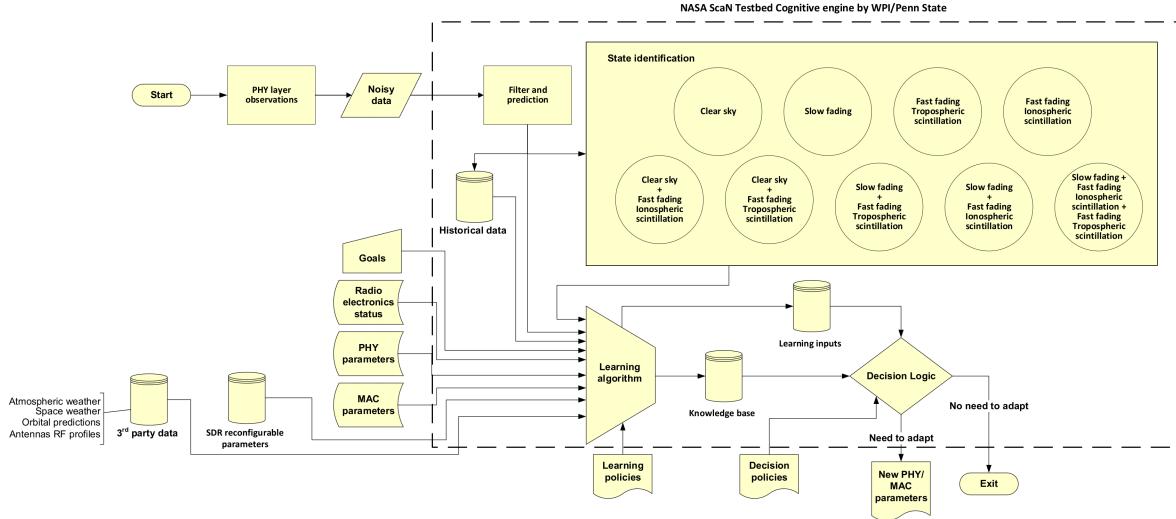


Figure 5.3: Low level diagram block of proposed cognitive engine showing inter-relationships between the channel state detector, learning algorithm, decision logic, and their parameters.

5.2.1 Reinforcement Learning Overview

The contents of this subsection provide a brief summary of relevant concepts underpinning RL found in [133] for the proposed algorithms presented in the following sections. RL is an algorithm designed to learn through interactions with the environment in a trial-and-error fashion, as shown in Figure 5.4. Based on predefined goals, RL looks for actions that optimize its performance.

Multi-armed bandit (MAB) [134–137] models RL problems in which an action set results in rewards, a measurement of how well a certain task was executed. Thus, it can be seen as an optimization problem to find the action set that results in the maximum reward.

Instead of using MAB, these problems could be **modeled as state-transition problems**. The state-transition itself is modeled as a Markov decision process (MDP) [138]. State-transitions can be deterministic, *i.e.*, executing a certain action will always lead the system to that same state, as assumed in this dissertation, or it will make the next state to behave as a random variable.

Usually control problems require the **computation of an optimal policy** that maps observed states into actions that will be taken when the system is in one of those states. Thus, the work presented herein is concerned about **controlling radio parameters** such that opti-

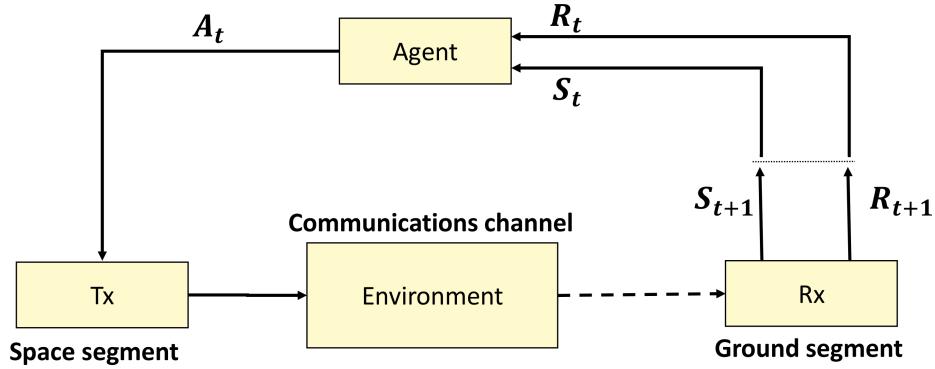


Figure 5.4: Diagram block of reinforcement learning elements within the feedback loop for a down-link pair with the receiver at ground and transmitter in space. A_t , S_t , and R_t are the action, state, and reward at time t , respectively.

mal performance is achieved based on the current environmental conditions and kept there for the entire time, like a regulator. The environment is comprised of the satellite communications channel through which propagating signals are affected by the dynamic geometry of the line-of-sight between transmitter and receiver and its surroundings (buildings in the vicinity of ground stations or structures in the vicinity of the antennas on-board the space-craft), as well as dynamics of atmospheric and space weather. Therefore, a state-transition model and the action-state mapping takes all these variables into account and it is assumed to be unknown due to its high level of complexity.

Fortunately, there are several techniques to compute policies, for which most of the time the environment model, *i.e.*, state-transition model, is unknown due to being too complex or difficult to obtain. In this case, the agent must interact with the environment in an efficient way to find the best policy possible while balancing exploration of new actions and exploitation of known actions. In these cases an action-value function $Q_{\text{RL}}(s, a)$ representing the value of a certain action a taken when in state s while following policy π , should be evaluated at all actions possible from state s through a greedy policy given by:

$$\pi(s) = \arg \max_a Q_{\text{RL}}(s, a), \quad (5.1)$$

where, for every state $s \in S$ an action $a \in A$ with maximal action-value is chosen given the state space S and action space A . For some problems with either continuous or discrete

A containing thousands of actions a , it may **not be feasible to evaluate all action-values** when in a certain state s . This is the case of radio communications, for which exploring each action a over the air may cost time and force the radio receiver to experience a certain performance level. The practical alternative is to ensure that the **agent keeps exploring them using either on-policy or off-policy approaches.** On-policy approaches evaluate or improve policies used to make decisions, whereas off-policy methods evaluate or improve a policy that is learned about, known as target policy, that is different from a policy used to generate behavior, known as behavior policy [133].

A common model-free method to find these policies is **Temporal-Difference (TD)**, which updates the action-value function $Q_{\text{RL}}(s, a)$ using past experiences at each time step, suitable for on-line, *i.e.*, time-sensitive applications. The on-policy TD control is known as State-Action-Reward-State-Action (SARSA) and updates Q_{RL} by computing:

$$\begin{aligned} Q_{\text{RL}k+1}(s_k, a_k) = & Q_{\text{RL}k}(s_k, a_k) + \\ & \alpha_{\text{RL}}[r_{\text{RL}} + \gamma Q_{\text{RL}}(s_{k+1}, a_{k+1}) - Q_{\text{RL}}(s_k, a_k)], \end{aligned} \quad (5.2)$$

where α_{RL} is the learning rate, r_{RL} is the reward, γ is the discount factor, s_{k+1} and a_{k+1} are the state and action chosen for the current target policy, before the Q_{RL} update. The difference within the brackets in Eq. (5.2) is known as **TD error** and computes the difference between the estimated value of $Q_{\text{RL}}(s_k, a_k)$ and a better estimate, $r_{\text{RL}} + \gamma Q_{\text{RL}k}(s_{k+1}, a_{k+1})$. The off-policy TD control algorithm is known as **Q -learning** and is computed by:

$$\begin{aligned} Q_{\text{RL}k+1}(s_k, a_k) = & Q_{\text{RL}k}(s_k, a_k) + \\ & \alpha_{\text{RL}}[r_{\text{RL}} + \gamma \max_a Q_{\text{RL}k}(s_{k+1}, a) - Q_{\text{RL}k}(s_k, a_k)], \end{aligned} \quad (5.3)$$

where the TD error uses the Q -value with the highest value independent of the action. Eqs. (5.2) and (5.3) are derived from the well-known Bellman equations [133, 138].

As mentioned in [4], within the context of decision-making in radio communications the **discounted factor does not have a practical meaning**, since the cognitive radio is interested in the immediate reward ($\gamma = 0$), and any action can be taken from any state without the need for planning. These assumptions result in a modified version of the Q -value functions for both on- and off-policy, which turn out to be the same, given by:

$$Q_{\text{RL}k+1}(s_k, a_k) = Q_{\text{RL}k}(s_k, a_k) + \alpha_{\text{RL}}[r_{\text{RL}k} - Q_{\text{RL}k}(s_k, a_k)]. \quad (5.4)$$

Even though the state-transition model is unknown, some functions need to be defined, for instance, the action exploration function, the state-action policy function h used in $a_k = h(s_k)$, and the reward function g , used in $r_{\text{RL}} = g(s_k, a_k)$.

5.2.2 States, Goals and Rewards

Within a communication system, the RL agent is located at the receiver and feeds the output control, *i.e.*, the action to be taken during the next transmission, back to the transmitter, as shown by Figure 5.4. A key requirement for any RL algorithm is the definition of its states, goals, and rewards.

In this case, the goal is to maximize a performance value, known as reward. After choosing an action and executing it in the environment, a reward is given to the RL agent, and the system is brought to a new state. Rewards measure how close to the goal the system was able to get using a certain action.

The performance value function is composed by several individual **performance functions**, which may or may not be conflicting among themselves, referred to throughout this text as multi-objective performance function. Each individual function is weighted according to the current communications mission profiles (described below).

Thus, the proposed approach in this section consists of **defining the RL state value as a reward value, in percentage**. Then, the RL agent might choose actions, *i.e.*, a new set of radio parameters that may bring this system closer to the “goal state”, represented by 100%. A threshold must be set so that **any chosen action** that brings the system performance **above the threshold is recognized as part of the solution set** by a reward. These actions are described in the following subsection.

Because communication channels are assumed to be a Markov process, with the current channel state being independent from the previous states, every time a RL agent chooses an action, the system can be sent to any possible state. Besides choosing actions, the **RL agent also builds a knowledge base, called a *Q*-table**, where it stores previous experiences, *i.e.*, learning. This *Q*-table maps updated *Q*-values, computed by the Bellman’s equations, Eqs. (5.2)-(5.4), to specific state-action pairs.

Actions that resulted in reward values below the defined threshold receive zero reward.

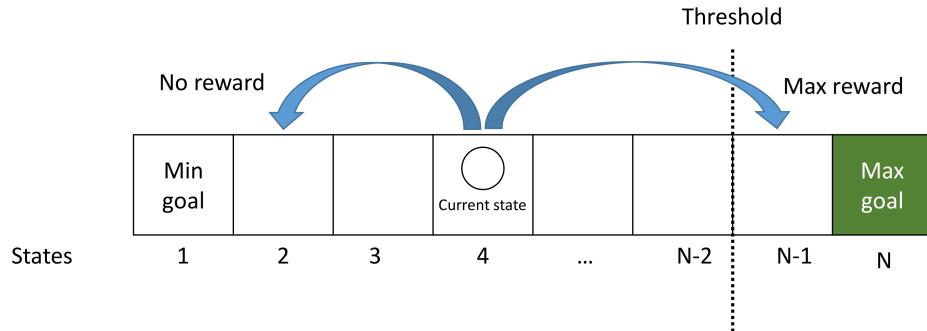


Figure 5.5: State change concept for communications system Markov process. Rewards are given for actions that send the system to states that are above a certain threshold. No rewards are given otherwise. Even if the previous state was rewarded, if a different action results in a state above threshold it must be rewarded.

Doing so, the system “reinforces” actions that make the system achieve its goals, as illustrated by Figure 5.5.

The following performance functions’ goals are considered: **maximizing throughput** (T_{RL}), **minimizing BER**, minimizing power consumption (P_{RL}) to maximize on-board satellite battery life, and keeping the bandwidth W constant.⁵ An analysis is provided based on the proposed approach for using RL to handle the cases when multiple goals are required during a certain satellite communications mission phase.

These goals are achieved individually by parameter adaptation, such as bit rate R_{RL} , modulation scheme M (hereafter comprised of modulation type and index, and coding rate), available power P_{max} for transmission (then E_b is enabled to adapt as well).

Table 5.1 provides relationships between the adaptable parameters, presented as equations for the individual goals [139]. The parameter W can be computed by:

$$W = \frac{N_{RL}R_{RL}}{2 \log_2(M)}, \quad (5.5)$$

where N_{RL} is the number of orthogonal dimensions in the modulation constellation, *e.g.*, $N_{RL} = 2$ for QAM. M is the constellation size of the modulation scheme being used.

⁵ Although a decrease in W could be beneficial for cooperation, for critical missions it could end up attracting secondary users that could cause interference.

Table 5.1: Radio parameter adaptation inter-relationship

Adaptation	Goals	Consequences	Conflicts	Constants
$\downarrow M$	$\min(\text{BER}), \min(P_{\text{RL}})$	$\downarrow T_{\text{RL}}, P_{\text{RL}}, \text{BER}$	$\max(T_{\text{RL}})$	E_b
$\uparrow M$	$\max(T_{\text{RL}})$	$\uparrow T_{\text{RL}}, P_{\text{RL}}, \text{BER}$	$\min(\text{BER}), \min(P_{\text{RL}})$	E_b
$\uparrow R_{\text{RL}}$	$\max(T_{\text{RL}}), W \text{ const.}^*$	$\uparrow W, P_{\text{RL}}$	$\min(P_{\text{RL}}), W \text{ const.}^*$	M, E_b
$\downarrow R_{\text{RL}}$	$\min(P_{\text{RL}}), W \text{ const.}^*$	$\downarrow W, T_{\text{RL}}$	$\max(T_{\text{RL}}), W \text{ const.}^*$	M, E_b

* Keeping W constant can be a goal or a conflicting goal while adapting R_{RL} .

P_{RL} is given by:

$$P_{\text{RL}} = R_{\text{RL}} E_b, \quad (5.6)$$

where E_b is the energy per bit. For two-dimensional modulation schemes, the parameter P_{RL} can be rewritten as:

$$P_{\text{RL}} = W \log_2(M) E_b. \quad (5.7)$$

Based on the bit error probability equations for QAM modulation schemes provided in Reference [139], BER relates to the previous equations by:

$$\text{BER} \approx \frac{1}{E_b}, \quad (5.8)$$

for a fixed M , and by:

$$\text{BER} \approx \log_2(M), \quad (5.9)$$

for fixed E_b . When both M and E_b are varied at the same time, there is a nonlinear relationship with BER.

Thus, the cognitive engine must learn how to tune these parameters while considering other objectives. Clearly, there are several trade-offs while adapting R_{RL} , M , P_{RL} , W , and BER in order to achieve a certain goal. Adaptation of these parameters for one goal might affect the achievement of other goals at the same time. These adaptation consequences are described in Table 5.1.

Attempting to achieve multiple goals at the same time can cause resource competition. An approach to minimize this dispute is to prioritize each goal individually using weights. Thus, communications mission phases are defined such that, for each mission, a predefined set of weight values, vector \bar{w} , is used during the decision-making process. Therefore, the goal of the approach presented is to maximize:

$$f_{\text{decision}_{\max}}(\bar{w}_k, \bar{x}_k, \bar{a}_k) = w_1 f_{\max}(R_{\text{RL}}) + w_2 f_{\min}(\text{BER}) + w_3 f_{\min}(P_{\text{RL}}) + w_4 f_{\text{constant}}(W), \quad (5.10)$$

with $\bar{w}_k = (w_{1_k}, w_{2_k}, w_{3_k}, w_{4_k})$, $0 < w_{1_k}, w_{2_k}, w_{3_k}, w_{4_k} < 1$, $\sum_{i=1}^n \bar{w}_{i_k} = 1$, and $\bar{x}_k = (T_{\text{RL}k}, \text{BER}_k, P_{\text{RL}k}, W_k)$, which is a vector containing the observed values after choosing an action $\bar{a}_k = (R_{\text{RL}k}, M_k, E_{bk})$ containing the adaptable parameters. Among the observed values, values of $T_{\text{RL}k}$, $P_{\text{RL}k}$, and W_k are monitored at the transmitter and sent to the receiver, where BER is estimated in real time at the receiver, by using Eq. (17) in Reference [140], which requires E_b/N_0 to be obtained by SNR measurements.

$f_{\max, \min}$ is a normalized value of each element of \bar{x}_k , based on the current system's available parameter ranges:

$$f_{\max}(T_{\text{RL}}) = \frac{T_{\text{RL}}}{T_{\text{RLmax}}}, \quad (5.11)$$

$$f_{\min}(\text{BER}) = \frac{\text{BER}_{\min}}{\text{BER}}, \quad (5.12)$$

and

$$f_{\min}(P_{\text{RL}}) = \frac{P_{\text{RLmin}}}{P_{\text{RL}}}, \quad (5.13)$$

with the exception of $f_{\text{constant}}(W)$, which is computed as:

$$f_{\text{constant}}(W) = \begin{cases} 0, & \text{if } W \leq \text{BW}, \\ -1, & \text{if } W > \text{BW}, \end{cases} \quad (5.14)$$

where BW is the bandwidth allocated to the communication channel of concern.

In order to reward a chosen action \bar{a}_k , the equivalent of state transition function, in this case, is:

$$s_{k+1} = f_{\text{decision}_k}(\bar{x}_k), \quad (5.15)$$

computed using the same structure of Eq. (5.10), where \bar{x}_k is observed at the receiver, which results in $f_{\text{decision}_k}(\bar{x}_k)$ reflecting the effects of the combined uncertainty imposed by

Table 5.2: Communication mission phases and weights w_i

Mission \ Weights	w_1	w_2	w_3	w_4
1 - Launch/Re-entry	0.1	0.6	0.2	0.1
2 - Multimedia	0.6	0.3	0.05	0.05
3 - Power saving	0.2	0.1	0.6	0.1
4 - Normal	0.25	0.25	0.25	0.25
5 - Cooperation	0.1	0.2	0.1	0.6

both channel impairments and spacecraft orbital dynamics. Therefore, values for r_{RLk+1} in Eq. (5.2) are given by:

$$r_{RLk+1} = \begin{cases} f_{\text{decision}_k}(\bar{x}_k), & \text{if } f_{\text{decision}_k}(\bar{x}_k) > tr, \\ 0, & \text{otherwise,} \end{cases} \quad (5.16)$$

where tr is a threshold value that guarantees that only actions that brought the system performance above a certain level should be accounted for in the learning process. Rewards computed that have a value less than the threshold do not represent a successful action choice, given the current channel environment. Thus, it must not be reinforced and should be forgotten instead.

For simulation purposes, Table 5.2 shows the communication mission phases considered in this section, with a suggestion for their respective weight vectors \bar{w} . Each mission phase is composed of some individual objectives, four in this case, demonstrating the multi-objective optimization facet of the proposed RL learning. Each objective is affected somehow by the combination of several impairment sources such as current weather conditions encountered by the LOS, spacecraft orbital dynamics and on-board electronics status, etc., described in the next section.

5.2.3 Actions and How to Choose Them

An action a , as mentioned above, consists of a set of radio parameters that can be employed by an SDR platform. When choosing an action, the agent might first decide if it should explore new actions in order to expand its knowledge about the current environmental conditions, or if it should exploit already known actions that resulted in bringing the system above the state's threshold tr in the past. It is worth noting that, while following a certain algorithm to choose an action, the decision-making process can be informed by additional knowledge of certain communication channel conditions, reading from external sensors, databases, or predictors such as the IMM proposed by the authors in Reference [78]. In future applications, this bias can be inserted directly into the decision function or indirectly as a modification of the main goal of the current mission.

Several methods were proposed in Reference [133] to solve the explore-exploit trade-off. They propose the classical ε_k -greedy algorithm with $\varepsilon \in (0, 1)$, which represents the exploration probability with uniform distribution that also picks a uniformly-distributed random action, with exploitation for the remaining portion of time. Since ε_k is function of k , it should decrease as the time goes by in order to reflect the system-learning capability, *i.e.*, explore more during the first steps.

One of the main challenges with using ε_k is that it needs to be manually tuned. The adaptive ε_k , value-difference based exploration (VBDE), was proposed to adaptively change the ε_k value based on a temporal-difference error, making the ε_k value a function of state, *i.e.*, $\varepsilon_k(\bar{x}_k)$ [141, 142]. Other common solutions for the exploration versus exploitation problem include the Boltzmann exploration [133], probability matching [143], contextual bandit [144], on-line clustering [145], etc., each requiring its own functions and parameters, usually a designer's choice. This work considers the classic ε_k -greedy algorithm, and analyzes the multi-objective performance with both fixed and varying ε values.

5.2.4 Algorithm

The proposed algorithm covered this subsection, Algorithm 2, implements the basic RL for communications systems illustrated by the block diagram in Figure 5.4. During

the communications mission, the range of radio parameters might change due to system upgrades or failures. When this is the case, \bar{U} , a matrix containing all the current possible adaptable combinations (for instance, T_{RL} , M , and E_b) must be updated.

The proposed RL method requires knowledge of the observable \bar{x}_k . Due to the impairments imposed by the current channel conditions, each different action might result in a specific overall performance value, measured by f_{decision_k} , that has the same structure as Eq. (5.10).

While deciding its actions, the agent uses policy h to choose an action a_k , an entry from matrix \bar{U} . This policy is detailed in the next subsection, and covers how the exploit-explore trade-off was treated. The vector of weights \bar{w} is chosen based on the multi-objective goals of the current communications mission profile. Next, the agent chooses an action, forwards the parameters set to the transmitter, which sends data back using that radio configuration. Finally, the receiver takes measurements, computes the new state and reward, and updates the Q -table.

5.2.5 Simulation Results

Before presenting the numerical results, the communication channel profiles, as well as some parameters and functions considered by the simulations are defined. It is assumed a GEO satellite equipped with flexible-radio payloads communicating at Ka-band (26 GHz) with a fixed ground station, with adaptation occurring on the return link only. Considering clear sky conditions, an additive white Gaussian noise (AWGN) channel is assumed.

The adaptive SDR parameters R_{RL} , M , and E_b are presented in Table 5.3. M is comprised of all combinations of the available QAM modulation constellation sizes M_s and encoding rates M_r . In this section, the action search space \bar{U} is comprised of all possible combinations among the values available within the parameters ranges shown in Table 5.3.

Any study of RL algorithms involving simulation or experimentation requires the researcher to address the challenge of efficiently exploiting known actions while exploring available actions, looking for better ones. An ideal strategy would consist of long exploitation periods mixed with short exploration periods to identify suitable actions as quick as possible. For comparison reasons, four different scenarios of ε values were simulated. The first is called

Algorithm 2 MORL for cognitive radio-based satellite communications

Require: Initial parameters $(\bar{R}_{\text{RL}}, \bar{M}, \bar{E}_b, P_{\text{RL}}^-, W, \text{tr}, \varepsilon_k, \alpha_k)$

```

1:  $\mathbf{U} \leftarrow$  all combinations of  $(\bar{R}_{\text{RL}}, \bar{M}, \bar{E}_b)$ 
2:  $Q_{\text{RL}0} \leftarrow 0$ 
3: Measure and compute initial state  $s_0$ :
4:    $a_0 \leftarrow \bar{U}$ 
5:   Apply  $a_0$  and measure  $\mathbf{x}_0 = (T_{\text{RL}0}, W_0, P_{\text{RL}0}, \text{BER}_0)$ 
6:   Compute  $f_{\text{decision}}(\mathbf{x}_0)$ 
7: while termination condition is not met,  $k = 0, 1, 2, \dots$  do
8:   if  $(\bar{R}, \bar{M}, \bar{E}_b, \bar{P})$  has changed then
9:      $\mathbf{U} \leftarrow$  all combinations of  $(\bar{R}_{\text{RL}}, \bar{M}, \bar{E}_b)$ 
10:    end if
11:    zz  $\leftarrow$  uniform random number  $[0, 1]$ 
12:    if  $zz > \varepsilon_k$  then                                 $\triangleright$  with probability  $1 - \varepsilon_k$  (Exploit)
13:       $a_k \leftarrow a \in \arg \max_{\bar{a}} Q_{\text{RL}k}(s_k, \bar{a})$ 
14:    else                                          $\triangleright$  with probability  $\varepsilon_k$  (Explore)
15:       $a_k \leftarrow \mathbf{U}$  randomly chosen action with uniform probability
16:    end if
17:     $\bar{w} \leftarrow$  Communications goals
18:    Apply  $a_k$  and measure  $\mathbf{x}_k = (T_{\text{RL}k}, W_k, P_{\text{RL}k}, \text{BER}_k)$ 
19:    Compute next state  $s_{k+1}$  and next reward  $r_{\text{RL}k+1}$ 
20:    Update  $Q_{\text{RL}k+1}(s_k, a_k)$ 
21: end while

```

Table 5.3: SDR parameter ranges

Parameters	Bit rate (R_{RL})	Constellation size (M_s)	Encoding rate (M_r)	Tx gain (E_b)
Values	[128 kbps, 1 Mbps] [*]	[4, 16, 64]	[4/7, 11/15]	[1, 20] [†] dB

^{*} Steps of 128 ksps.

[†] Transmitter gain range achievable depending on combinations of available values for E_b and R .

the “brute force” (BF) scenario, in which all actions are evaluated once. In this case, $\varepsilon = 1$ and only exploration is performed, with its duration in time being equal to the total number of possible actions. The second and third scenarios consider fixed values for ε , 0.5 and 0.01, respectively.

The last scenario considers a varying ε that decreases with the iteration number k , and gets reset when it reaches a value below a threshold, in this case assumed to be equal to 10^{-4} . The analysis of the rate at which ε decreases and gets reset is beyond the scope of this work.

Decisions of whether to explore or exploit consist of drawing a random number that is uniformly distributed before the transmission of each packet. The current ε has the role of serving as the threshold, since it represents the exploration probability.

In the case of exploitation, the chosen action is the one associated with the current maximum Q -value, $h = \arg \max_a Q_{RL}(s, a)$, computed by Eq. (5.4). When exploring, a random action is chosen from a uniform distribution, its corresponding radio parameters are applied at the transmitter, and its performance is measured by the receiver. Regarding the threshold shown in Figure 5.5, it is assumed zero in order to allow a more detailed study of the algorithm’s behavior at any performance level. Also, the learning rate is initialized as 1 for each possible action and decreased as it gets reused, making sure that selecting repeated actions does not provide new knowledge.

Assuming an SNR reading rate of 1 reading/second, each action is used during 1 second. Given an universe of 540 available actions, combinations of the available tunable radio

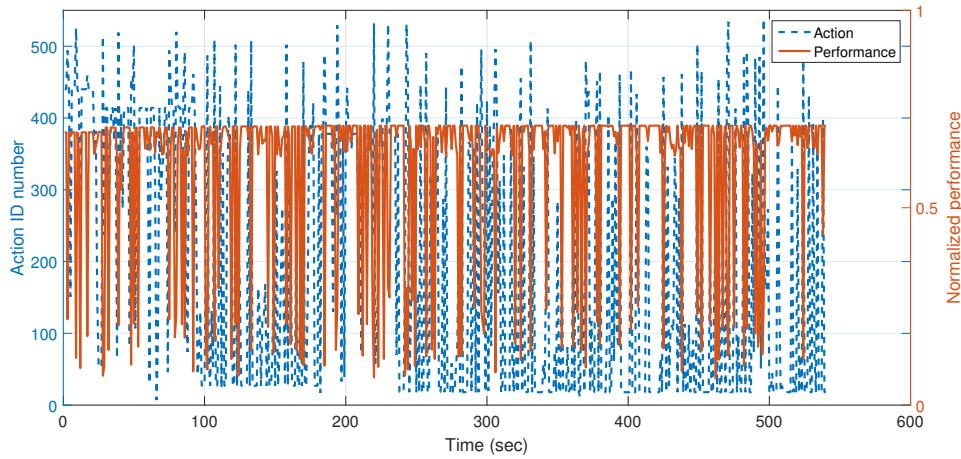


Figure 5.6: Time-series for chosen action and multi-objective communication performance for $\varepsilon = 0.5$. Knowledge of good performing actions is taken for granted due to the fixed exploration probability.

parameters, the BF scenario also lasts 540 seconds. In order to allow a fair comparison, all scenarios also have the same simulation duration as the BF.

Examples of time series for actions chosen and their respective multi-objective performance value as normalized are provided in Figure 5.6 for a fixed ε , and in Figure 5.7 for varying ε . The communication channel assumed was a direct link from a GEO satellite during clear sky conditions to a fixed ground station at Ka-band.

One can observe that, for a fixed ε , the system does not take full advantage of the knowledge previously acquired. Thus it keeps exploring with a constant pace even after finding sufficient rewarding actions. On the other hand, the varying ε the knowledge is used wisely with exploration being performed only sporadically.

In order to provide an overview of the multi-objective performance, each mission was simulated 1,000 times, with a duration of 540 seconds each. The 25th, 50th and 75th percentiles for the mean multi-objective performance of the four scenarios, for each mission, are shown in Tables 5.4–5.8. Higher percentile values represent a better performance, since the percentiles themselves refer to multi-objective performance levels, with the 25th percentile considered a baseline for performance evaluation between different ε values. Having large

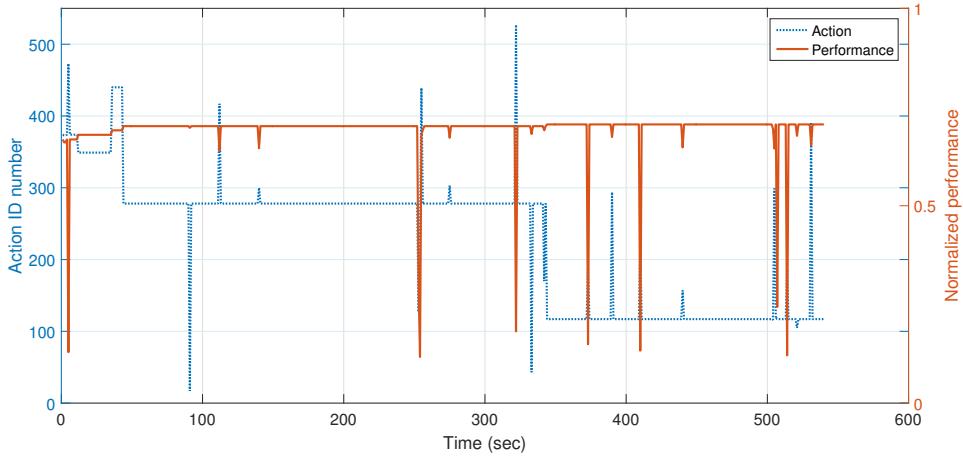


Figure 5.7: Time-series for chosen action and multi-objective communication performance for varying ε . The best known actions are exploited for a while, and eventually new ones are explored.

25th percentile values means that the lowest 25% of the multi-objective performance values are equal or less than that percentile. In addition to performance percentiles, integral values of histograms' areas are also provided.

Histograms for BF and varying ε scenarios, including standard error bars for these, for each mission are provided in Figure 5.8. For all missions, the effect of using the RL algorithm to manage exploration and exploitation periods while learning which actions were the best ones, results in a higher time spent on those high-rewarding actions when compared to a scenario without this reinforced learning capability, such as the BF.

Since the multi-objective function weights are different for each mission, each one has a different performance profile. Independent of the profile, based on both results shown in Tables 5.4–5.8 and in Figure 5.8, the RL effect is to focus on the most rewarding actions, located on the rightmost side of the BF histograms, for the whole operation time duration.

The best performances for clear sky conditions are achieved for Missions 1 and 2, since the 25th percentile difference between BF and varying ε scenarios, as well as their respective integral values, are the largest ones. For instance, the improvement achieved for Missions 1 and 2 was sufficiently good that the 25th percentile difference between BF and varying

Table 5.4: Multi-objective Communications Performance Distribution for Mission 1

Percentile \ Scenarios	BF	$\varepsilon = 0.5$	$\varepsilon = 0.01$	Varying ε	
Percentile	Scenarios	BF	$\varepsilon = 0.5$	$\varepsilon = 0.01$	Varying ε
25th		0.1917	0.6109	0.5923	0.6892
50th		0.6476	0.6203	0.6467	0.6937
75th		0.6769	0.6282	0.6754	0.7095
Integral		0.4863	0.6224	0.6294	0.7002

Table 5.5: Multi-objective Communications Performance Distribution for Mission 2

Percentile \ Scenarios	BF	$\varepsilon = 0.5$	$\varepsilon = 0.01$	Varying ε	
Percentile	Scenarios	BF	$\varepsilon = 0.5$	$\varepsilon = 0.01$	Varying ε
25th		0.3699	0.708	0.6559	0.8114
50th		0.5487	0.7151	0.7209	0.8289
75th		0.6903	0.7205	0.7665	0.8679
Integral		0.531	0.7235	0.7092	0.8428

ε scenarios was 0.49 and 0.44, respectively. Also, depending on the mission requirements multiple parameters need to be taken into consideration. Even though Mission 1 had a larger difference value, representing an improvement of more than 3.59 times between these two scenarios, the 25th percentile for Mission 2 was the highest among all the missions simulated.

5.2.6 Conclusions

For clear-sky GEO satellite channel, results demonstrated improvements of 3.59 times in the 25th percentile compared to systems without a learning algorithm under the same channel condition. The results also demonstrated that, even though the multi-objective performance can be improved by the proposed algorithm, depending on the communication

Table 5.6: Multi-objective Communications Performance Distribution for Mission 3

Percentile	Scenarios	BF	$\varepsilon = 0.5$	$\varepsilon = 0.01$	Varying ε
25th		0.1861	0.3824	0.2752	0.3524
50th		0.2285	0.4222	0.3023	0.3791
75th		0.2721	0.441	0.3211	0.403
Integral		0.2308	0.4094	0.2982	0.3789

Table 5.7: Multi-objective Communications Performance Distribution for Mission 4

Percentile	Scenarios	BF	$\varepsilon = 0.5$	$\varepsilon = 0.01$	Varying ε
25th		0.2379	0.4172	0.3996	0.4688
50th		0.3438	0.4203	0.4309	0.4758
75th		0.4233	0.4236	0.4501	0.4934
Integral		0.3291	0.4251	0.4247	0.4851

Table 5.8: Multi-objective Communications Performance Distribution for Mission 5

Percentile	Scenarios	BF	$\varepsilon = 0.5$	$\varepsilon = 0.01$	Varying ε
25th		0.1139	0.2506	0.2427	0.2877
50th		0.2375	0.2526	0.2649	0.2904
75th		0.2693	0.2544	0.2771	0.2979
Integral		0.2007	0.2613	0.2598	0.2998

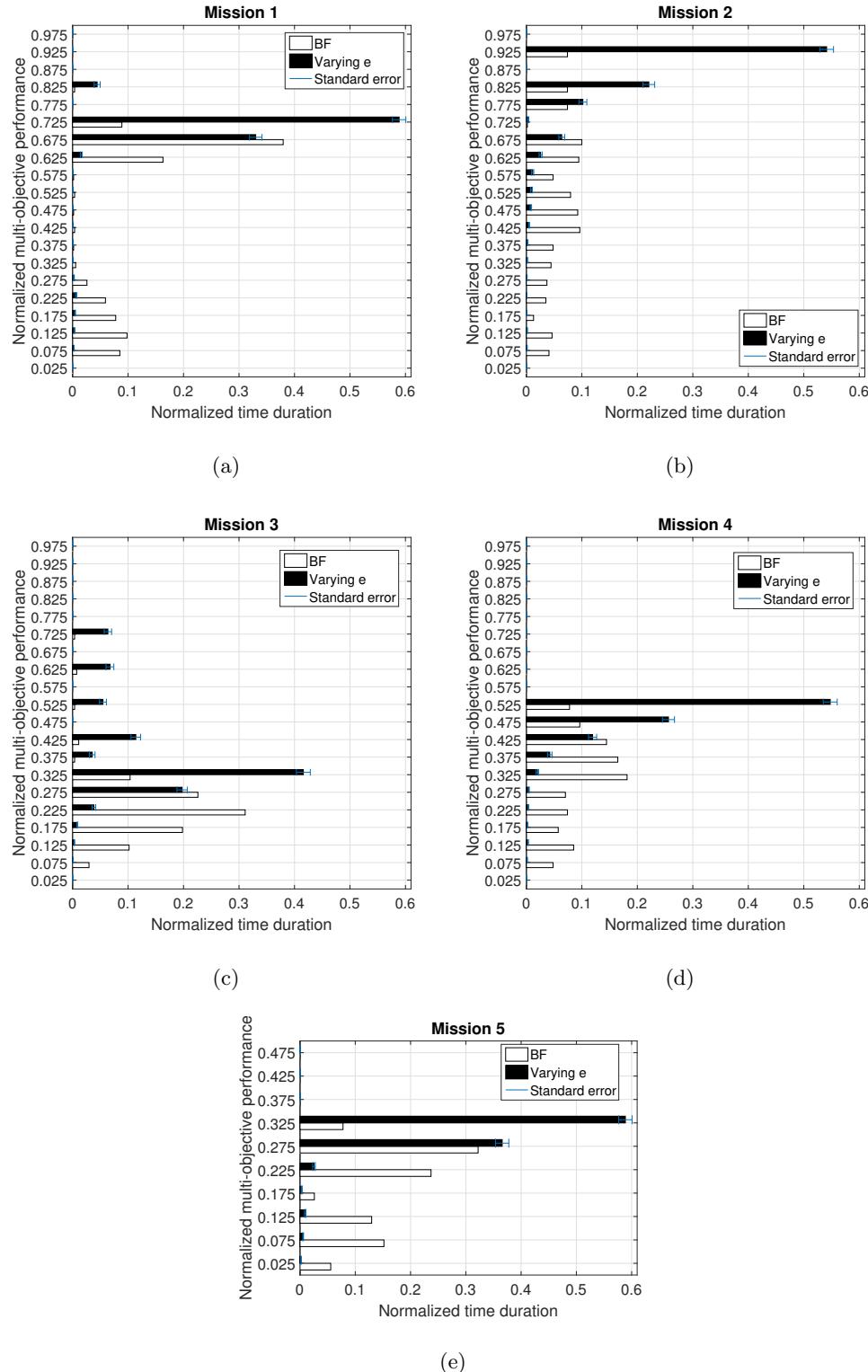


Figure 5.8: Normalized histograms for average time spent on a certain multi-objective normalized performance level for a GEO satellite-based link under clear sky conditions. For varying ϵ scenario, in all missions, more time was spent at higher performance levels when compared to the BF scenario.

mission requirements certain choices of the exploration probability ε can be made to achieve a specific goal, such as minimum acceptable performance level, or overall performance within a time window period, by imposing conditions on the percentiles and integral values.

Based on the improvements shown in this subsection, the potential for RL is further explored in Section 5.3 to assist cognitive satellite-based systems dealing with varying communication channels.

5.3 Hybrid Machine Learning Methods for Cognitive Space Communications

In the previous section the benefits of RL were presented as a method to control radio adaptations on return links from a GEO satellite to a fixed ground station assuming an AWGN channel during clear sky conditions. However, there are several other situations in which the proposed approach does not seem to be suitable, requiring a complete new way of approaching the problem. This new approach is covered in this section and resulted in a whole new set of solutions, described in the following subsections.

Some of the main issues with the previous approach is that it relied completely on the *Q-table*, where discrete performance percentage values were mapped into actions through *Q-values*. In that case, the number of actions were relatively small, only 540 parameter combinations. For real-world applications, both state and action universe needs to be considered continuous, making the usage of a *Q-table* impractical from a practical implementation perspective, due to memory limitations.

In addition to both adaptive parameters and multi-objective function being limited to four parameters each, adaptations took place just once per second, and the channel was assumed constant over time. For future applications to be more realistic and robust, a learning system is expected to operate across a dynamically changing channel, on a packet basis, and capable of dealing with several thousands of different actions compliant with current state-of-the-art communications standards.

In Section 5.2.1 the normalized multi-objective performance time-series in Figures 5.6 and 5.7 showed several time instants when the system experienced performance values extremely

below the one currently being exploited. Future critical space-based communications systems may not tolerate the problem of RL spending too much time on exploring actions that result in low performance scores.

Thus, a novel hybrid algorithm is proposed to solve this issue. It is composed by the RL structure proposed in Section 5.2 and neural networks (NN) [146], known as NN-based RL, or RLNN. This proposed algorithm enables the radio to predict the effect of multi-dimensional radio parameters on multi-dimensional conflicting performance goals before allowing the radio to actually try these parameters over the air, avoiding the cost of spending time and resources on learning action–performance mapping that will not be useful in the near future. In addition, the second version of the RLNN, RLNN2, proposes to implement the RL exploitation through an array of ensembles of shallow NNs.

5.3.1 Machine Learning Overview

Machine learning (ML) is a term used to describe several theories and algorithms that aim to automate computational decision-making tasks. The ultimate goal of an autonomous system is to learn, reason, and make decisions without human interventions, applying the knowledge acquired from different tasks to tasks it has never learned before. Another big challenge is to accumulate all that knowledge over time in an efficient way without the need for additional storage, making the information available whenever it is required to be used on different tasks. However, such complex autonomous systems are tens of years away from being implemented due to hardware and software not being available today. Thus, researchers have been working on developing and improving simpler algorithms capable of learning a few different specific tasks.

Two big research advancements recently took place in 2015, with the DQN system used to play Atari games better than a human [147] and in 2016, after the AlphaGo [5] system won the world championship in the game of Go. Both systems, and the ones that have been built upon them, have been pushing the research boundaries of ML in all different fields of application, with the main driver being computer vision systems to assist self-driving cars [148]. Among the most notorious companies involved with ML R&D are Google Brain [149], DeepMind [150], OpenAI [151], Facebook AI Research [152], NVIDIA [153],

and all self-driving car research companies. These companies conduct research on a wide range of topics, including neuroscience and non-convex optimization algorithms and their implementation in dedicated hardware processors. Other specialized research centers try to focus on the big picture, putting together all these different knowledge fields in the same office, such as MIT’s Center for Brains, Minds, and Machines (CBMM) [154].

All these systems have in common the following: both leverage RL, fully described in Section 5.2.1, and deep NNs, briefly described below (including references that will provide the reader with an in-depth theoretical description). This recent technological revolution has inspired the proposal of a satellite-based communications system control that makes use of both these concepts. Nevertheless, the requirements are different and, to the best of the author knowledge, lead to the research and development of algorithms that were not available in the current literature. With the basic principles of RL and NN, the proposed hybrid approach design is presented, followed by simulation results and a brief discussion on the findings.

5.3.2 Neural Networks Overview

An artificial NN is a method for mapping inputs to outputs, usually used for classification, such as in pattern recognition problems, or non-linear function approximation, such as in function fitting problems [146]. For instance, NN is used to approximate the non-linear environment effects by mapping actions into rewards, and states into actions. Over the past 10 years considerable improvements have been made to NN that resulted in the “deep learning” concept [155], where very complex relationships between inputs and outputs can be mapped through “deep” NN consisting of three layers or more.

NN algorithms are composed of two main steps: training and prediction. Initially, examples containing input and output data are preprocessed and provided to the NN for training. After meeting some minimum performance requirements, the trained NN, which consists basically of the NN architecture and its weights, can be used as a predictor.

For detailed description of NN basics, derivations, and algorithm details, the interested reader should refer to Chapter 6 in [155]. When using NN, each different problem seems to require a specific NN architecture, comprised of a training function, performance me-

trics, and number and size of hidden layers such that its usage becomes feasible for the desired application in terms of required processing capabilities and processing time, as well as generalization error performance. Currently there are no general guidelines in the literature on how to pick these items. Reference [156] provides some useful comments and advice on what to consider when making these decisions. For implementation purposes, the standard multi-layer fully connected NN architecture is considered, which is trained by a backpropagation-based algorithm. More details on the chosen NN architectures are provided in the following subsection.

5.3.3 MORL and Deep Neural Networks: The RLNN Algorithm

The NN is used as an approximation of the environment in terms of the RL experience, chosen actions, and its respective rewards achieved so far, as shown in Figure 5.9. By having the luxury of approximating the mapping of actions into states and rewards, the NN allows for actions to be explored all at once, or as many as one would like to, without having to actually spend time trying those actions in the real environment. This proposed approach is called “virtual exploration”. It improves exploration performance by eliminating the time the RL agent would spend exploring actions that are predicted to result in bad performance. This additional feature is called “action rejection”.

Poor actions are defined by a “rejection performance threshold” value defined by the user, such that actions resulting in performance below that threshold are classified as bad. The action rejection probability defines the time percentage that bad actions will be rejected during exploration, *i.e.*, prevented from being used over the air by the radio. Whenever the RL agent is exploring, it predicts the actions’ performance using the trained NN and classifies them into good or bad. Then, according to the rejection probability it randomly chooses one action from either a good or bad set.

Regarding the deep exploration NN architecture, a feedforward with Levenberg–Marquardt backproapgation training algorithm [157, 158] was used. The NN has three fully-connected layers without bias: two hidden layers that contain 7 and 50 neurons each (resulting in 449 weight parameters per NN), both using a log-sigmoid transfer function, and the output layer with one neuron using the standard linear transfer function. With respect to the

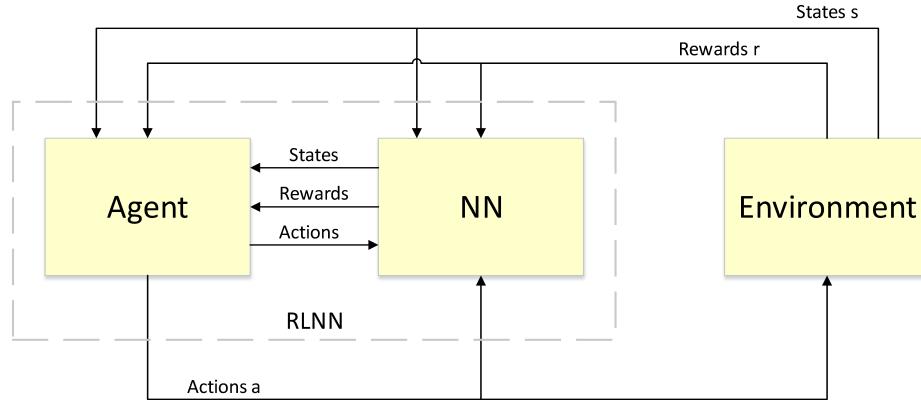


Figure 5.9: The proposed RLNN block diagram, composed by the hybrid multi-objective reinforcement learning assisted by a deep neural network. “Virtual exploration”, driven by “action rejection” probability, prevents time expenditure exploring “bad actions” using radio resources over-the-air.

performance function, the mean-squared error was used with two different training stop conditions: minimum error gradient of 10^{-12} and maximum validation checks equal to 20. During training the data were randomly split into 70% for training, 15% for testing, and 15% for validation, all scaled to the $[-1, 1]$ range.

In order to improve the NN prediction error, an ensemble of $NN_1 = 20$ NNs were used during both training and prediction, as shown in Figure 5.10, and the output was simply the average among all these NNs. The reason for the choice of the ensemble size was made using the MSE during NN training, similar to the way the NN architecture itself is chosen, due to the lack of a more formal theoretical method. Additional discussion on this size is provided at the end of Section 5.3.7.

This hybrid approach is depicted in Figure 5.9 where the RL interacts with the ensemble of exploration NNs, receiving the same actions sent to the environment and the same rewards and state information from the environment during training. When used for prediction, this information is exchanged only with the RL agent, avoiding the cost of executing such actions in a real-world environment.

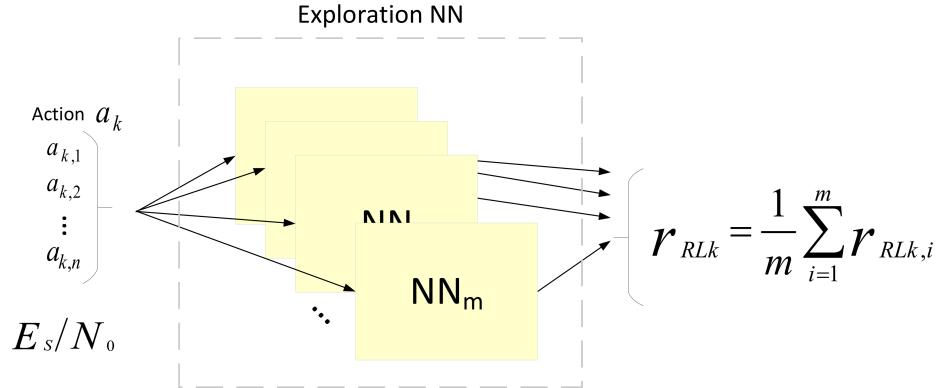


Figure 5.10: Deep neural networks used by RL exploration receive the same multi-dimensional input. Ensemble learning trains several NNs in parallel, and averages their outputs into a single multi-objective performance value. For AWGN channel, the mean E_S/N_0 was kept constant.

5.3.4 RLNN Simulation Results

In order to comply with the DVB-S2 standard, the radio-adaptable parameters are the same as defined in [159], considering all four modulation schemes (QPSK, 8-PSK, 16-APSK, and 32-APSK) and their respective encoding schemes. In addition to all roll-off factors, the following were considered: bandwidth range of [0.5–5 MHz], additional variable transmission symbol power range of [0–10 dB] in steps of 1 dB, and long-frame with frame length equal to 64,800 bits. The action space is comprised of more than 30,400 possible actions. Each action vector \bar{a} is composed of six parameters a_n , where $n = 1, \dots, 6$: symbol rate (Rs), energy per symbol (E_s), roll-off factor (rof), modulation order (M), number of bits per symbol (b_s), and encoding rate (er). Values for the ranges of each of these parameters are described in Table 5.9.

The GEO satellite channel is assumed to be an AWGN during clear sky conditions, with adaptation taking place on the downlink to a fixed ground station only, similar to the channel presented in [4]. In these simulations, for proof-of-concept purposes it was assumed that the satellite's transmitter amplifier operates in the close-to-linear region.

Regarding performance, the conflicting multi-objective target considered is comprised of six parameters: bit error rate (BER) estimated at the receiver, throughput (Thrp), band-

width (BW), spectral efficiency (Spc_eff), consumed power (Pwr_con), and power efficiency (Pwr_eff) measured at the transmitter and sent over to the receiver, all of which are scaled to the range of $[0, 1]$. The reward function g is computed by the fitness function f_{obs} given by the weighted sum computed by:

$$\begin{aligned} f_{\text{obs}}(x) = & w_1 f_{\text{Thrp}} + w_2 f_{\text{BER}} + w_3 f_{\text{BW}} + \\ & w_4 f_{\text{Spc_eff}} + w_5 f_{\text{Pwr_eff}} + w_6 f_{\text{Pwr_con}}, \end{aligned} \quad (5.17)$$

where x is a vector containing the performance parameters, described above, and the weights w_i for each performance parameter, specified according to each different communications mission and defined by the user. The following simulation results considered all $w_i = 1/6$.

As mentioned in Section 5.2.1, the action exploration functions are: (i) constant exploration probability $\epsilon = 0.5$ and the well-known ϵ -greedy exploration algorithm [160, 161] with variable exploration probability $\epsilon = 1/k$, where k is the step size between resets of ϵ back to 1 whenever it reaches a minimum, in this case assumed to be equal to $4 \cdot 10^{-3}$. In this proposed hybrid solution, the state-action policy function h is approximated by the NN during virtual exploration followed by uniform random sampling between the bad and good action sets, based on the rejection probability value. During exploration, h is greedy and chooses the action with the maximum Q -value, computed by Eq. (5.4), with $\alpha_{\text{RL}k} = \alpha_{\text{RL}k-1}/2$, another user-defined parameter that decreases from 1 until it meets a threshold of 10^{-3} , when it gets reset.

Figure 5.11 illustrates an example of a time series performance of the NN rejecting actions chosen by the RL exploration algorithm that were predicted to perform below the threshold of 0.7. Instead, it suggests actions predicted to perform above that threshold value.

Simulations were performed for the proposed hybrid RLNN algorithm and for a modified version of the RL algorithm proposed by the authors in [4], this time considering DVB-S2 and additional adaptable parameters as mentioned above. Figure 5.12 presents the average distribution of the amount of network packets according to their fitness score for both algorithms while using exploration probabilities equal to $\epsilon = 0.5$ and $\epsilon = 1/k$. This distribution accounts for performance during exploration only. A total of 100 simulations

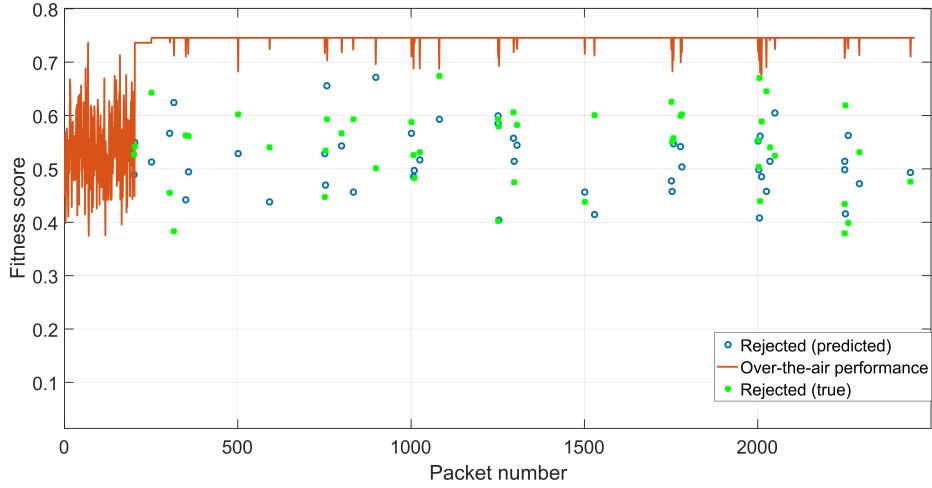


Figure 5.11: Example of a 50 seconds time series multi-objective RL using only “virtual exploration” NN with action rejection probability equal to 1 and performance threshold equal to 0.68. During the first 200 packets, the NN collects training data. Clearly, exploration of actions with predicted performance values below the threshold are avoided. Performance predictions had good accuracy with respect to their respective true values.

were run for each of the four different configurations (combinations between exploration functions and virtual exploration set on/off), with the same simulation duration of 512 seconds. Even though the channel considered in these simulations is assumed constant over time (no slow or fast fading), this time duration represents the average duration of a LEO orbit and may allow performance comparisons to be done in future research by the authors.

5.3.5 Remarks on RLNN Performance

As expected, the introduction of the NN for virtual exploration allows the radio to drastically decrease the time spent, and consequently the number of packets, on exploring actions that resulted in poor performance when compared to the maximum performance achieved while rejecting all those actions predicted to perform below a threshold. This improvement can be seen as a shift to the left in the distributions shown on the right-hand side panels in Figure 5.12. For these results the rejection performance threshold was considered to be equal to 95% of the current maximum performance predicted by the NN.

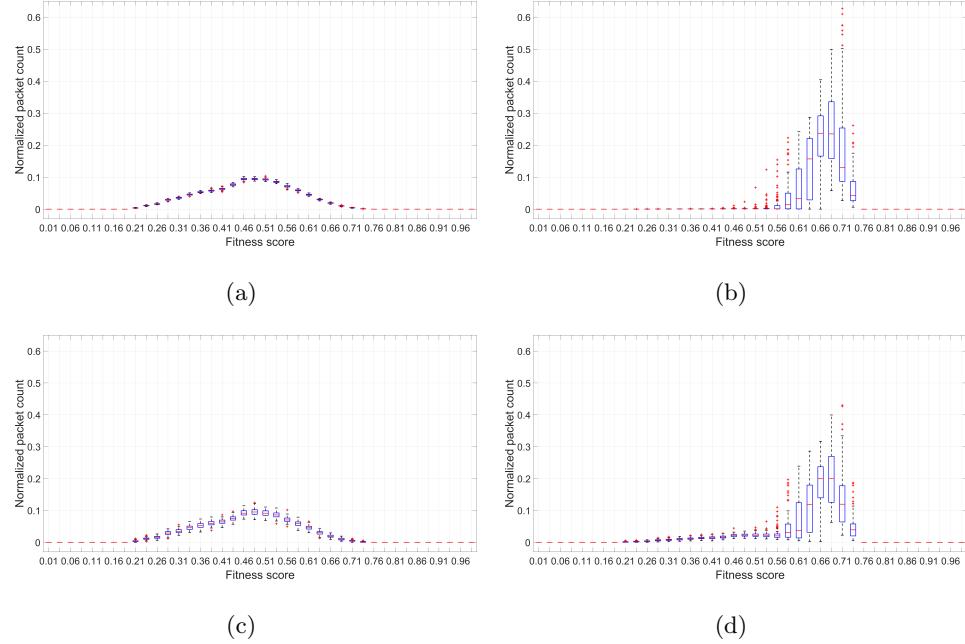


Figure 5.12: Boxplots of proposed RLNN algorithm multi-objective performances. On the left panels virtual exploration was turned off (likewise the algorithm proposed in [4] considering the DVB-S2 standard), and on the right panels it was turned on with rejection probability equal to 1. Top panels used fixed exploration probability value equal to 0.5, and bottom panels used the variable exploration probability function. High number of packets concentrated around larger fitness scores is better.

The virtual exploration feature was disabled and enabled by setting the rejection probability to 0 and 1, respectively, meaning that 0 no action is to be rejected and 1 all actions with performance below the selected performance threshold will be rejected.

In terms of resultant numerical performance, in scenarios with virtual exploration disabled, the average number of packets experiencing multi-objective performance values above 0.56 when using a fixed and variable exploration probability values, was 33% and 25% respectively, as shown by Figure 5.12 panels (a) and (c). In both scenarios, the majority of packets experienced a performance score value of 0.485.

However, in scenarios with virtual exploration enabled, the average number of packets experiencing multi-objective performance values above 0.56 when using a fixed and variable

exploration probability values, was 82% and 98%, respectively, as shown by Figure 5.12 panels (b) and (d). In both scenarios, the majority of packets experienced a performance score value of 0.685. This represents an improvement in the number of packets experiencing performance values above 0.56 of 2.48 times and 3.92 times for fixed and variable exploration probabilities, respectively.

In terms of the integral values of the average of histograms, both scenarios with virtual exploration disabled have an integral equal to 0.472, while the scenarios with that feature enabled have an integral equal to 0.67 and 0.62, for fixed and variable exploration values, respectively. Thus, improvements of 1.32 times and 1.42 times on the integral values were achieved by the proposed RLNN for fixed and variable exploration probability values, respectively.

It should be noted that using a fixed exploration probability value, the number of packets used during exploration represented 50.58% and 50.7% of the total for scenarios with and without virtual exploration, respectively. When using the variable exploration function these percentage values were 3.17% and 3.12%, respectively. These values combined with the distributions shown in Figure 5.12 and with the improvements in the integral values demonstrate the effectiveness of the proposed virtual exploration in increasing the number of packets experiencing high multi-objective performance values independently of the exploration probability function chosen.

For comparison, a GA simulation was run for 100 times, each with the same time duration of the RLNN simulations mentioned above. Its average performance distribution is shown in Figure 5.13. Although it was able to find higher performance scores than the proposed RLNN, the cost to achieve that was to spend 66% of the time exploring actions that resulted in very low performance values, scored between 0.18 and 0.26. Only 0.8% of the time was spent on performance values between 0.69 and 0.81.

Batch methods, such as GA might have an advantage over standard RL and the proposed RLNN for the cases when the environment remains constant and/or the system can spend a long time exploring a large number of different actions, resultant from different GA generations. However, if the environment changes, a reset may be required, which will make the system to spend a considerable amount of time experiencing low performance values

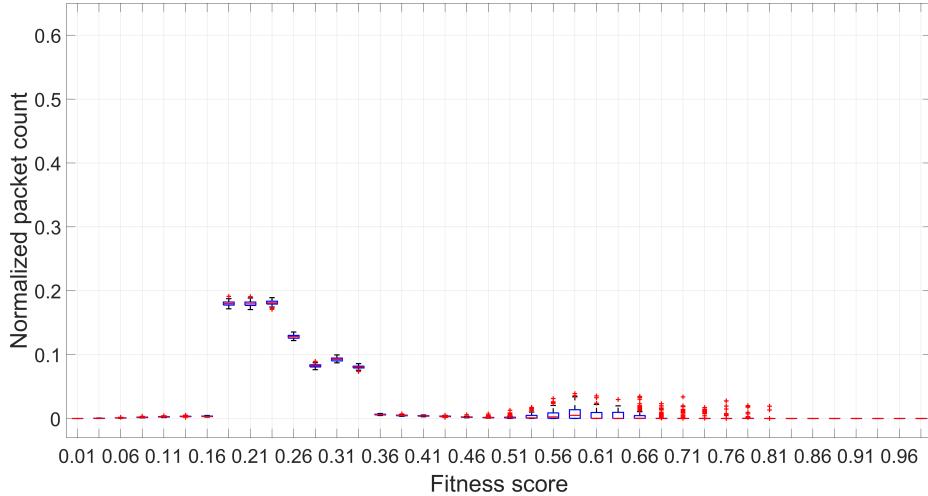


Figure 5.13: Boxplot of the genetic algorithm multi-objective performance during the same time duration used by the proposed RLNN simulations. The majority of the time is spent in very low performance levels.

after restarting the search again. If the system cannot wait until the GA convergence, it might stick with using an action that can have any performance level.

Even though evolutionary methods might be good as searching methods, they do not guarantee a minimum performance. The proposed RLNN method does not guarantee a specific performance level either. However, through virtual exploration it provides guidance in which actions to explore, giving control over the performance levels experienced during exploration by performing action rejection. In addition to that, through the rejection probability value, the RLNN provides control over the amount of time spent on actions that may result in a certain performance level.

5.3.6 Exploiting Reinforced Multi-dimensional Actions: The RLNN2 Algorithm

A novel algorithm combines the RLNN algorithm with a new one capable of dealing with dynamically changing attenuation levels on the satellite communications channel is proposed in this subsection. To address this issue the usage of a hybrid MORL using two different NN's is proposed: one for exploration and another one for exploitation, as illustra-

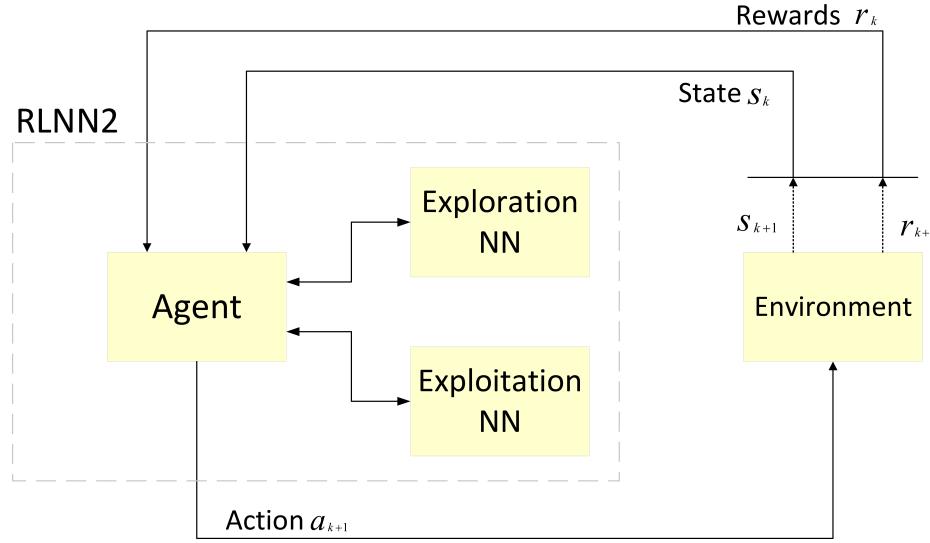


Figure 5.14: Multi-objective reinforcement learning using deep neural networks ensembles block diagram. Virtual environment exploration prevents communications systems from spending time exploring combinations of radio parameters that would result in poor performance, while exploitation networks enables learning of actions best suitable for a dynamically changing environment.

ted by Figure 5.14. Leveraging the proposed RLNN algorithm, the agent interacts with the environment, and either explores different actions or exploits actions already tried before. This is shown by Figure 5.15. During exploration the aforementioned virtual exploration is used, and during exploitation a novel technique is introduced, called “multi-dimensional action predictor”.

Before proceeding with the proposed algorithm, it is worth mentioning the distinction of two important terms: RL state and environment state levels. RL states are the observed individual performance metrics measured and/or monitored from the environment and they change in response to the execution of different RL actions. As described in Section 5.12, these RL states are features of the fitness function. In satellite communications, the channel itself is assumed to convey all the environment changes, represented and measured by the SNR levels at the receiver. Thus, environment state levels are also referred to as channel conditions.

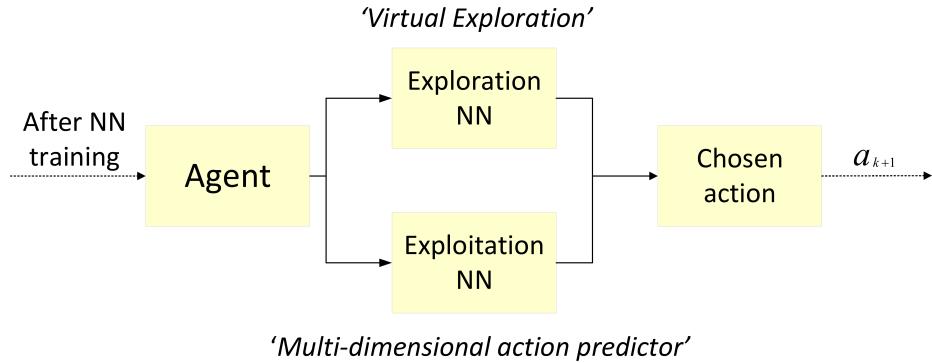


Figure 5.15: RL work flow for choosing an action to be used in the real-world environment after training both exploration and exploitation NN. The exploration probability aides the choice between exploration and exploitation.

One of the main limitations of the basic RL, proposed in Section 5.2, and the RLNN algorithms is that both consider only non-changing environments, since the channel was assumed to be a downlink from a GEO satellite to a fixed ground station. When the channel remains constant over time, the action performance vector also holds the same values. Figure 5.16 shows a time series example of the multi-objective fitness score over time for each action when a constant environment level was assumed during the communications Mission 4 (more details on mission profiles are given in Section 5.12). After randomly exploring actions for a few seconds, the RL starts to alternate between exploitation (using the same selected action during a certain period of time, shown by the constant amplitude bands in Figure 5.16) and exploration.

Each discrete action used over-the-air had its fitness score value stored in a Q -vector (assuming the rewards to be equal to the RL states, see References [4,122]), which received new values for new positions during exploration and updated existing ones during exploitation. Therefore, every explored action decision considered performance values of actions explored in the past.

However, there are three main limitations in using this approach when the channel is assumed to change dynamically: (i) A discrete action performance value in the present might be different from a value registered in the past for that same action; (ii) Performance values regarding a certain action must be stored for each different multi-dimensional RL

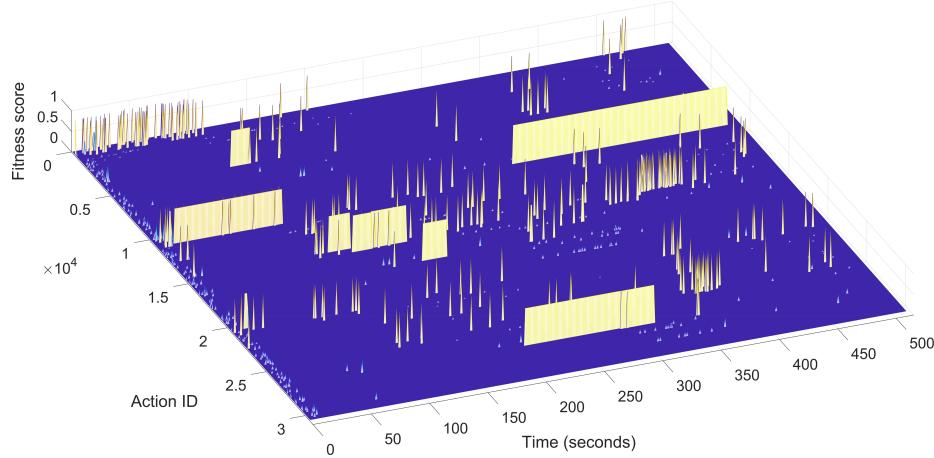


Figure 5.16: RL algorithm searching for better actions to be exploited over a GEO satellite communications channel during Mission 4. Before any transmission the RL exploitation action decision-maker considers performance values of actions when they were last explored. Exploited actions are represented by bands of constant amplitudes for a certain time duration.

state, with each RL state being represented by a continuous variable; and (iii) Each action-RL state performance value must be stored for a specific channel condition that changes dynamically, which is also a continuous variable.

Limitation (i) results in outdated performance values leading to erroneous decision-making when deciding to exploit a certain action that was explored in the past when the environment condition was different. A possible solution to this issue would be to update that specific action performance. Although this update must be done for all other actions previously explored, it would be just another exploration round, still making their performance unknown during the exploitation round. In Figure 5.16, during the very first seconds the RL agent only explores different actions. Next, it starts to alternate exploration with exploitation, resulting in the constant fitness score value bands for exploited actions during a certain time duration. When looking for an action to exploit the RL agent considers the fitness scores of all actions when they were last explored (actions never explored were assumed to have a score equal to 0). For an environment that does not change over time this does not seem to be an issue.

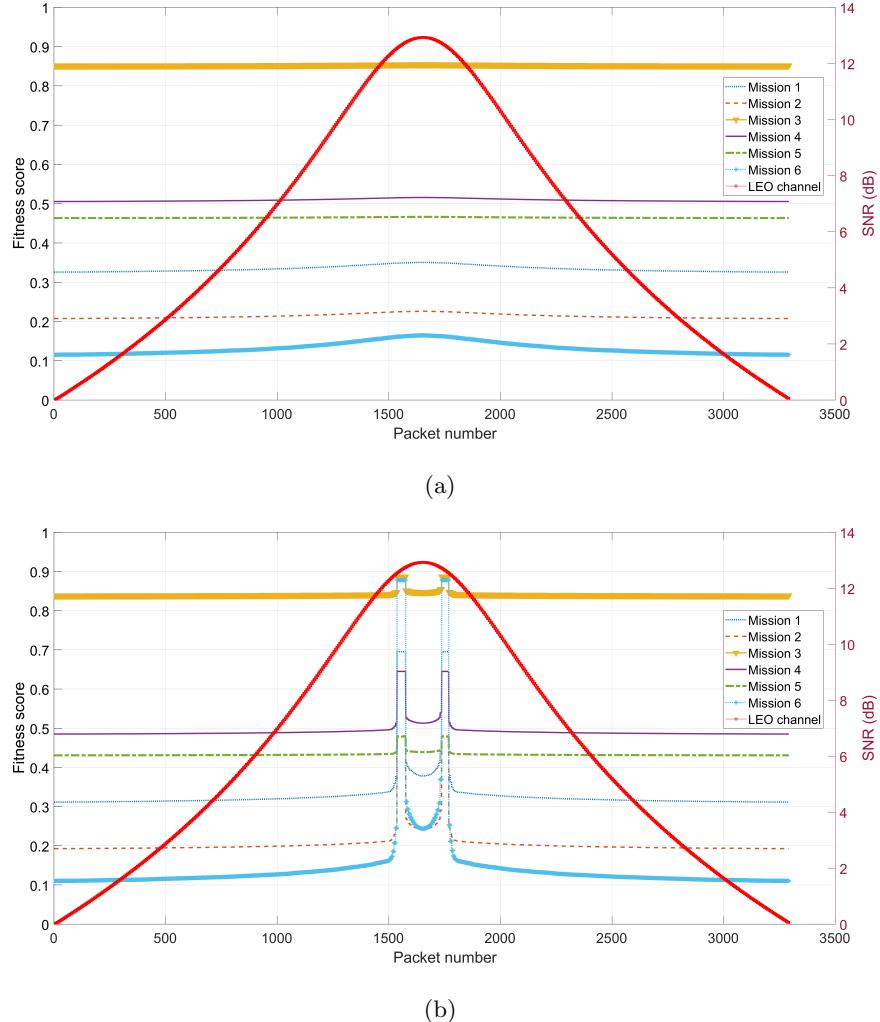


Figure 5.17: Multi-objective performance for two different actions a_A and a_B , shown in panels (a) and (b), respectively. Each action was kept fixed during the entire time the environment changed, following the SNR profile, also shown in both panels. The performance of action a_A does not seem to change too much over time for each different mission. However, the performance of action a_B changes abruptly in all missions as the environment changes.

However, for dynamically changing channels, one certain action might have different fitness score values for each different environment condition, making previous explorations' fitness scores outdated. This is well illustrated by Figure 5.17 that shows the simulated

performance values of two different actions a_A ⁶ and a_B ⁷ (action structure is described in Section 5.12), which were kept fixed while the communications channel changed. Some actions performance show minimal difference over time, as shown by the left-side panel, even for different communications missions (more details on these missions are provided in Section 5.12). On the other hand, there are actions that have performance changes over time, according to the different environment conditions. Sometimes these performance variations have a non-linear behavior, as is the case for the action performance shown on the right-side panel, with different actions showing different behaviors for different missions during different environment conditions. This fact represents one of the main drivers for the research presented by this section.

Limitation (ii) results in an exponential increase in memory size requirement due to the different performance levels for each action when experiencing different environment condition levels, when these are assumed to be discrete. The former Bellman's equation for the communications problem, Eq. (5.4), would then have the RL state and action to be function of continuous time t , computed by:

$$Q_{\text{RL}k+1}(s_k(t), a_k) = Q_{\text{RL}k}(s_k(t), a_k) + \alpha_{\text{RL}}[r_{\text{RL}k}(t) - Q_{\text{RL}k}(s_k(t), a_k)], \quad (5.18)$$

which represents the dynamic behavior of a given action's fitness score over time.

In addition to the limitations imposed by (ii), limitation (iii) adds another dimension to the performance values, which is their variation over time, making the data storage problem even worse. Thus, from the implementation perspective of online system operations it is impractical and almost impossible to save all that information.

Seeking to solve the limitations mentioned above the computation of the Q -values was dropped and, instead, the usage of another NN was proposed to predict which action should be exploited, given the current environment state level, *i.e.*, the communications channel conditions. The proposed “exploitation NN” provides the following features: (i) it solves the problem of using outdated performance values for deciding which action to exploit, (ii) it does not require storage of all action-RL state performance values for all environment

⁶ $a_A = (4.16 \cdot 10^5, 0, 5, 0.35, 32, 0.9)$

⁷ $a_B = (4.16 \cdot 10^5, 0, 5, 0.20, 32, 0.75)$

conditions, and (iii) it allows for the action performance, *i.e.*, the RL state to be decoupled from the action itself. This last feature is the most important because it allows the agent to exploit different actions in an attempt to achieve the same previous performance level while the environment dynamically changes.

Therefore, the proposed exploitation NN architecture is composed of an array of ensembles of NNs, each predicting one dimension of the multi-dimensional action vector. All exploitation NNs have the same shallow architecture and receive the same input multi-dimensional performance vector, as shown in Figure 5.18. The exploitation NN output vector is the average output of $NN2_m = 10$ parallel NNs for each output vector element. The choice of this architecture was made based on the one that showed the smallest MSE averaged over 100 simulation runs, and its performance analysis is outside the scope of this work. Each NN is composed by two fully-connected layers without bias: the hidden layer contains 20 neurons (resulting in 160 parameters) and uses the log-sigmoid transfer function, and the output layer uses the linear transfer function. All the other functions are the same as the ones used for the individual exploration NNs mentioned in Section 5.3.3.

The logic used to decide the exploitation NN input, the multi-objective performance values, are quite complex, requiring several conditional statements. Algorithm 3 describes the general operational procedure of the novel proposed hybrid architecture, called RLNN2, including the interactions between the RL, exploration NN and exploitation NN. Algorithm 4 describes the logic used to choose the exploitation NN inputs, required within Algorithm 3. All the required parameters are described and defined in Section 5.12.

Regarding the exploration method presented in Section 5.3.3, it was demonstrated that one NN ensemble can be used to explore the environment by predicting each action performance [122]. Making action performance predictions for all possible actions before every single packet transmission over-the-air might be impractical if the action space of all possible actions is continuous. Additionally, it breaks the principle of RL exploitation, in which an action should be exploited only if it has actually been explored, otherwise it is only another action exploration. However, it is feasible if the number of selected actions is discrete and there is enough processing power to perform computations before a packet transmission, as is the case assumed in this work.

Algorithm 3 RLNN2 operational routine

Require: Exploration NN (NN1) and Exploitation NN (NN2) setup, and initial parameters

$$(\bar{R}_s, \bar{E}_s, \bar{b}_s, \bar{\beta}, \bar{M}, \bar{er}, \text{NN}_{bs}, \text{NN}_{dump}, f(\varepsilon), \text{tr}_a, \text{min_good\%})$$

```

1: U  $\leftarrow$  all combinations of  $(\bar{R}_s, \bar{E}_s, \bar{b}_s, \bar{\beta}, \bar{M}, \bar{er})$ 
2: loop
3:   if  $(\bar{R}_s, \bar{E}_s, \bar{b}_s, \bar{\beta}, \bar{M}, \bar{er})$  has changed then
4:     U  $\leftarrow$  all combinations of  $(\bar{R}_s, \bar{E}_s, \bar{b}_s, \bar{\beta}, \bar{M}, \bar{er})$ 
5:   end if
6:   while NN training data buffer not full do
7:     ‘Forced exploration’: only explore
8:   end while
9:   zz  $\leftarrow$  uniform random number  $[0, 1]$ 
10:  if  $zz < f(\varepsilon) = \varepsilon_k$  then                                 $\triangleright$  with probability  $\varepsilon_k$  (Explore)
11:    Predict actions using NN1
12:    Classify actions into ‘good’ or ‘bad’ using  $\text{min}_{\text{good\%}}$ 
13:    u  $\leftarrow$  uniform random number  $[0, 1]$ 
14:    if  $u < \text{tr}_a$  then                                 $\triangleright$  Action rejection
15:      Randomly select one ‘good’ action  $a$ 
16:    else
17:      Randomly select one ‘bad’ action  $a$ 
18:    end if
19:  else                                               $\triangleright$  with probability  $1 - \varepsilon_k$  (Exploit)
20:    Predict action to be exploited using NN2
21:    Execute selected  $a$ 
22:    Measure and/or read RL states  $\bar{s}$ 
23:    Compute multi-objective fitness function  $f_{\text{obs}}(x)$ 
24:    Select next  $\text{NN2}_{\text{input}}$                                  $\triangleright$  see Algorithm 2
25:    Update NN training data buffer
26:    Build NN1 and NN2 training datasets
27:    if NN training dataset is full then
28:      Remove  $\text{NN}_{\text{dump}}$  entries from NN training dataset
29:    end if
30:  end if
31: end loop

```

Algorithm 4 RLNN2 exploitation NN input selection algorithm

Require: $\max_f_{obs} = 0$, m_{reset} , $\text{Exploit}_{\text{score}} = []$

```

1: while Training data buffer not full do
2:   if ‘Forced exploration’ then
3:      $\text{Exploit}_{\text{score}} = \max(f_{obs}(x))$ 
4:   end if
5: end while
6: if  $f_{obs} > \max\_f_{obs}$ 
7:    $\max\_f_{obs} = f_{obs}(x)$ 
8:   if  $zz < \varepsilon_k$  then                                 $\triangleright$  with probability  $\varepsilon_k$  (Explore)
9:      $\text{NN2}_{\text{input}} = \bar{s}$ 
10:    else                                          $\triangleright$  with probability  $1 - \varepsilon_k$  (Exploit)
11:      if  $f_{obs}(x) < \text{Exploit}_{\text{score}}$  then
12:        if  $\text{Exploit}_{\text{score}} - f_{obs}(x) > m_{reset} \& a_k = a_{k-1}$  then
13:          Reset NN training data buffer
14:          Start ‘Forced exploration’
15:        else if  $f_{obs}(x) < 0.9\text{Exploit}_{\text{score}}$  then
16:           $\text{NN2}_{\text{input}}$  receives  $\bar{s}$  from NN training data buffer
17:        else if  $f_{obs}(x) > 0.9\text{Exploit}_{\text{score}}$  then
18:           $\text{Exploit}_{\text{score}} = f_{obs}(x)$ 
19:        else
20:           $\text{NN2}_{\text{input}} = \text{last\_NN2}_{\text{input}}$ 
21:        end if
22:      else
23:         $\text{Exploit}_{\text{score}} = f_{obs}(x)$ 
24:         $\text{last\_NN2}_{\text{input}} = \bar{s}$ 
25:      end if
26:    end if
27:  end if

```

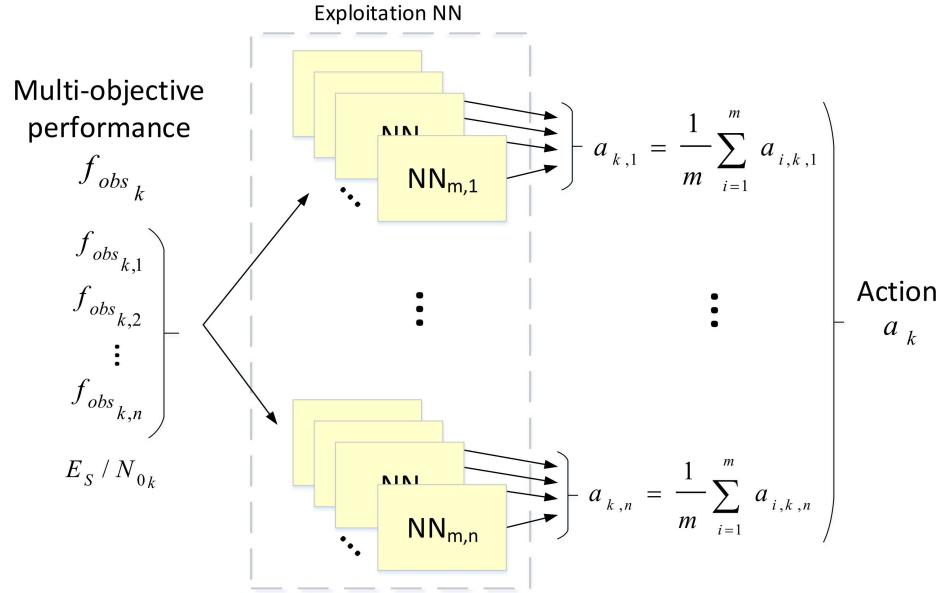


Figure 5.18: Ensemble of NN for action prediction by performance exploitation. Different sets of NN predict a specific action element by averaging their outputs. All NN receive the same multi-objective performance vector, in addition to the measured E_s/N_0 .

For the initial RLNN exploration ensemble of deep NNs, shown in Figure 5.10 to be used by the proposed RLNN2 algorithm, different E_s/N_0 values are used at the input while operating on dynamically changing environments. **Explored actions are still classified based on the action rejection threshold and rejection probability**, as shown by Figure 5.19. The Exploration NN block is an ensemble of parallel NNs, with the same architecture defined in Section 5.3.3.

5.3.7 RLNN2 Simulation Results

Before presenting any results achieved by the proposed hybrid solution RLNN2, some terms and other additional requisites that are part of the simulation need to be defined.

In order to make the systems simulations compliant with the current state-of-the-art satellite communications technology, the DVB-S2 standard [159] adaptable parameters, such as modulation scheme and encoding rate, were assumed to be used by both space-based

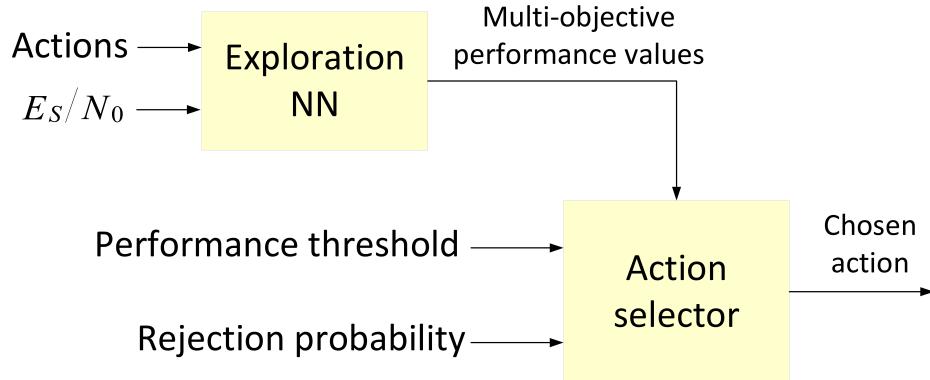


Figure 5.19: Actions are classified into ‘good’ or ‘bad’ based on their multi-objective performance values predicted by the ensemble of exploration NN using the performance threshold value. Next, a class is selected based on the ‘rejection probability’ value. The chosen action to be used on the environment is then randomly picked up from that class.

transmitter and ground-based receiver radios. Other parameters⁸ were also considered to be adaptable, with ranges described in Table 5.9.⁹ Following the operational sequence described in Algorithm 3, a multi-dimensional action composed of six parameters $\bar{a} = (R_s, E_s, b_s, \beta, M, c)$ is used by the transmitter for a transmission to the receiver over a dynamically changing environment.

For the simulations results presented below, a communications channel was assumed with the attenuation time series given in Figure 5.17 for clear sky conditions. This profile is an example of path loss, obtained from the STK orbit simulator [99], during a single LEO satellite pass at an orbit similar to the one flown by the ISS. This represents the environment variations through the natural attenuation experienced by the transmitted signal due to the satellite flight trajectory over the ground station.

Optimum action adaptation is the ultimate goal of the proposed RLNN2 algorithm. Because NN training is not perfect, NN outputs represent a best effort towards predicting an action as close as possible to the optimum action, given the currently available knowledge of the training buffers. These adaptations occur by the communications system

⁸All the simulations presented in this work assumed that the satellite transmitter amplifier operates in the close-to-linear region.

⁹Different modulation schemes use different encoding rate sets

Table 5.9: RLNN2 adaptable parameters

Parameter	Variable	Value range
Modulation order	\bar{M}	[4, 8, 16, 32]
Bits per symbol	\bar{b}_s	[2, 3, 4, 5]
Encoding rate	\bar{er}	[1/4 – 9/10]
Roll-off factor	$\bar{\beta}$	[0.2, 0.3, 0.35]
Bandwidth	\bar{BW}	[0.5 – 5] MHz
Symbol rate	\bar{R}_s	[0.41 : 0.1 : 3.7] MSamples/sec
Additional Tx E_s	\bar{E}_s	[0 : 1 : 10] dB

selecting different actions during either exploration or exploitation phases of the RL algorithm. Throughout transmissions, the RL agent seeks to optimize the fitness function score composed of conflicting multi-objective functions. Communications mission target is a vector comprised of six metrics: bit error rate (BER), throughput (Thr), bandwidth (BW), spectral efficiency (Spc_eff), additional consumed power (Pwr.con), and power efficiency (Pwr_eff), with all values scaled to the range of [0, 1]. The additional power is assumed to be a variable power amount added to the constant power rated for the worst case condition already used by the satellite.

For proof-of-concept purposes, the multi-objective considered in this work is composed by functions that compute the throughput by:

$$f_{\text{Thr}} = R_s b_s er, \quad (5.19)$$

the bandwidth is computed by:

$$f_{\text{BW}} = R_s(1 + \beta), \quad (5.20)$$

the spectral efficiency is computed by:

$$f_{\text{Spc.eff}} = b_s er / (1 + \beta), \quad (5.21)$$

the power efficiency is computed by:

$$f_{\text{Pwr_eff}} = (b_s er) / [10^{[(E_s/N_0)/10]} R_s], \quad (5.22)$$

and the additional power consumed is computed by:

$$f_{\text{Pwr_con}} = E_s R_s. \quad (5.23)$$

The BER¹⁰ is computed by interpolating the curve functions acquired by simulations of the DVB-S2 communications system provided by MATLAB [98]. Although MATLAB provides the end-to-end modulators, demodulators, LDPC, and BCH encoders, both 16-APSK and 32-APSK modulation schemes constellations and all other encoding rates had to be added to the simulator. All the frames were assumed to have a length of 64,800 bits, representing the DVB-S2 long-frame type.

Then, the reward r_k , previously defined by function g is computed as the multi-objective fitness function $f_{\text{obs}}(x)$, given by the weighted sum computed by:

$$\begin{aligned} f_{\text{obs}}(x) = & w_1 f_{\text{Thrp}} + w_2 f_{\text{BER}} + w_3 f_{\text{BW}} + \\ & w_4 f_{\text{Spc_eff}} + w_5 f_{\text{Pwr_eff}} + w_6 f_{\text{Pwr_con}}, \end{aligned} \quad (5.24)$$

where w_i are the weights for each performance parameter according to the communications mission selected, defined by the user. Examples of mission profiles used in the simulations are given by Table 5.10, and x represents all w_i values and all the other variables that the multi-objective functions depend on.

Algorithm 3 defines the usage of a NN training data buffer, containing a total of NN_{bs} entries each composed by the action, the RL states s , and the fitness function value, collected during exploration. Whenever this buffer gets filled up both exploration NN (NN1) and exploitation NN (NN2) are re-trained and the oldest NN_{dump} entries are removed from the buffer.

Before each packet transmission the RL agent decides between exploration and exploitation by computing the exploration probability value ε_k through the exploration probability function $f(\varepsilon)$. Then, if a random number z is less than ε_k , the agent virtually explores

¹⁰Plots of BER curves and MATLAB functions are available at [162].

Table 5.10: Weight values for communications missions

Mission	w_1	w_2	w_3	w_4	w_5	w_6
1 - Launch/reentry	0.2	0.4	0.1	0.1	0.1	0.1
2 - Multimedia	0.5	0.3	0.05	0.05	0.05	0.05
3 - Power saving	0.05	0.05	0.05	0.05	0.3	0.5
4 - Normal	1/6	1/6	1/6	1/6	1/6	1/6
5 - Cooperation	0.05	0.05	0.4	0.4	0.05	0.05
6 - Emergency	0.1	0.8	0.025	0.025	0.025	0.025

the environment through NN1, draws another random value u , and checks it against the “action rejection” value tr_a in order to pick an action from either the good or bad sets. These sets contain virtually explored actions classified by their respective fitness score based on a threshold value defined by the percentage of the maximum fitness score achieved during virtual exploration, defined by $\text{min}_{\text{good}\%}$.

For the simulation results presented in this section, it was assumed $\text{NN}_{bs} = 200$, $\text{NN}_{\text{dump}} = 50$, $\text{tr}_a = 0.95$, $\text{min}\% = 0.9$. $f(\varepsilon)$ varies with time steps k , assumed to be equal to $\varepsilon_k = 1/k$, starting at 1 and reset every time $\varepsilon_k = 10^{-4}$.

Algorithm 4 defines the logic to select the input for NN2 through two main mechanisms: slow recovery and fast recovery. The former refers to the exploitation of performance values stored on the NN training buffer in a descending order of fitness score values. The latter resets the NN training buffer, which results in re-training of both NNs after enough data is collected while in the “forced exploration” mode. Since the fast recovery forces continuous exploration for the duration of NN_{bs} packets, a threshold value m_{reset} controls when it is triggered. Although it is an operator parameter, the results consider $m_{\text{reset}} = 0.5$ for all missions.

Simulations were run for all the missions presented in Table 5.10. Because the time duration required by each iteration simulation for each mission was too long (simulation runs varied between 4 and 8 hours each, a total of 320 simulation hours), the performance

results were averaged over 10 runs for each mission.

For the interested reader, Figure 5.20 provides examples of fitness score time series for each mission, with the duration of a LEO orbit mentioned in previous sections, *i.e.*, 512 seconds. These time series include performance values during both exploration and exploitation periods.

5.3.8 Remarks on RLNN2 Performance

As shown in Figure 5.21, panels (a) to (f) present the normalized packet count distribution over fitness score values for Missions 1 to 6, respectively. Even though each mission has a different profile, which is function of the chosen weight values, in all panels the performance is concentrated on high fitness score values. Since the goal is to have the majority of packets experiencing the highest fitness score as possible, the plots show that the RLNN2 was able to select good actions, learning the relationship between rewards and actions while the channel dynamically changed.

In order to provide a better understanding of the accuracy achieved by the proposed solution, the simulations' fitness score values of each mission were compared against their respective ideal performance values. These ideal values were computed by evaluating the performance of all actions at each channel profile value through exhaustive search. The error between the performance achieved by the proposed algorithm and the ideal solution is shown as accuracy distributions in Figure 5.22. For all missions, the RLNN2 algorithm resulted in the majority of packets being concentrated around very low fitness score error values, *i.e.*, at high accuracy values, as desired. Mission 3, shown in panel (c) had the best performance in terms of accuracy with a minimum accuracy percentage value of 91%, and an average of 80% of packets experiencing an accuracy of 99%. Mission 2 had the worst accuracy performance, where some packets experienced accuracy values as low as 61%, and less than 40% of packets experienced accuracy values of 99%. These accuracy distributions are expected to serve as benchmarks for future research on cognitive engines for space-based communications systems.

Another metric used to measure the performance of both the average fitness score and its error against the ideal case is the integral value of the areas under the distribution

curves shown in Figures 5.21 and 5.22, shown in Table 5.11. In terms of performance, the average integrals in Figure 5.21 present a minimum equal to 0.72 for Mission 4, and a maximum equal to 0.88 for Mission 6. In terms of accuracy, the integral values for the error distributions in Figure 5.22 has a minimum equal to 0.01 for Mission 3 and a maximum equal to 0.06 for Mission 2. Thus, because of the values chosen for the fitness score function weights, each mission has its own ideal performance profile, and the error becomes a better metric to assess the efficiency of the proposed solution in choosing good actions during each different channel condition.

Regarding the ensemble size, Table 5.11 also shows the integral values for performance and the error between that performance and its ideal values when the ensemble size is equal to one for both exploration and exploitation NNs $NN1_m = NN2_m = 1$. This ensemble size choice directly impacts on the processing time of both training and prediction for NN, with the former being the most sensitive for the case being studied. Although, for simulations running on MATLAB, the processing time was decreased by 12 times, when implemented in dedicated hardware, such as FPGA, this gain in terms of execution time has yet to be defined, and is out of the scope of this work. This analysis was aimed at showing the trade-offs involved with accelerating the simulation time versus the performance errors.

Therefore, for all missions, the integral of the performance distribution is similar when a larger ensemble size is chosen. However, with the exception of Mission 6, the error increased for all other mission performances. Mission 3 experienced an error of almost 6 times higher, with all other errors being around 2 times higher or more.

It is worth noting that future research on this topic of CE space-based communications systems might need to compare against real-world performance. Comparisons between real-world experiments are expected to be performed against the accuracy for the missions profiles suggested in this dissertation. However, if any additional limitations on the number of adaptable parameters or number of goals change, new simulations need to be run and the accuracy for that specific case needs to be registered. Therefore, the process is the same and the RLNN2 general architecture to be used is the same, with the caveat that sizes of NNs ensembles, number of array elements, inputs, and outputs must match the ones being considered in the experiment.

Table 5.11: Integral values for average performance and error distribution curves

Mission	Integral values			
	$NN1_m = 20, NN2_m = 10$		$NN1_m = NN2_m = 1$	
	Performance	Error	Performance	Error
1 - Launch/reentry	0.7605	0.0525	0.7631	0.1278
2 - Multimedia	0.7878	0.0643	0.7643	0.158
3 - Power saving	0.8316	0.0177	0.7894	0.1055
4 - Normal	0.7237	0.0491	0.7049	0.0927
5 - Cooperation	0.8382	0.0366	0.7834	0.0651
6 - Emergency	0.8836	0.0387	0.8931	0.044

5.3.9 RLNN2 Performance Trade-off

As mentioned above, the ideal performance achieved by exhausted search establishes an upper-bound performance. The simulation results in Section 5.21 verifies that the proposed solution is compliant with this bound, since the results never achieve a performance higher than the ideal. Besides the limitations that inspired the Exploitation NN proposal, mentioned in Section 5.3.6, using NN for choosing actions to be exploited provides flexibility instead of performance improvements, when compared to systems that do not use NN. Systems without Exploration NN can achieve similar performance if the granularity provided by the state-action mapping is high enough (assuming there is enough memory to store this information, not to mention sufficient memory access time), and the exploration probability is also high enough.

However, the cost for training and predicting using NN provides flexibility in terms of relieving the requirements of memory size by allowing the state-action mapping to be performed for continuous action and/or state dimensions, a more realistic approach to the real world. As a byproduct of using NN for exploitation, a fixed memory size learning

capability is achieved. Although the research presented in this dissertation used the learned knowledge to predict actions, future communication systems could potentially use it to achieve more optimized performance or other tasks.

In addition to that, the usage of NN also provides scalability. Future implementations can extend the Exploitation NN design to deal with larger scales of action parameters available, and more state variables monitored, trading off memory size and access time for NN training and prediction processing requirements. This dissertation shows that this trade-off is technically feasible by providing algorithms to implement the proposed solution, and error benchmarks against ideal cases, for future research reference.

5.3.10 Hardware Implementation Considerations

The simulation results presented in this dissertation were executed using MATLAB scripts, a very high level language that is not optimized for performance. For instance, the real-time duration of the simulated missions was 15 minutes, but the total time spent on the RLNN2 simulations were more than four hours using an Intel-based multi-core processor being tasked by many other applications at the same time.

Since the proposed algorithms are intended to be deployed in real-world applications they are expected to run in real-time. For example, for a network experiencing round-trip time (RTT) delay of 30 ms, adaptation can then take place anytime after this interval, but as fast as possible, *e.g.*, for a parameter update rate of 40 ms, the computation of the new parameters should be done in the next 10 ms or so after the RTT.

As mentioned in Section 5.3.8, this execution time can be decreased, while keeping the same hardware, by decreasing the size of the NN ensembles, trading-off latency for accuracy of results. Since this kind of trade-off is not recommended, future space-based communication systems are expected to have enough computing power in order to keep the latency of performing this processing close to an acceptable limit.

In addition, the current radio receivers in space are very limited in processing power, since they were not designed to deal with NN training or more advanced processing tasks. This is one of the main reasons NASA has been flying the SCaN Testbed and why this dissertation is proposing this algorithm, so future communication systems will be cognition-

enabled. Thus, it might take a while for the space industry to start making available more powerful radiation-hardened FPGAs on-board spacecraft.

One major recommendation is the usage of real-time operating systems in order to decrease the variability of task execution time for the CE, and to perform the processing on GPUs, especially the training and predictions for the NNs, given their great capability of dealing with floating point arithmetics.

Regarding currently available technology to allow experimentation, by the time of the writing of this dissertation, there was a limitation in the radio waveforms available on the terminal modems and the ones implemented on the SCaN Testbed. Because the current implementation uses the DVB-S2 standard, it does not allow full adaptability to be done on all the parameters as proposed by the RLNN2 algorithm, and that is expected to be available for space-based radio systems in the future.

5.4 Chapter Summary

This chapter provided algorithms that constitute the cognitive engine core. A novel RL algorithm for satellite-based communications concept was proposed, and simulations presented promising improvements in the multi-objective performance when compared to a totally random algorithm. Next, the proposed RLNN algorithm introduced the virtual exploration concept, reducing costs of testing actions in the real-world environment. Finally, the RLNN2 was presented. It relies on exploring and exploring RL actions using NNs for dynamically changing channels. Simulation results showed that very low performance errors can be achieved when compared to the ideal solutions.

It is worth noting here that, to the best of the author's knowledge, there is no similar approach in the literature related to space communications systems capable of finding multi-dimensional actions to achieve a multi-dimensional objective while the environment changes dynamically. Thus, it would not be fair to compare the proposed algorithms against individual approaches that aim to adapt only individual parameters, such as ACM. Other approaches that operate in batch mode, as GA, would cause extremely high latency to the system, as compared to the incremental approach of RLNN2. Therefore, the results and er-

ror distributions presented here considered the ideal solutions and allow for future research on CE for space communications systems to be compared against.

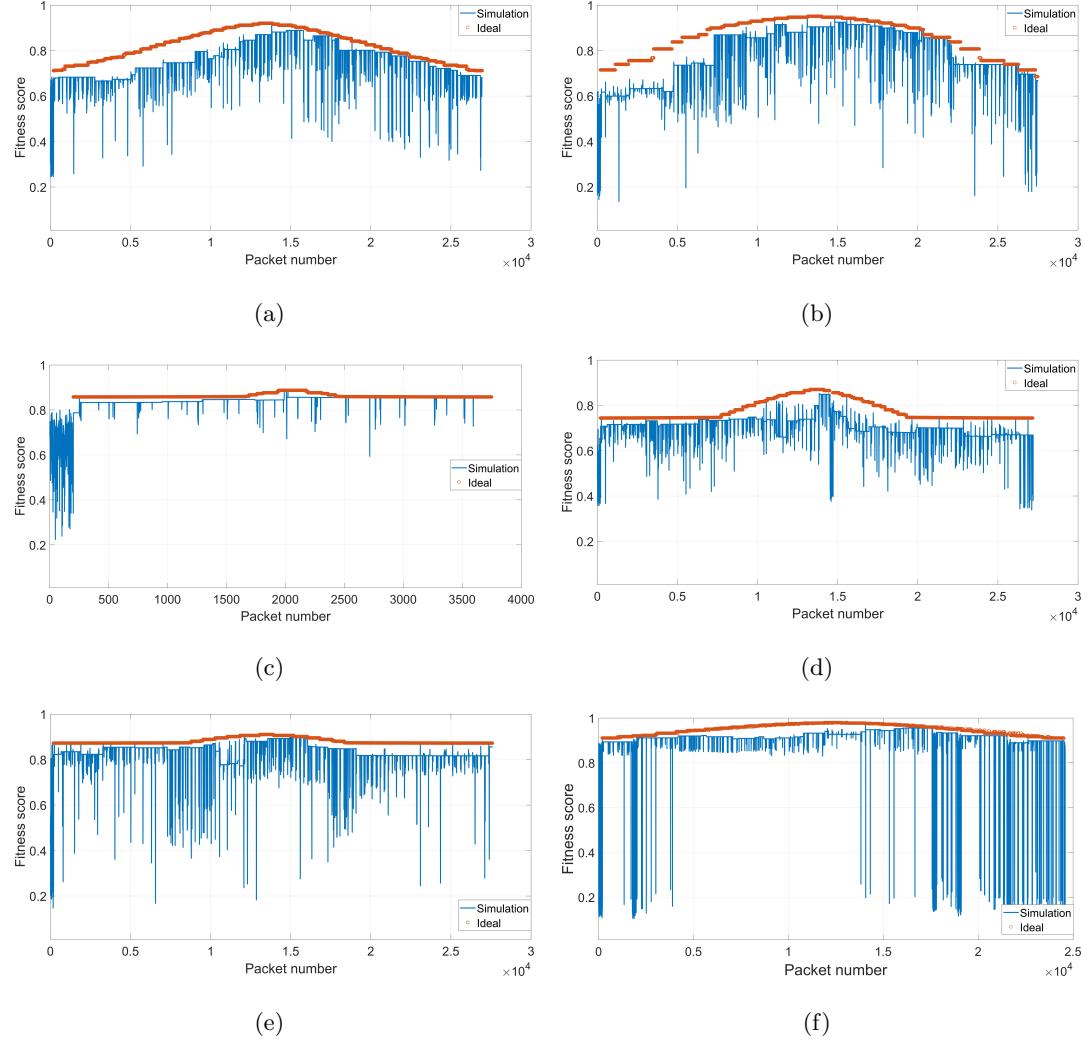


Figure 5.20: Examples of fitness score value time series per packet. Panels (a) to (f) show performances for Missions 1 to 6, respectively. In all missions, the majority of simulation packets experienced performances close to the ideal. Constant score levels represent exploitation, while spikes below and above them represent exploration.

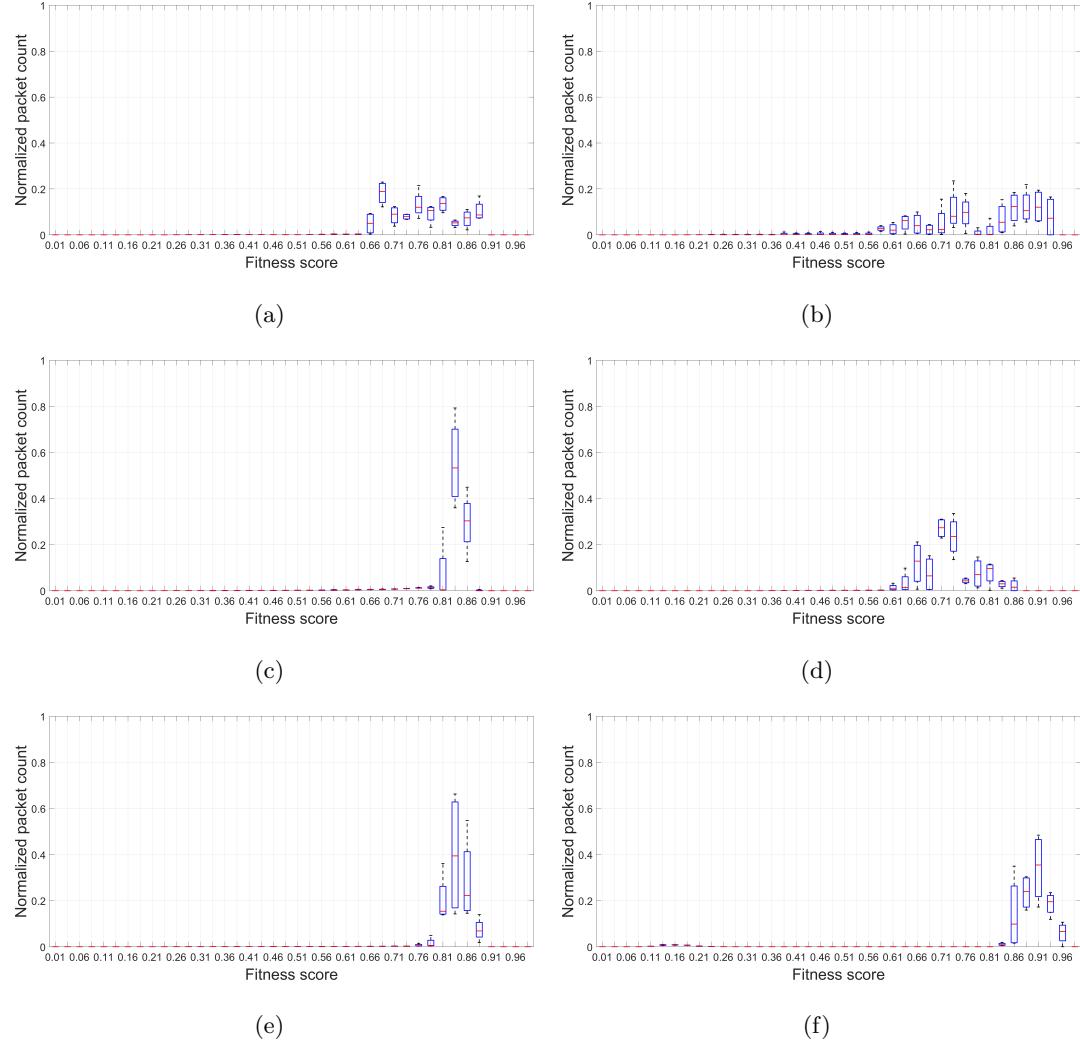


Figure 5.21: Boxplots of normalized packet count distribution over fitness score values obtained using RLNN2. Panels (a) to (f) represent performances for Missions 1 to 6. More packets concentrated around larger score values is better.

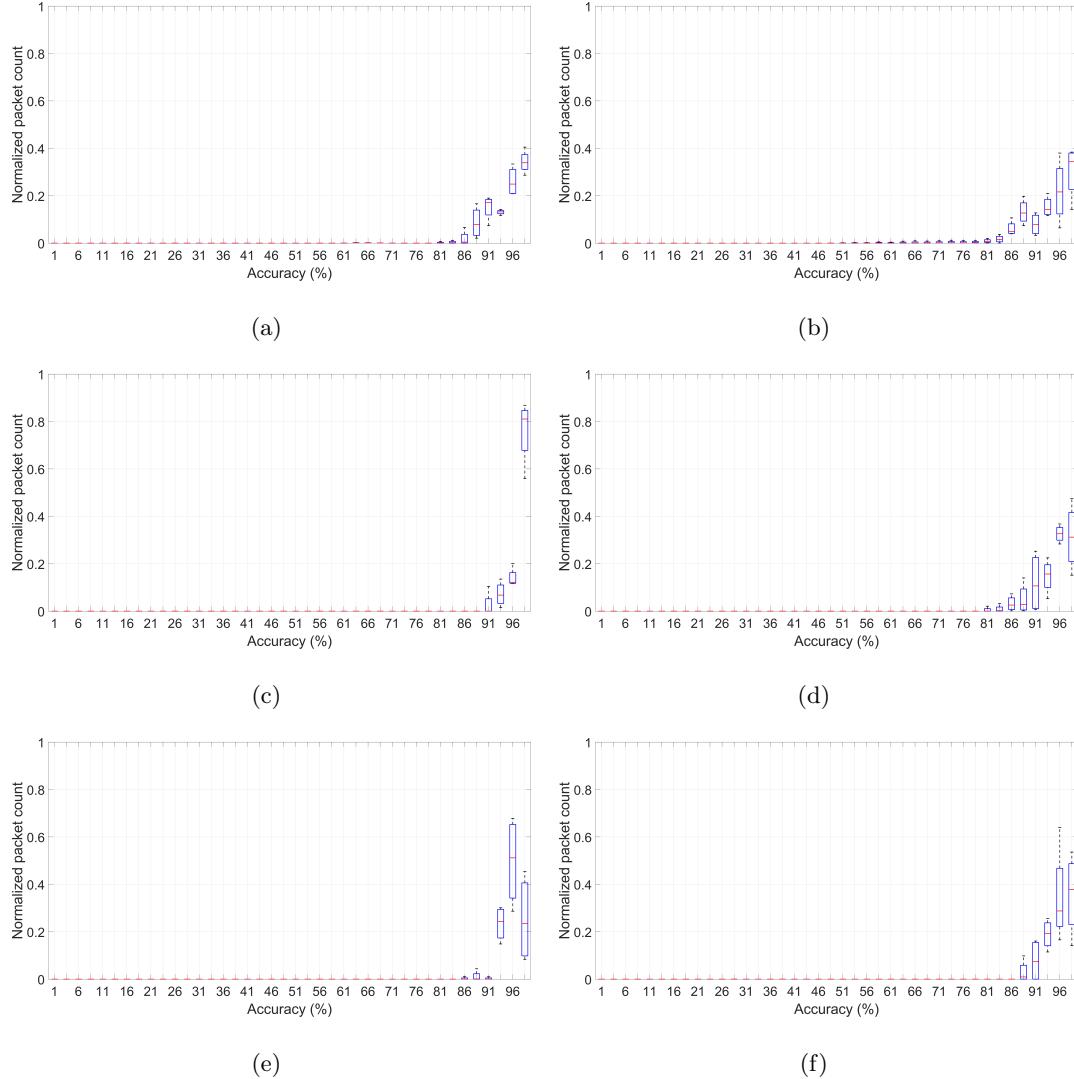


Figure 5.22: Boxplots of normalized packet count distribution over fitness score error values between proposed RLNN2 and ideal solution. Panels (a) to (f) represent performances for Missions 1 to 6. More packets concentrated around lower error score values is better.

Chapter 6

Conclusion and Future Work

In this dissertation, several research gaps were covered by providing more realistic satellite channel communications models and time series synthesizers. Also, leveraging these models and synthetic data generators, channel state detectors for fixed and mobile ground receivers at Ka-band were proposed, aimed to increase the QoS for the user during rainy conditions. Finally, a novel cognitive engine design for future space-based communications systems was proposed for both fixed and dynamically changing channel conditions. Below, the main achievements described throughout this study, and future work are summarized.

6.1 Research Achievements

Based on the current state-of-the-art, several research gaps were identified. Each gap represented specific challenges that are addressed by this dissertation through the following contributions:

- **A new Rician factor equation for more realistic channel models:** A novel Rician factor equation was derived coupling it directly with the reflection coefficient of the terrain's material constitution and electrical properties. The study case considered the application of this new derivation on computing the BER experienced by an airborne receiver communicating with a GEO satellite during different levels of ionospheric scintillation events. The results showed the potential of the terrain structure

in affecting the communications performance.

- **Novel attenuation time series synthesizers:** Rain time series generators for LEO orbits, and LMS scenarios for satellite-based communication systems were proposed. These algorithms considered operation at Ka-band. The novel method for LEO orbits enables the recent ITU-R P.1853 recommendation for GEO to be applied at any elevation angle profile desired. For mobile nodes experiencing rain attenuation at Ka-band, the current LMS channel model had the rain attenuation synthesizer incorporated.
- **QoS improvements using Kalman filters:** Leveraging the novel attenuation time series synthesizers proposed by this dissertation, one of its first usages was in the design of channel state detectors and predictors. It was showed that, during rain attenuation at Ka-band, ACM schemes can decrease the number of modcod switching by using Kalman filtered signals., which keeps the service quality more steady for the receiver. For mobile receivers experiencing rain attenuation at Ka-band in urban environments, IMM filters are more accurate in detecting signal shadowing when compared to the current state-of-art methods.
- **A novel cognitive engine for autonomous space communications:** A cognitive communications algorithm for future aerospace applications was proposed. It leverages hybrid machine learning algorithms emphasized to satellite communications dynamic channels. Its performance is evaluated for different communications missions, and it is described in terms of multi-objective performance scores. Comparison of simulated performance results against optimal values show that the proposed approach is feasible, and qualifies for in-orbit experiments.

6.2 Future Work

The contributions presented in this dissertation provides solutions to some research problems related to satellite communications. In addition, novel algorithms were also proposed that enable further investigations to be performed that may support the development of future communications systems. Several future works are enabled by the research discussed

in this dissertation, with the most important ones being the following:

- The proposed Rician K -factor derivation enables future research to evaluate more complex communication scenarios, with receivers moving at different speeds experiencing signal multipath reflections from different materials surface compositions, as well as, various terrain conditions.
- Statistical research may be done through the analysis of rain attenuation time series experienced by LEO for receivers located at any region of the planet during any elevation angle profiles for different passings.
- Further investigations might study the statistics of satellite downlink signals in various urban conditions while being affected by atmospheric weather, such as attenuation caused by rain at Ka-band.
- Next-generation satellite-based networking systems may rely on IMM-based solutions to detect signal blockages and trigger node relay mechanisms.
- The RLNN2 could be investigated from a Generative Adversarial Nets (GANs) perspective. Since the exploration NNs outputs become inputs to the exploitation NNs, and vice-versa. One of the interests in this method resides on the possibility of using it as a technique to generate more training data.
- Future cognitive engines for space-based communications systems will enable operations during critical mission phases, for instance, during launch and re-entry and emergency modes. An autonomous radio scheduler is a very important feature required future exploration missions, employed by either orbiting satellites or spacecraft, when no immediate human intervention may be possible. Thus, the proposed CE is among the first ones to qualify for in-orbit experimentation, defining the path for future research on more robust cognitive communications systems for space applications.

Appendix A:

Kalman Filter Matrices

We model the true attenuation value x based on derivations found in the open literature [163], [164]. Our system observable state x_t is a past value plus or minus a certain amount dependent on the estimated slope Δx :

$$x_t = x_{t-1} + \Delta x. \quad (6.1)$$

We can measure only the state variable, but not its slope. On the other hand, we can estimate both. Thus, we define our continuous linear system by:

$$\dot{x} = Ax. \quad (6.2)$$

The state matrix \hat{X} is given by:

$$\hat{X}_t = \begin{pmatrix} \hat{x}_t & \frac{d\hat{x}_t}{dt} \end{pmatrix}^T, \quad (6.3)$$

and the A matrix in continuous mode is:

$$A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}. \quad (6.4)$$

While in discrete mode, we have:

$$F_t = e^{\Delta t A} = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix}. \quad (6.5)$$

with Δt being the discrete sampling interval. $\Delta t = 1$ for filtering and $\Delta t = k$ for predictions $k -$ steps ahead. The measurement consists only of the noisy SNR at the receiver, and as mentioned above, the slope will be computed on-line. The measurement mapping matrix H is given by:

$$H = \begin{pmatrix} 1 & 0 \end{pmatrix}; \quad (6.6)$$

and it maps the values from y_t , the measurement vector, given by:

$$y_t = \begin{pmatrix} y_t & 0 \end{pmatrix}^T \quad (6.7)$$

that are used by the filter. The variance matrix Q was modeled in the same way as in [23] for continuous case:

$$Q = \begin{pmatrix} 0 & 0 \\ 0 & q \end{pmatrix}, \quad (6.8)$$

where q is the noise in the added portion Δx , and for discrete case:

$$Q = \int_0^{\Delta t} e^{A\tau} Q e^{A^T \tau} d\tau = \begin{pmatrix} \frac{\Delta t^3 q}{3} & \frac{\Delta t^2 q}{2} \\ \frac{\Delta t^2 q}{2} & \Delta t q \end{pmatrix}. \quad (6.9)$$

The covariance matrix Z_t has the form:

$$Z = \begin{pmatrix} z_v & z_c \\ z_c & z_v \end{pmatrix}, \quad (6.10)$$

where z_v is all the diagonal elements representing the variance and z_c are the elements representing the covariance between the measurement and the rate it is changing.

Bibliography

- [1] P. V. R. Ferreira and A. M. Wyglinski, “Performance analysis of UHF mobile satellite communication system experiencing ionospheric scintillation and terrestrial multipath fading,” *82nd 2015 IEEE Vehicular Technology Conference*, 2015.
- [2] ——, *Satellite-Communications: Nonlinear weighted two dimensional interpolation of rain attenuation time series for LEO satellites based on elevation angle profile using ITU-R P.1853 v1.1*. GitHub, 2016, doi: 10.5281/zenodo.44696.
- [3] “Rec. ITU-R P.838-3 - Specific attenuation model for rain for use in prediction methods,” International Telecommunication Union, Tech. Rep., 2005.
- [4] P. V. R. Ferreira, R. Paffenroth, A. M. Wyglinski, T. M. Hackett, S. G. Bilén, R. C. Reinhart, and D. J. Mortensen, “Multi-objective reinforcement learning for cognitive radio-based satellite communications,” *34th AIAA International Communications Satellite Systems Conference*, 2016.
- [5] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, “Mastering the game of Go with deep neural networks and tree search,” *Nature*, pp. 484–489, January 2016.
- [6] “Luxembourg Invests €25 Million in Asteroid Mining,” IEEE Spectrum. [Online]. Available: <http://spectrum.ieee.org/tech-talk/aerospace/space-flight/luxembourg-invests-25-million-in-asteroid-mining>

-
- [7] Z. Ye and E. Satorius, “Channel modeling and simulation for mobile user objective systems - part I: Flat scintillation and fading,” *Proceedings of IEEE International Conference on Communications*, 2003.
 - [8] ——, “Channel modeling and simulation for mobile user objective systems - part II: Selective scintillation and terrestrial multipath fading,” *Proceedings of IEEE International Conference on Communications*, 2003.
 - [9] R. Kumar, “Scattering functions for fading channels with ionospheric scintillation and terrestrial multipath,” *IEEE Aerospace Conference Proceedings*, 2005.
 - [10] J. Hant, V. Lin, and P. Anderson, “Verification of satellite-channel simulator with scintillation, terrestrial multipath and shadowing effects,” *Proceedings of IEEE Aerospace Conference*, 2005.
 - [11] “NASA ACTS Program,” National Aeronautics and Space Administration. [Online]. Available: <http://www.nasa.gov/centers/glenn/about/fs13grc.html>
 - [12] “RF Propagation Database,” National Aeronautics and Space Administration. [Online]. Available: <https://propagation.grc.nasa.gov/background/what-are-we-measuring>
 - [13] C. D. Hughes, “The OLYMPUS satellite programme,” *IEE Colloquium on Results of Experiments Using the Olympus Satellite*, p. 5, December 1991.
 - [14] H. Fukuchi, N. Abe, T. Takahashi, and T. Asai, “Ka-band satellite communication experiments and rain attenuation measurements using WINDS,” in *7th International Conference on Information, Communications and Signal Processing*, Dec 2009, pp. 1–4.
 - [15] T. Kostulski and S. Reisenfeld, “Spectral analysis of experimental Ka-band propagation measurements over the Australian LEO microsatellite FedSat,” in *Personal Satellite Services*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Springer Berlin Heidelberg, 2009, vol. 15, pp. 41–48.

-
- [16] X. Boulanger, B. Gabard, L. Casadebaig, and L. Castanet, “Four years of total attenuation statistics of Earth-space propagation experiments at Ka-band in Toulouse.” *IEEE Transactions on Antennas and Propagation*, vol. 63, no. 5, pp. 2203–2214, February 2015.
 - [17] A. Goldsmith and S.-G. Chua, “Adaptive coded modulation for fading channels,” *IEEE Transactions on Communications*, vol. 46, no. 5, pp. 595–602, May 1998.
 - [18] D. Tarchi, G. Corazza, and A. Vanelli-Coralli, “Adaptive coding and modulation techniques for next generation hand-held mobile satellite communications,” in *IEEE International Conference on Communications (ICC), 2013*, June 2013, pp. 4504–4508.
 - [19] M. Emmelmann and H. Bischl, “An adaptive MAC layer protocol for ATM-based LEO satellite networks,” in *IEEE 58th Vehicular Technology Conference, 2003. VTC 2003-Fall.*, vol. 4, Oct 2003, pp. 2698–2702.
 - [20] K. Butchart and R. Braun, “An adaptive modulation scheme for low Earth orbit satellites,” in *Proceedings of the 1998 South African Symposium on Communications and Signal Processing, 1998. (COMSIG '98)*, Sep 1998, pp. 43–46.
 - [21] “ETSI EN 302 307-1: 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,” ETSI, Tech. Rep., 2014.
 - [22] “ETSI EN 302 307-2: 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 2: DVB-S2 Extensions (DVB-S2X),” ETSI, Tech. Rep., 2014.
 - [23] G. Welch and G. Bishop, “An introduction to the Kalman filter,” in *Annual Conference on Computer Graphics and Interactive Techniques in Computer Graphics*, Los Angeles, CA, August 2001.

-
- [24] L. R. Leon Ojeda, A. Y. Kibangou, and C. Canudas De Wit, “Adaptive Kalman filtering for multi-step ahead traffic flow prediction,” in *2013 American Control Conference (ACC 2013)*, Washington, USA, 2013.
 - [25] R. Chrobok, O. Kaumann, J. Wahle, and M. Schreckenberg, “Different methods of traffic forecast based on real data,” *European Journal of Operational Research*, vol. 155, no. 3, pp. 558–568, 2004.
 - [26] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, “Interacting Multiple Model methods in target tracking: a survey,” *IEEE Transactions on Aerospace and Electronic systems*, vol. 34, pp. 103–123, January 1998.
 - [27] B. Feng, F. Ma, M. Fu, and C. Yang, “Real-time state estimator without noise covariance matrices knowledge–fast minimum norm filtering algorithm.” *IET Control Theory and Applications*, vol. 9, pp. 1422–1432, June 2015.
 - [28] T. Kirubarajan and Y. Bar-Shalom, “Kalman filter versus IMM estimator: when do we need the latter?” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, pp. 1452–1457, October 2003.
 - [29] B. Feng, M. Fu, H. Ma, Y. Xia, and B. Wang, “Kalman filter with recursive covariance estimation - sequentially estimating process noise covariance.” *IEEE Transactions on Industrial Electronics*, vol. 61, pp. 6253–6263, November 2014.
 - [30] P. Vergez, L. Sauter, and S. Dahlke, “An improved Kalman filter for satellite orbit predictions,” *The Journal of the Astronautical Sciences*, vol. 52, no. 3, September 2004.
 - [31] A. Wannasarnmaytha, S. Hara, and N. Morinaga, “Two-step Kalman-filter-based AFC for direct conversion-type receiver in LEO satellite communications,” *IEEE Transactions on Vehicular Technology*, vol. 49, no. 1, pp. 246–253, January 2000.
 - [32] J. Vil-Valls, P. Closas, and C. Fernndez-Prades, “On the identifiability of noise statistics and adaptive KF design for robust GNSS carrier tracking,” *2015 IEEE Aerospace Conference*, March 2015.

- [33] J.-H. Won, “A novel adaptive digital phase-lock-loop for modern digital GNSS receivers,” *IEEE Communications Letters*, vol. 18, no. 1, pp. 46–49, January 2014.
- [34] J. Vil-Valls, P. Closas, C. Fernndez-Prades, J. A. Lpez-Salcedo, and G. Seco-Granados, “Adaptive GNSS carrier tracking under ionospheric scintillation: Estimation vs. mitigation,” *IEEE Communications Letters*, vol. 19, no. 6, pp. 961–964, March 2015.
- [35] A. H. Sayed, *Adaptive Filters*. Wiley, 2008.
- [36] R. R. L. Jr., *Kalman and Bayesian Filters in Python*, 2nd ed., 2015.
- [37] L. A. Johnston and V. Krishnamurthy, “An improvement to the Interacting Multiple Model algorithm,” *IEEE Transactions on Signal Processing*, vol. 49, pp. 2909–2923, December 2001.
- [38] T. Sathyan and T. Kirubarajan, “Markov-jump-system-based secure chaotic communication,” *IEEE Transactions on Circuits and Systems*, vol. 53, pp. 1597–1609, July 2006.
- [39] J. F. Kurose and K. W. Ross, *Computer Networking: A top-down approach*. Pearson, 2013.
- [40] S. Chen, T. R. Newman, J. B. Evans, and A. M. Wyglinski, “Genetic algorithm-based optimization for cognitive radio networks,” *IEEE Sarnoff Symposium*, 2010.
- [41] E. Hossain, D. Niyato, and D. I. Kim, “Evolution and future trends of research in cognitive radio: A contemporary survey,” *Wiley Wireless Communications and Mobile Computing*, vol. 15, pp. 1530–1564, 2013.
- [42] A. He, K. K. Bae, T. R. Newman, J. Gaeddert, K. Kim, R. Menon, L. Morales-Tirado, J. Neel, Y. Zhao, J. H. Reed, and W. H. Tranter, “A survey of artificial intelligence for cognitive radios,” *IEEE Transactions on Vehicular Technology*, vol. 59, no. 4, 2010.
- [43] D. M. Roijers, P. Vamplew, S. Whiteson, and R. Dazeley, “A survey of multi-objective

- sequential decision-making,” *Journal of Artificial Intelligence Research*, vol. 48, pp. 67–113, 2013.
- [44] N. Abbas, Y. Nasser, and K. E. Ahmad, “Recent advances on artificial intelligence and learning techniques in cognitive radio networks,” *EURASIP Journal on Wireless Communications and Networking*, 2015.
- [45] L. Y. Bkassiny, M. and S. K. Jayaweera, “A survey on machine-learning techniques in cognitive radios,” *IEEE Communications Surveys and Tutorials*, vol. 15, no. 3, pp. 1136–1159, 2013.
- [46] K. Kotobi, P. B. Mainwaring, C. S. Tucker, and S. G. Bilén, “Data-Throughput Enhancement Using Data Mining-Informed Cognitive Radio,” *Electronics MDPI*, 2015.
- [47] K. Kotobi and S. G. Bilén, “Short Paper: Introduction of Vigilante Players in Cognitive Networks with Moving Greedy Players,” *2015 IEEE 82nd Vehicular Technology Conference (VTC Fall)*, 2015.
- [48] S. Chatzinotas, B. Ottersten, and R. D. Gaudenzi, *Cooperative and Cognitive Satellite Systems*. Elsevier, Academic Press, 2015.
- [49] S. K. Sharma, S. Maleki, S. Chatzinotas, and J. Grotz, “Implementation issues of cognitive radio techniques for Ka-band (17.7–19.7 GHz) SatComs,” *7th Advanced Satellite Multimedia Systems Conference and the 13th Signal Processing for Space Communications Workshop*, 2014.
- [50] K. Kotobi, P. B. Mainwaring, C. S. Tucker, and S. G. Bilén, “Puzzle-based Auction Mechanism for Spectrum Sharing in Cognitive Radio Networks,” *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2016.
- [51] M. Kutner, C. Nachtsheim, and J. Neter, *Applied Linear Regression Models 4th Ed.* McGraw Hill, 2004.
- [52] C. Chatfield, *The Analysis of Time Series*. Chapman & Hall/CRC, 2004.

-
- [53] O. C. Ibe, *Fundamentals of Applied Probability and Random Processes*. Elsevier, 2005.
 - [54] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering*. Addison-Wesley, 1994.
 - [55] A. C. Clarke, “Peacetime Uses for V2,” *Wireless World*, p. 58, February 1945.
 - [56] ——, “Extra-Terrestrial Relays,” *Wireless World*, pp. 305–308, October 1945.
 - [57] G. Maral and M. Bousquet, *Satellite Communications Systems, Techniques and Technology*. John Wiley and Sons, 2009.
 - [58] “Global Positioning System,” National Coordination Office for Space-Based Positioning, Navigation, and Timing. [Online]. Available: <http://www.gps.gov/>
 - [59] “Galileo,” European Global Navigation Satellite Systems Agency. [Online]. Available: <https://www.gsa.europa.eu/>
 - [60] “GLONASS,” Information and Analysis Center for Positioning, Navigation and Timing. [Online]. Available: <https://www.glonass-iac.ru/en/index.php>
 - [61] “Beidou,” BeiDou Navigation Satellite System. [Online]. Available: <http://en.beidou.gov.cn/index.html>
 - [62] “IRNSS,” Indian Space Research Organization. [Online]. Available: <http://www.isro.gov.in/irnss-programme>
 - [63] “UCS Satellite Database,” Union of Concerned Scientists. [Online]. Available: <http://www.ucsusa.org/nuclear-weapons/space-weapons/satellite-database#.WDNUwoWcGM9>
 - [64] B. W. Carroll and D. A. Ostlie, *An Introduction to Modern Astrophysics*. Pearson, 2006.
 - [65] “Jason-3,” NOAA National Environmental Satellite, Data, and Information Service (NESDIS). [Online]. Available: <https://www.nesdis.noaa.gov/jason-3/>

-
- [66] “Terra Bella,” Google Inc. [Online]. Available: <https://terrabella.google.com/>
 - [67] “Planet,” Planet Labs Inc. [Online]. Available: <https://www.planet.com/>
 - [68] “World Radiocommunication Conference,” International Telecommunications Union. [Online]. Available: <http://www.itu.int/en/ITU-R/conferences/wrc/Pages/default.aspx>
 - [69] “U.S. Frequency Allocation Chart,” National Telecommunications and Information Administration. [Online]. Available: <https://www.ntia.doc.gov/page/2011/united-states-frequency-allocation-chart>
 - [70] “Rec. ITU-R P.531-12 - Ionospheric propagation data and prediction methods required for the design of satellite services and systems,” International Telecommunication Union, Tech. Rep., 2013.
 - [71] “Rec. ITU-R P.618-12 - Propagation data and prediction methods required for the design of earth-space telecommunication systems,” International Telecommunication Union, Tech. Rep., 2015.
 - [72] “Rec. ITU-R P.676-11 - Attenuation by atmospheric gases,” International Telecommunication Union, Tech. Rep., 2016.
 - [73] T. S. Humphreys, M. L. Psiaki, J. C. Hinks, B. O’Hanlon, and P. M. Kintner, “Simulating ionosphere-induced scintillation for testing gps receiver phase tracking loops,” *IEEE Journal of Selected Topics in Signal Processing*, 2009.
 - [74] W. Liu and D. Michelson, “Fade slope analysis of Ka-band Earth-LEO satellite links using a synthetic rain field model,” *IEEE Transactions on Vehicular Technology*, vol. 58, no. 8, pp. 4013–4022, Oct 2009.
 - [75] P. V. R. Ferreira and A. M. Wyglinski, *Satellite-Communications: 3D rain field and LEO time series v1.1*. GitHub, 2016, doi: 10.5281/zenodo.44881.
 - [76] ——, “QoS improvement for LEO-based satcom at Ka-Band,” *Submitted to IEEE Transactions on Aerospace and Electronic Systems*, 2016.

-
- [77] ——, *Satellite-Communications: LMS time series gen GEO v1.1*. GitHub, 2015, doi: 10.5281/zenodo.44695.
 - [78] P. V. R. Ferreira, R. Paffenroth, and A. M. Wyglinski, “Interactive Multiple Model filter for land-mobile satellite communications at Ka-band,” *IEEE Access*, 2017.
 - [79] S. Naylor, *Not a Good Day to Die: The Untold Story of Operation Anaconda*. Berkley Hardcover, 2005.
 - [80] L. Neville, *Takur Ghar—The SEALs and Rangers on Roberts Ridge Afghanistan 2002*. Osprey, 2013.
 - [81] M. A. Kelly, J. M. Comberiate, E. S. Miller, and L. J. Paxton, “Progress toward forecasting of space weather effects on UHF satcom after Operation Anaconda,” *American Geophysical Union*, 2014.
 - [82] M. C. Kelley, *The Earth’s Ionosphere. Plasma Physics and Electrodynamics*. Academic Press, 1989.
 - [83] J. Huba and G. Joyce, “Global modeling of equatorial plasma bubbles,” *Geophysical Research Letters*, 2010.
 - [84] “National Aeronautics and Space Administration,” National Aeronautics and Space Administration. [Online]. Available: <http://www.nasa.gov>
 - [85] “National Oceanic and Atmospheric Administration satellite information system,” National Oceanic and Atmospheric Administration. [Online]. Available: <http://www.noaasis.noaa.gov/NOAASIS/ml/genlsatl.html>
 - [86] “About ionospheric scintillation,” Australian Government Bureau of Meteorology. [Online]. Available: <http://www.ips.gov.au/Satellite/6/3>
 - [87] “Sun unleashes 1st monster solar flare of 2015,” Space.com. [Online]. Available: <http://www.space.com/28797-sun-unleashes-monster-solar-flare-x2.html>
 - [88] “Severe solar storm hitting earth,” USA Today. [Online]. Available: <http://www.usatoday.com/story/weather/2015/03/17/solar-geomagnetic-storm/24901903/>

-
- [89] “Embrace,” Instituto Nacional de Pesquisas Espaciais. [Online]. Available: <http://www2.inpe.br/climaespacial/portal/sci-home/>
 - [90] Y. Feria and et. al., “Solar scintillation effects on telecommunication links at Ka-Band and X-Band,” *TDA Progress Report 42-129*, May 1997.
 - [91] T. S. Rappaport, *Wireless Communications Principles and Systems*. Prentice-Hall, 2002.
 - [92] E. C. Jordan and K. G. Balmain, *Electromagnetic Waves and Radiating Systems*. Prentice-Hall, 1968.
 - [93] M. Patzold, *Mobile Radio Channels*. Prentice-Hall, 2002.
 - [94] M. Failli, “COST 207: Digital Land Mobile Radio Communications,” Luxembourg: Commission of the European Communities, Tech. Rep., 1989.
 - [95] D. C. Cox, “Delay Doppler characteristics of multipath propagation at 910 MHz in a suburban mobile radio environment,” *IEEE Transactions on Antennas and Propagation*, 1972.
 - [96] P. A. Bello, “Aeronautical channel characterizations,” *IEEE Transactions on Communications*, 1973.
 - [97] A. Neul, J. Hagenauer, W. Papke, F. Dolinsky, and F. Edbauer, “Aeronautical channel characterization based on measurement flights,” *Proceedings IEEE Global Communications Conference*, 1987.
 - [98] MATLAB, *version 9.2.0.538062 (R2017a)*. Natick, Massachusetts: The MathWorks Inc., 2017.
 - [99] STK, *version 10.1.0*. Exton, PA: Analytical Graphics, Inc., 2014.
 - [100] “Rec. ITU-R P.837-6 - Characteristics of precipitation for propagation modeling,” International Telecommunication Union, Tech. Rep., 2012.

-
- [101] C. Capsoni, F. Fedi, C. Magistroni, A. Paraboni, and A. Pawlina, “Data and theory for a new model of the horizontal structure of rain cells for propagation applications,” *Radio Science*, vol. 22, no. 3, pp. 395–404, 1987.
 - [102] J. Goldhirsh, “Two-dimension visualization of rain cell structures,” *Radio Science*, vol. 35, no. 3, pp. 713–729, 2000.
 - [103] “Software to compute the rain fall rate exceed at a certain percentage of the year.” International Telecommunication Union. [Online]. Available: <http://www.itu.int/oth/R0A04000025/en>
 - [104] “Rec. ITU P.839-4 - Rain height model for prediction methods,” International Telecommunication Union, Tech. Rep., 2013.
 - [105] “Rec. ITU-R P.1853-1 - Tropospheric attenuation time series synthesis,” International Telecommunication Union, Tech. Rep., 2012.
 - [106] “ITU Study Group 3 - Working party 3M. Software, data and validation examples for ionospheric and tropospheric propagation and radio noise.” International Telecommunication Union. [Online]. Available: <http://www.itu.int/en/ITU-R/study-groups/rsg3/Pages/iono-tropo-spheric.aspx>
 - [107] “Rec. ITU-R P.681-8 - Propagation data required for the design of Earth-space land mobile telecommunication systems,” International Telecommunication Union, Tech. Rep., 2015.
 - [108] G. Corazza, *Digital Satellite Communications*. Springer, 2007.
 - [109] “Technical note on the implementation of the land mobile satellite channel model—Software usage,” German Aerospace Center DLR, Tech. Rep., 2007.
 - [110] R. Prieto-Cerdeira, F. Perez-Fontan, P. Burzigotti, A. Bolea-Alamanac, and I. Sanchez-Lago, “Versatile two-state land mobile satellite channel model with first application to DVB-SH analysis,” *International Journal of Satellite Communications and Networking*, vol. 29, pp. 291–315, June 2010.

-
- [111] W. Li, C. L. Law, J. T. Ong, and V. Dubey, “Ka-band land mobile satellite channel model: with rain attenuation and other weather impairments in equatorial zone,” *IEEE 51st Vehicular Technology Conference*, 2000.
 - [112] P. V. R. Ferreira, R. Metha, and A. M. Wyglinski, “Cognitive radio-based geostationary satellite communications for Ka-band transmissions,” *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 1093–1097, 2014.
 - [113] L. Ottavj, E. Duros, J. Webert, F. Gamberini, C. Niephaus, J. Perex-Trufero, and S. Watts, “Intelligent gateways enabling broadband access via integrated terrestrial and satellite systems,” *7th International Conference on Wireless and Satellite Systems*, vol. 7, pp. 92–102, July 2015.
 - [114] T. M. Hackett, S. G. Bilén, P. V. R. Ferreira, A. M. Wyglinski, and R. Reinhart, “Implementation of a parameterized Interacting Multiple Model filter on an FPGA for satellite communications,” *34th AIAA International Communications Satellite Systems Conference*, 2016.
 - [115] A. Assa and F. Janabi-Sharifi, “A Kalman filter-based framework for enhanced sensor fusion,” *IEEE Sensors Journal*, vol. 15, pp. 3281–3292, June 2015.
 - [116] W. Qi, P. Zhang, N. G.H., and Z. Deng, “Robust weighted fusion Kalman predictors with uncertain noise variances,” *Digital Signal Processing*, vol. 30, p. 3754, July 2014.
 - [117] R. Faragher, “Understanding the basis of the Kalman filter via a simple and intuitive derivation,” *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 128–132, September 2012.
 - [118] M. Yasuyuki, F. Tadashi, and S. Yoshiaki, “Effects of tropical rainfall to the Ku-Band satellite communications links at the equatorial atmosphere radar observatory,” *Journal of the Meteorological Society of Japan*, 2006.
 - [119] A. K. Shukla, S. Das, and B. Roy, “Rain attenuation measurements using synthetic storm technique over Ahmedabad,” *In Proceedings of International Conference on Computers and Devices for Communication*, 2009.

- [120] H. Bischl, H. Brandt, T. de Cola, R. D. Gaudenzi, E. Eberlein, N. Girault, E. Alberty, S. Lipp, R. Rinaldo, B. Rislow, J. A. Skard, J. Tousch, and G. Ulbricht, “Adaptive coding and modulation for satellite broadband networks: From theory to practice,” *International Journal of Satellite Communications and Networking*, vol. 28, no. 2, pp. 59–111, 2010. [Online]. Available: <http://dx.doi.org/10.1002/sat.932>
- [121] N. Tudoroiu and K. Khorasani, “Satellite fault diagnosis using a bank of interacting Kalman filters,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, pp. 1334–1350, October 2007.
- [122] P. V. R. Ferreira, R. Paffenroth, A. M. Wyglinski, T. M. Hackett, S. G. Bilén, R. C. Reinhart, and D. J. Mortensen, “Multi-objective reinforcement learningbased deep neural networks for cognitive space communications,” in *IEEE Cognitive Communications for Aerospace Applications Workshop*, June 2017.
- [123] ——, “Multi-objective Reinforcement Learning for Cognitive Satellite Communications using Deep Neural Networks Ensembles,” in *In preparation*, April 2017.
- [124] NASA GRC SCaN Testbed, available at <http://spaceflightsystems.grc.nasa.gov/SOPO/SCO/SCaNTTestbed/>.
- [125] D. Chelmins, J. Downey, S. K. Johnson, and J. Nappier, “Unique challenges testing sdrs for space,” *Proceedings of the 2013 IEEE Aerospace Conference, Big Sky, Montana*, 2013.
- [126] S. Johnson, D. Chelmins, D. Mortensen, M. Shalkhauser, and R. Reinhart, “Lessons learned in the first year operating software defined radios in space,” *Proceedings of the 2014 IEEE Aerospace Conference, Big Sky, Montana*, 2014.
- [127] S. K. Johnson, R. C. Reinhart, and T. J. Kacpura, “CoNNeCTs approach for the development of three software defined radios for space application,” *Proceedings of IEEE Aerospace Conference*, pp. 1–13, 2012.
- [128] ——, “CoNNeCTs approach for the development of three software-defined radios for space application,” *Proceedings of IEEE Aerospace Conference*, pp. 1–13, 2012.

-
- [129] “Space telecommunications radio systems architecture standard,” available at <https://standards.nasa.gov/standard/nasa/nasa-std-4009>.
 - [130] “Space telecommunications radio systems architecture standard rationale,” available at <https://standards.nasa.gov/standard/nasa/nasa-hdbk-4009>.
 - [131] R. C. Reinhart, “Using international space station for cognitive system research and technology with space-based reconfigurable software defined radios,” *66th International Astronautical Congress*, 2015.
 - [132] “NASA Technology Roadmaps,” National Aeronautics and Space Administration. [Online]. Available: <https://www.nasa.gov/offices/oct/home/roadmaps/index.html>
 - [133] A. Barto and R. S. Sutton, *Reinforcement Learning: An Introduction*. MIT Press, 1988.
 - [134] D. A. Berry and B. Fristedt, *Bandit Problems: Sequential Allocation of Experiments*. Springer, 1985.
 - [135] T. Lai and H. Robbins, “Asymptotically efficient adaptive allocation rules,” *Advances in Applied Mathematics*, vol. 6, no. 1, 1985.
 - [136] A. Motamedi and A. Bahai, “Dynamic channel selection for spectrum sharing in unlicensed bands,” *European Transactions on Telecommunications and Related Technologies*, 2007.
 - [137] L. Lai, H. Jiang, and H. V. Poor, “Medium access in cognitive radio networks: A competitive multi-armed bandit framework,” *Proceedings of IEEE Asilomar Conference on Signals, Systems, and Computers*, 2008.
 - [138] L. Busoniu, R. Babuska, B. D. Schutter, and D. Ernst, *Reinforcement Learning and Dynamic Programming Using Function Approximators*. CRC Press, 2010.
 - [139] J. G. Proakis and M. Salehi, *Digital Communications*. McGraw Hill, 2007.
 - [140] K. Cho and D. Yoon, “On the general BER expression of one- and two-dimensional amplitude modulations,” *IEEE Transactions on Communications*, vol. 50, no. 7, 2002.

- [141] M. Tokic and G. Palm, “Value-difference based exploration: Adaptive control between epsilon-greedy and softmax,” *KI 2011: Advances in Artificial Intelligence: 34th Annual German Conference on AI*, pp. 335–346, 2011.
- [142] M. Tokic, “Adaptive e-greedy exploration in reinforcement learning based on value differences,” *KI 2010: Advances in Artificial Intelligence: 33rd Annual German Conference on AI*, pp. 203–210, 2010.
- [143] S. L. Scott, “A modern Bayesian look at the multi-armed bandit,” *Applied Stochastic Models in Business and Industry*, vol. 26, no. 6, pp. 639–658, 2010.
- [144] J. Langford and T. Zhang, “The epoch-greedy algorithm for multi-armed bandits with side information,” *Advances in Neural Information Processing Systems*, vol. 20, 2007.
- [145] C. Gentile, S. Li, and G. Zappella, “Online clustering of bandits,” *Proceedings of the 31st International Conference on Machine Learning*, vol. 20, 2014.
- [146] S. S. Haykin, *Neural Networks : A Comprehensive Foundation*. Prentice Hall, 1999.
- [147] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, pp. 529–541, February 2015.
- [148] “Self-driving cars,” The Guardian. [Online]. Available: <https://www.theguardian.com/technology/self-driving-cars>
- [149] “Google Brain,” Google Inc. [Online]. Available: <https://research.google.com/teams/brain/>
- [150] “Deep Mind,” Google Inc. [Online]. Available: <https://deepmind.com/>
- [151] “OpenAI,” OpenAI. [Online]. Available: <https://openai.com/>
- [152] “Facebook AI Research,” Facebook. [Online]. Available: <https://research.fb.com/category/facebook-ai-research-fair/>

- [153] “NVDIA Deep Learning,” NVDIA. [Online]. Available: <https://www.nvidia.com/en-us/deep-learning-ai/>
- [154] “Center for Brains Minds and Machines,” MIT Center for Brains Minds and Machines. [Online]. Available: <https://cbmm.mit.edu>
- [155] Y. B. Ian Goodfellow and A. Courville, *Deep Learning*. The MIT Press, 2016.
- [156] “Neural Networks FAQ,” available at <http://www.faqs.org/faqs/ai-faq/neural-nets/part2/>.
- [157] M. Hagan and M. Menhaj, “Training feed-forward networks with the Marquardt algorithm,” *IEEE Transactions on Neural Networks*, vol. 5, p. 989993, 1994.
- [158] H. Yu and B. M. Wilamowski, *Levenberg–Marquardt Training*. The Industrial Electronics Handbook, 2nd Edition, 2012.
- [159] “DVB-S2 Standard,” available at <https://www.dvb.org/standards/dvb-s2>.
- [160] M. Tokic and G. Palm, “Value-difference based exploration: Adaptive control between epsilon-greedy and softmax,” *KI 2011: Advances in Artificial Intelligence: 34th Annual German Conference on AI*, pp. 335–346, 2011.
- [161] M. Tokic, “Adaptive ϵ -greedy exploration in reinforcement learning based on value differences,” *KI 2010: Advances in Artificial Intelligence: 33rd Annual German Conference on AI*, pp. 203–210, 2010.
- [162] P. V. R. Ferreira and A. M. Wyglinski, *Simulated DVB-S2 BER curves for long-frames*. GitHub, 2017, doi: 10.5281/zenodo.495732.
- [163] C. T. Chen, *Linear System Theory and Design*. Oxford University Press, 2012.
- [164] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*. Prentice Hall PTR, 2011.