Design and Implementation of a Robust Downlink Communication System for Nanosatellites

Jagdish Mevada[1,2], Joseph Samuel[1], Sourbh Bhadane[1], Akshay K. Gulati[1], R. D. Koilpillai[1]
[1]Department of Electrical Engineering, Indian Institute of Technology Madras, Chennai, India
[2]Indian Space Research Organization, Ahmedabad, India
ee12b127@ee.iitm.ac.in

***Abstract*—The design of a robust downlink communication system for nanosatellites in UHF amateur radio band involves many challenges like radio spectrum constraints, time-varying Doppler, clock drifts and synchronization in a LEO satellite channel. This paper presents the system designed and implemented for IITMSAT, a student satellite project at IIT Madras, which successfully overcomes these challenges using a robust physical layer design implemented on a generic Software Defined Radio platform. The performance of the proposed design is verified through simulations as well as hardware. The communication system gives a good performance which is only about 1 dB away from theory.**

Keywords—Software Defined Radio; GNU Radio; Physical layer; Signal acquisition; Nanosatellite; Synchronization; Doppler compensation

I. INTRODUCTION

Indian Institute of Technology Madras Satellite (IITMSAT) [1] is a student satellite project at the Indian Institute of Technology Madras, Chennai, India. The satellite, which is to be launched in a Low Earth Orbit (LEO), aims to study the precipitation of high-energy charged particles (protons and electrons) from Van Allen belts [2]. The satellite payload generates large amounts of scientific data that has to be downlinked. A communication system which achieves the best possible link performance and channel utilization needs to be designed to reduce retransmission overheads. Due to limited resources available on-board, the communications subsystem cannot be made very complex. This necessitates an efficient and robust receiver setup at the ground station. The receiver is designed and implemented on Software Defined Radio platform which offers many advantages in this scenario.

II. SYSTEM DESCRIPTION

A. Challenges

The IITMSAT payload generates about 1 MB of data per day which needs to be downlinked in less than 20 minutes of useful visible time for LEO satellite over ground station. A major challenge here is the spectrum constraint - IARU allocates 20 kHz of bandwidth in the 435 MHz range and ITU-R recommendation [3] for satellite applications requires that the power emission outside the assigned bandwidth should not exceed 1% of the total power within the assigned bandwidth.

For a LEO satellite, the relative motion between satellite and ground station will be high and accelerated in nature. Due to the time-varying relative velocity, the Doppler offset keeps varying and the rate of change of Doppler offset will be at its peak at the highest elevation angle. In particular, for a satellite in a nearly circular LEO orbit with a mean altitude of about 700 km, the maximum Doppler offset on UHF carrier would be about 10 kHz. The rate of change of Doppler offset, in this case around zenith point, would be about 120 Hz/sec and on an average across multiple useful passes, it will be about 60 Hz/s. Fig. 1 shows Doppler pattern (called S-pattern) observed over one pass for Swisscube [4] which is a nanosatellite with mean altitude 712 km. (The highest elevation angle in this pass is 84 degrees.) This frequency offset has to be corrected at the ground station accurately to achieve the best link performance. A rough estimate of this offset can be calculated using the Two Line Element (TLE) set which is made available online [5] once the satellite is in its final orbit. This data is subsequently updated intermittently by NORAD. The Doppler estimate from TLE gets increasingly inaccurate as the TLE data gets older. Hence there is a need for accurate Doppler frequency offset estimation and tracking which needs to be taken care of in the physical layer (PHY) design of the proposed communication system.

Acquiring the satellite signal for the first time at low elevation angles is particularly challenging. Due to high slant

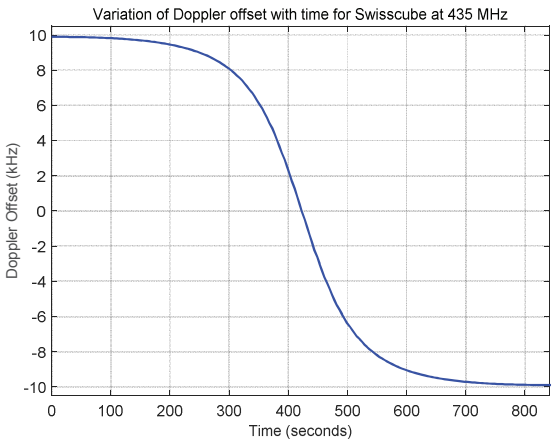

Figure 1. Variation of Doppler offset during a pass for a typical nanosatellite Swisscube [4]

range, the propagation loss will be at its peak which will result in lower SNR. Also at low elevation angles, the channel fading effect will be considerable and Doppler offset will be at its maximum.

In any practical system, the transmitter and receiver will have differences in clock rate - one may be running faster than the other. In addition to this, the time-varying slant range (varying from 3000 km at low elevation to 700 km at the highest elevation angle) causes a time-varying propagation delay. This necessitates a robust timing synchronization mechanism.

The size of the nanosatellite introduces another challenge in isolating the on-board uplink and downlink RF signals, which may result in on-board transmitter power leaking into the on-board receiver system that may potentially damage the receiver or cause interference and degrade the uplink performance. Another challenge is a peak power constraint for the on-board transmitter (which is 1 W for IITMSAT).

B. Known Methods

Most nanosatellites implement packet radio schemes used in amateur radio applications which use low data rates (like 1200 bps) and simple but spectrally inefficient modulation schemes like FSK or OOK [6]. For such systems, simple non-coherent demodulation methods will be satisfactory. This allows their ground stations to use commercial-off-the-shelf (COTS) radios which can support these schemes. Scope of such COTS radios is limited to applications where standard protocols and modulation schemes are used.

III. Proposed Method

The approaches described above are not suitable for IITMSAT due to its high data requirements. High data rates imply wider bandwidths which will increase receiver noise in the system and thereby lower the link margin. This will prevent simple non-coherent schemes from meeting the performance requirements. For IITMSAT, although the system is designed for full-duplex mode communication, the link is operated in a half-duplex mode to avoid self-interference. However to operate in half-duplex mode, transmitter needs to be turned OFF and ON many times in a given pass of satellite over the ground station. The proposed design allows the receiver to synchronize with the transmitter anytime during a pass with minimum overhead, thereby maximizing the channel utilization for data transmission.

A. Data Rate, Channel Coding and Modulation Scheme

A raw data rate of 9600 bps was identified to be sufficient based on mission requirements. Channel coding using rate ½ convolutional code with a constraint length of 5 is employed to increase the link reliability. This makes the effective downlink channel rate 19200 bps.

To send symbols at a rate of 19.2 kbps in 20 kHz channel bandwidth, a spectrally-efficient modulation scheme has to be chosen which meets the spectral mask requirements. GMSK with BT value 0.3 was chosen. This introduces ISI due to the partial response nature of GMSK which requires the receiver to use channel equalization techniques to achieve the best BER performance.

B. SDR-based Receiver

Most COTS transceivers have 5 or 15 kHz filters at the IF stage complying with different standards. These narrow band receivers cannot be used in our design. This makes Software Defined Radio (SDR) an appropriate choice for this application. SDR not only helps to overcome the filter bandwidth issue posed by COTS receivers but also gives the flexibility to implement enhanced algorithms at the physical layer. By moving all computation tasks to software running on a PC, SDR offers the flexible use of resources to achieve the best possible system performance. GNU Radio, a free software radio platform, is used as the software part of SDR system. For the hardware part, Universal Software Radio Peripheral (USRP) is used for testing purposes but the final design is being implemented on Perseus Radio, which has better RF specifications [7, 8].

C. Physical Layer Design

SDR offers the flexibility to have custom physical layer protocols. To utilize this advantage, a physical layer frame structure has been designed that provides robust RF synchronization at reasonable overheads. The receiver algorithms are designed based on this frame structure.

The frame structure is designed to allow the receiver to synchronize with the transmitter at any time during the visibility window. The synchronization is made robust enough such that it can be achieved even at the worst case SNR. During long communication sessions, the proposed frame

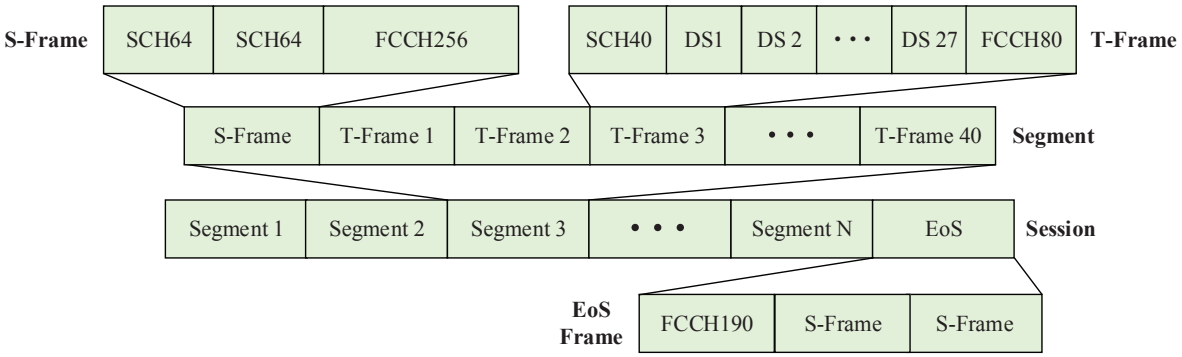

Figure 2. Physical Layer Frame Structure

structure allows the receiver to resynchronize in case the synchronization is lost due to a deep fade or other channel perturbations during low elevation communication. This means that only the data corresponding to the duration of deep fade (or other outage conditions) will be affected and this can be replaced via retransmission (ARQ).

The frame structure is shown in Fig. 2. The components are as follows.

1) Synchronization Frames (S-Frames)

The first part of an S-Frame consists of two 64-bit pseudo random sequences, each of which is called Synchronization Channel (SCH). This sequence of bits, as in the GSM standard [9], is a pseudo-random sequence which is used for detecting the presence of signal and to establish coarse timing synchronization between the transmitter and receiver. Two SCHs are incorporated in an S-Frame which together ensures robust detection of signal even in severe channel and noise conditions.

The second part of the S-Frame contains the Frequency Correction Channel or FCCH. As in the GSM standard [9], the FCCH contains a sequence of zeros which corresponds to a pure tone at one-fourth of channel baud rate after GMSK modulation. The receiver can estimate the overall frequency offset in the system (combination of Doppler shift and frequency offsets at transmitter and receiver) by calculating the frequency of this tone.

Thus the total length of an S-Frame is 384 bits which correspond to 20 milliseconds. The algorithms used for detection of S-Frames are presented in Section IV.

2) Traffic Frames (T-Frames)

These frames carry the downlink data from satellite to ground station. T-Frames start with a 40-bit SCH which is used for bit-timing tracking to mitigate the effect of clock drifts between transmitter and receiver and possible time-varying propagation delays. It is also used as a training sequence for channel estimation for decoding the first data slot following it. After SCH, there are 27 data slots each of length 40 bits. Out of those 40 bits, 38 bits correspond to channel coded bits and 2 bits correspond to the zero padding required for channel estimation and demodulation (described later). The last part of T-frame is an 80-bit FCCH which allows the receiver to track the Doppler variation between successive frames.

The dimension of each of the fields in S-frames and T-frames were suitably chosen to maintain a balance between channel efficiency, overhead and link performance.

3) Segment and Session

The combination of one S-Frame and 40 T-Frames is called a **segment**. Each segment lasts for 2.52 seconds. A sequence of downlink data 'segments' is called a **session**. There can be any number of segments in a single session depending on the amount of data to be transmitted. This means that an S-frame will be observed by the receiver every new segment (2.52 seconds), which ensures that only a segment of data would be lost even if the receiver misses the first S-Frame.

4) End of Session (EoS) Frame

The end of a session is designed to be identified in the physical layer itself. This makes the physical layer design independent of higher layer design. It also helps to reduce the switching time between downlink and uplink, thus improving channel utilization. The transmitter indicates EoS by sending a unique EoS frame after the last segment. This frame consists of a 190-bit FCCH followed by two S-frames. Such a structure allows the receiver to determine EoS with a very high probability. The algorithm for detecting EoS frame is given in Section IV.

5) Overhead Calculation

The total overhead in the physical layer can be computed from the overhead of one segment.

TABLE I. OVERHEAD CALCULATION

(1)	Data bits in one slot	38
(2)	Data slots in one T-frame	27
(3)	Data bits in one T-frame = (1) * (2)	1026
(4)	T-frames in one segment	40
(5)	Data bits in one segment = (3) * (4)	41040
(6)	Total bits in one segment	48384
(7)	Overhead = 1-(5)/(6)	**15.18 %**

As seen from Table I, the total overhead is kept to a

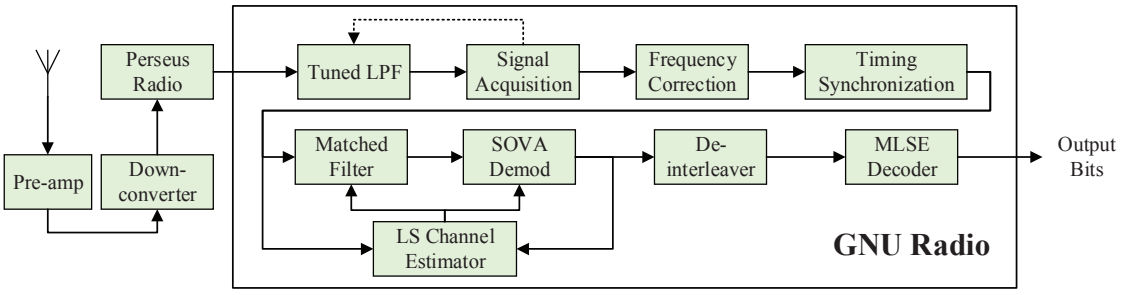

Figure 3. Receiver Block Diagram

minimum. It corresponds to an effective link throughput of 8143 bps out of 9600 bps.

D. Receiver Design and Implementation

The overall structure of the receiver is shown in Fig. 3. The received signal is first amplified (using a low noise pre-amplifier) and downconverted to 26-36 MHz band to match the input frequency range of Perseus Radio [7]. The Perseus Radio then gives baseband IQ samples to a PC running GNU Radio software. The sampling rate is double the symbol rate (i.e., 38400 samples per second) and each sample is of single precision complex data type.

In GNU Radio, the incoming samples are first passed through a tuned low-pass filter to remove out-of-band noise. This LPF will be first tuned using TLE-based estimated frequency offset and later using the computed frequency offset from Signal Acquisition block. The filtered signal is then passed to the Signal Acquisition block which determines if the signal is present by checking for S-Frames, and once it detects an S-Frame, it passes the subsequent 40 T-frames to the frequency correction and bit-timing synchronization blocks which utilize the FCCH and SCH blocks respectively. The frequency and timing corrected data part is demodulated using Soft Output Viterbi Algorithm (SOVA) [10] demodulator after matched filtering and channel estimation. The demodulated symbols (soft output) are de-interleaved and then decoded using outer Viterbi MLSE channel decoder. The decoder output bits are given to Ground Station Front End Software for higher layer processing.

There are two modes in which the receiver operates:

1. Acquisition Mode
2. Traffic Mode

By default, the receiver operates in Acquisition Mode until the signal is acquired for the first time in a given pass, i.e., when the S-frame of the first segment corresponding to the first session is detected. Then the receiver switches to Traffic Mode in which the Signal Acquisition block is bypassed and the rest of the receiver is activated to decode T-frames of the corresponding segment. The receiver again switches back to acquisition mode on the arrival of the second segment and corrects the frame synchronization using S-frame and then switches again to traffic mode. This operation is repeated until EoS-Frame is detected, after which the receiver goes back to default Acquisition Mode to acquire the signal during next session.

IV. Receiver Algorithms

Receiver algorithms, which are implemented in GNU Radio, are divided into three functional blocks:

(a) Signal Acquisition

(b) Frequency and Timing correction

(c) Demodulation and Decoding.

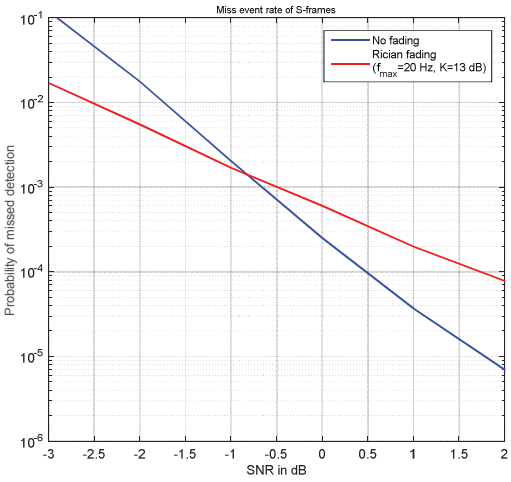

Figure 4. Performance of the proposed S-Frame acquisition

A. Signal Acquisition

The first task of the receiver is to initially acquire the satellite signal at low elevation angles (where the satellite channel impairment is maximum compared to other angles) and to establish the receiver synchronization. A robust signal acquisition algorithm was designed based on the PHY frame structure mentioned earlier. The algorithm uses S-frames to detect the presence of signal and ensures receiver synchronization within a timing error of one symbol duration until the session gets over. In case the synchronization is lost due to an outage before the session gets over (i.e., prior to EoS), the algorithm can re-acquire synchronization during next segment.

The algorithm starts by estimating the frequency of the incoming symbols (assuming that the symbols at the respective positions correspond to FCCH of the S-frame) on a symbol-by-symbol basis. A computationally-efficient estimator is used for this purpose [11] which requires the computation of two autocorrelation lags (which has been implemented using AutoRegressive Moving Average (ARMA) model) and then a small lookup table to estimate the frequency. 80 symbols are used for computing the autocorrelation lags using ARMA such that sufficient number of consecutive estimates will have nearly the same estimated value if the symbols correspond to FCCH of an S-frame. These estimates are used to correct the symbols that could correspond to SCH. These corrected symbols are passed through a matched filter corresponding to the actual SCH. The magnitude output of this filter is stored in a circular buffer of length 128 symbols. If the symbols correspond to an S-frame, then this 128-symbol circular buffer will contain frequency-corrected and matched-filtered magnitude output of the two SCHs in the S-frame.

A peak-detection algorithm is then applied on this buffer. The algorithm leverages the fact that there are *two* SCH's in an S-frame which should correspond to *two* peaks separated by 64 symbols. Hence the peak detector considers only two triplet samples separated by 64 symbols from this buffer and applies relative-peak and noise averaging methods to robustly detect the S-frame. The peak detector algorithm does not have any 'absolute' comparisons. This means that an arbitrary scaling of

the signal will not affect the algorithm performance as long as it does not exceed the floating point precision. This overcomes the effect of fading and unknown noise power.

The performance of the proposed signal acquisition algorithm was analyzed through MATLAB simulations. The probability of missed detection at various SNR was calculated for both AWGN and Rician slow flat fading channels. Fig. 4 shows that the algorithm can reliably detect S-frames even in the presence of fading if SNR is above 2 dB, which is ensured by the link margin of the communication link.

The signal acquisition method allows the rest of the receiver to "wake up" only when an S-frame is detected. Once the first S-frame is detected, i.e., the signal is acquired for the first time, the acquisition method knows the expected location of next S-frame or EoS frame. By leveraging this fact, the receiver switches back and forth between Acquisition Mode and Traffic Mode. This allows the rest of the receiver algorithms to use full processing resources to decode T-frames during Traffic Mode.

Before switching to Traffic Mode, the acquisition block provides an accurate estimate of the effective frequency offset to the next block. This estimate is also fed back to the tuned LPF for tracking purposes. This estimate is computed by making use of FCCH256 in the S-frame through a complex but highly accurate Guided Search [12] algorithm which requires the computation of FFT. This algorithm will be run only once in acquisition mode. The low-complexity frequency estimator will run continuously until the first S-frame of the session is detected after which both estimators will run only during the S-frame part of every segment.

The acquisition block also performs End-of-Session detection by detecting the EoS frame. The EoS frame is designed such that the existing S-frame detection algorithms can be used - the EoS frame contains S-frames but the location of these S-frames differ from the expected locations of S-frame in a segment. This difference in position is used to differentiate the EoS-frame from a normal S-frame. Two S-frames are repeated in an EoS-frame so that the EoS-frame missed detection ideally will be the square of S-frame missed detection probability shown in Fig. 4.

B. *Frequency and Timing Correction*

Once the signal is acquired and initial synchronization is achieved using the above method, the receiver's next task is to keep track of Doppler and timing drifts during the traffic mode. Receiver achieves this by making use of FCCH80 and SCH40 of each T-frame.

1) Doppler Frequency Tracking

As mentioned earlier, acquisition block passes all traffic frames of each segment along with an accurately estimated value of frequency offset at the beginning of that segment. Using this estimated frequency, Doppler frequency tracking will be performed progressively on every T-frame by fine tuning the estimated frequency using the iterative part of Guided Search algorithm [12] which requires DFT computation of 1 bin per iteration. This updated frequency will be used for frequency correction of the T-frame. Processing is done on frame-by-frame basis. The frame structure is designed such that for any T-frame, it will have FCCH on either side. In addition, the monotonic Doppler S-pattern can be approximated as a straight line over a small interval of time. These two factors together allow us to average frequency estimates over a small number of frames. Estimates are averaged over 4 T-frames which reduces the effective variance by a factor of 4. For T-frames at the edges of the segment, the highly accurate estimate provided by acquisition block is incorporated with a higher weightage in the averaging process.

2) Clock Timing Drift Tracking

Bit timing synchronization and clock drift tracking are performed using SCH part of the T-frame. The receiver operates at double the symbol rate, which results in two streams at symbol rate, viz., even and odd separated by a delay of half the symbol period. The SCH40 (containing a pseudo-random sequence) is passed through two identical matched filters for even and odd streams. The global peaks for each stream at a neighbourhood of 5 symbols around the expected location is determined. The stream corresponding to the larger peak is given to the Demodulation block. The resulting stream will have a timing synchronization accuracy within half a symbol period. The clock can drift over a period of time and so this operation is repeated for every T-frame to track it.

C. *Demodulation and Decoding*

For coherent demodulation, the receiver requires the knowledge of Channel Impulse Response (CIR). Standard Least Squares channel estimator, which achieves the Cramer Rao Lower Bound (CRLB) [13], is used for this purpose. It is implemented using Gauss Elimination method by exploiting circulant property of the symbol matrix. Initially the SCH40 training sequence is used for estimation; then it uses decoded bits (decision-directed mode) for the remaining data slots of each T-frame, i.e., decoded data of a given data slot will be used as the known sequence for estimating CIR for the data slot following it. Once the CIR is estimated, the demodulation is performed using standard Soft Output Viterbi Algorithm (SOVA) [10]. Soft outputs are used to enhance the

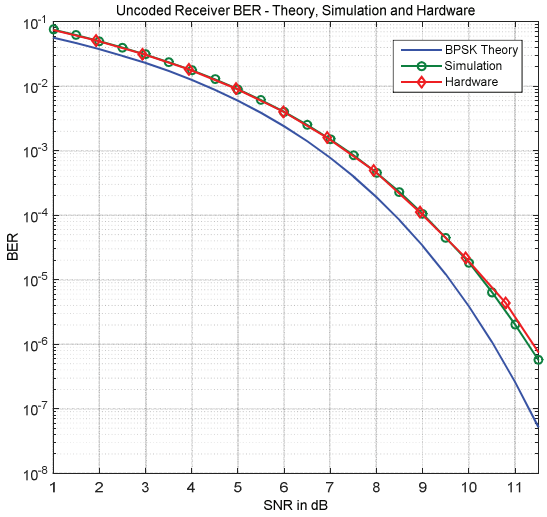

Figure 5. Performance of the system without channel coding and fading

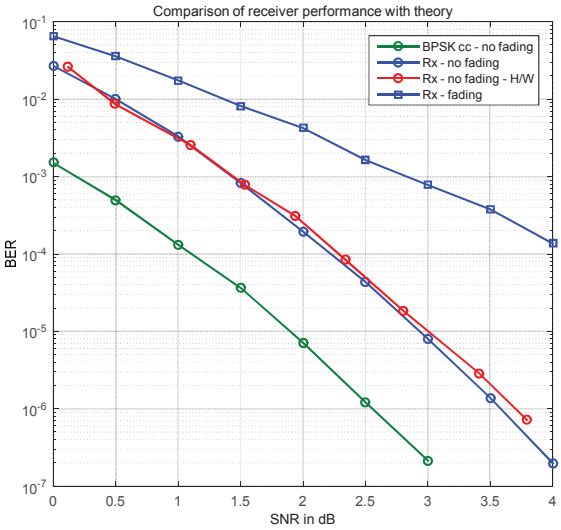

Figure6. Performance of the system with channel coding

performance of outer Viterbi decoder. Maximum Likelihood Sequence Estimation (MLSE) using the Viterbi algorithm is implemented for channel decoding.

V. Simulation and Hardware Testing Results

To validate the performance of the proposed design, the receiver algorithms were tested through simulations as well as on hardware. Hardware testing was done at a 435 MHz carrier with two USRPs each having WBX daughterboard which has a maximum transmit power of 100 mW. The two USRPs were connected with a cable and a 30 dB hardware attenuator. 20 dB of additional attenuation was implemented in software (at the transmitter) resulting in a maximum transmit power of 1mW and a received power of < 1 uW. In all simulations and hardware tests, a sinusoidal Doppler S-pattern was considered with a maximum Doppler rate of 100 Hz/sec. With the T-frame size of 62.5 ms, the maximum Doppler offset progression would be 6.25 Hz/frame.

Fig. 5 shows the results of BER simulation and hardware testing of the receiver without channel coding and fading. The simulation result matches BPSK theory with a difference of 0.7 dB. Also, the performance in the hardware test matches that of the simulation.

Fig. 6 shows the performance of the complete system with channel coding. The hardware performance again verifies the simulation results. In the absence of fading, the performance degradation from theory is about 1 dB at a BER of 1E-5. This degradation in performance is predominantly due to the residual frequency error which makes the estimated channel time-varying. Since averaging is used to estimate and track the Doppler, the estimated value will correspond to the middle of each T-frame. This means that the error (in perfect noiseless case with maximum Doppler offset rate of 100 Hz/sec) will range from -3.125 Hz at the beginning of the frame to +3.125 Hz at the end of the frame. As long as the magnitude is less than 4 Hz, this error does not degrade the performance significantly. Another factor affecting performance at low SNR is the error propagation when the LS channel estimator operates in decision-directed mode. Erroneous decisions from one data slot will be fed back to the estimator during the decoding of the next data slot which impacts performance.

Fig. 6 also shows the performance of the receiver in Rician flat fading conditions. To model the channel at low elevation angles, Jakes' model [14] was used with maximum residual Doppler frequency of 20 Hz and Rician factor of 13 dB. The performance loss in fading is about 3 dB compared to BPSK with channel coding.

VI. Conclusion

The proposed design overcomes the challenges involved in downlinking large amount of data from a nanosatellite. Moreover, it gives a performance which is only about 1 dB away from theory for a LEO satellite channel without fading. The design achieves this performance without sacrificing channel efficiency by restricting overhead to about 15%. The entire design is implemented in generic Software Defined Radio platform. The proposed communication system is well-suited for nanosatellites using amateur UHF bands to support more payloads requiring high data rate downlink.

References

[1] IITMSAT [Online]. Available: https://iitmsat.iitm.ac.in/

[2] N. Sivadas et al., "A Nano-satellite Mission to Study Charged Particle Precipitation from the Van Allen Radiation Belts caused due to Seismo-Electromagnetic Emissions", The 5th Japan Nano-Satellite Symposium, Tokyo, November 2013.

[3] ITU-R, Bandwidth Occupancy, Sec. 1.147 and 1.153, in International Telecommunication Union - Radio Regulations, 2006.

[4] Swiss Cube [Online]. Available: http://swisscube.epfl.ch

[5] CelesTrak: Current NORAD Two-Line Element Sets [Online]. Available: http://www.celestrak.com/NORAD/elements/

[6] B. Klofas, "A Survey of CubeSat Communication Systems: 2009-2012", CubeSat Developers' Workshop, April 2013.

[7] Perseus Receiver User Manual, Microtelecom s.r.l., Pavia di Udine, Italy, 2009.

[8] WBX Product Details, Ettus Research [Online]. Available: http://www.ettus.com/product/details/WBX

[9] GSM Technical Specification - GSM 05.02, Version 5.0.0, May 1996

[10] J. Hagenauer and P. Hoeher, "A Viterbi Algorithm with Soft-Decision Outputs and its Applications", in Proc. of GLOBECOM '89, IEEE Global Telecommunications Conf., pp. 1680-1686, November 1989.

[11] D. Tufts, and P. Fiore, "Simple, effective estimation of frequency based on Prony's method," in IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 5, pp. 2801-2804, 1996..

[12] E. Aboutanios and S. Reisenfeld, "Frequency estimation and tracking for low earth orbit satellites," in Vehicular Technology Conference, Vol.4, pp. 3003-3004, May 2001.

[13] S. M. Kay, Fundamentals of Statistical Signal Processing, Volume 1: Estimation Theory, Prentice-Hall PTR, 1998.

[14] C. Xiao, Y. R. Zheng, and N. C. Beaulieu, "Novel sum-of-sinusoids simulation models for Rayleigh and Rician fading channels," in IEEE Trans. Wireless Commun., vol. 5, no. 12, pp. 3673, December 2006.