Cognitive Radio Experiment Design for the Space Communications and Navigation (SCaN) Testbed

[1]Asif Shanidullah, [1]Hamed Asadi, [1]Haris Volos, [2]Bo Ryu, [1]Tamal Bose, [3]Richard Reinhart, and [3]Janette Briones
[1]Dept. of Electrical and Computer Engr., The University of Arizona, Tucson, AZ
{asif, hasadi, hvolos, tbose}@email.arizona.edu
[2]EpiSys Science, Inc., Poway, CA
boryu@episyscience.com
[3]NASA Glenn Research Center, Cleveland, OH
{ richard.c.reinhart, janette.c.briones}@nasa.gov

***Abstract*—Variable channel conditions present a challenge to traditional communication systems that have to balance reliable delivery of information and data throughput. Reliable communication schemes that can operate in with a low signal-to-noise ratio typically have a low data throughput. Channel conditions may sometimes can support higher throughput communication schemes. Typically, traditional communication systems cannot leverage these momentary conditions since they are often configured for the worst-case scenarios. This paper proposes a configurable radio platform controlled by a cognitive engine algorithm that optimizes the throughput given different channel scenarios. This system can ensure reliable data delivery in noisy channel conditions, while exploiting channel conditions conducive to higher throughput. The target application for our work is space communications. Specifically the NASA's Space Communications and Navigation (SCaN) Testbed. Our goal is to adapt our cognitive link adaptation algorithms to be tested on the SCaN testbed. To this end, we have begun the implementation and testing of our algorithms on a GNU radio based platform that we developed specifically for the needs of the project. It was found that the cognitive radio platform could maximize data throughput in noiseless systems, while choosing a reliable communication scheme for noisy situations. The work presented in this paper, details the initial steps in the implementation of a larger system that will use cognitive engines and channel profiling to intelligently choose between transmission configurations to maximize data throughput.**

I. Introduction

Communications with the International Space Station (ISS) is limited to noise resistant modulation schemes, such as binary phase-shift keying (BPSK) that the channel conditions may support. The unpredictability of the channel between ground stations and the ISS requires using a modulation scheme capable of operating with a low signal-to-noise ratio (SNR). Lower modulation schemes, however, limit the data throughput. A limited visibility duration between the ISS and ground stations, because of the flight path of the ISS, exacerbates the low throughput issue.

Although, channel conditions may require using lower modulation schemes sometimes, momentary channel conditions may allow using higher modulation schemes, with higher SNR requirements, such as 8PSK or 16QAM. The greater throughput of these schemes allow a larger volume of data to be transmitted during brief transmission windows. The ability to switch the modulation scheme during real-time communications would address low channel throughput, as when the channel can support a higher modulation scheme, more data would be pushed through.

To decide on the modulation scheme and forward error correction (FEC), a cognitive engine (CE), an algorithm that makes decisions based on several input parameters [1] may be used. Typical CE implementations initialy explore the available configurations available to it, i.e., combinations of modulation schemes, coding, transmission power, etc., and profiles the performance of these configurations [1]. The exploration phase concludes once the CE has *blindly* learned, as there is no intelligence associated with its decision other than its past experience, the best configuration for the given input parameters. The convergence time, i.e., the time taken for the CE to become confident of its decision, is one metric used to measure the performance of the CE [2], [3]. Varied input parameters cause a CE to leverage its past experience to form a new decision, though the CE may choose to continue exploiting its past experience because of "greedy" behavior.

A layer of intelligence can be added by adding a metacognitive engine (Meta-CE). The Meta-CE may choose a CE that best suits the current channel conditions (or lack of knowledge thereof), e.g., for a channel whose conditions are unknown, the Meta-CE may choose a CE that explores all available configurations before exploiting its knowledge, allowing the channel to be profiled.

The ability of a Meta-CE communications system to profile the channel, and adapt to its conditions presents a solution to the low throughput, though noise-resistant, communication scheme currently implemented to communicate with the ISS [4]. NASA is seeking a communication system that leverages the Space Communication and Navigation (SCaN) Testbed, developed at the Glenn Research Center in Cleveland [5]. The SCaN Testbed is an experimental software define radio (SDR) platform designed to develop communication, navigation, and networking schemes that utilize SDR techniques to alter the radio characteristics during operation and optimize communication metrics, e.g., throughput.

Research on SDR and cognitive techniques are critical to the

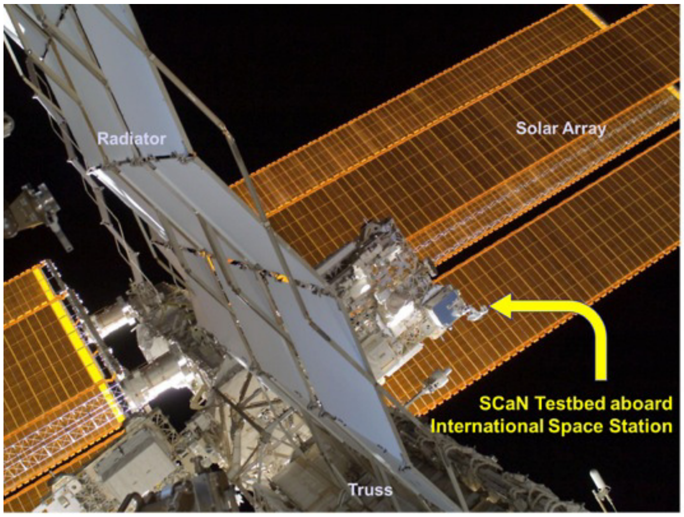

Fig. 1. The SCaN Testbed, affixed onto the ISS.

advancement of communication capabilities in limiting channels. The development of adaptable systems presents solutions not only to the ISS communication issue, but also to deep space communications, and many situations where channels can vary between supporting lower (high traffic times) and higher (low traffic times) modulation schemes. While initially this research may be used in specialized applications (such as the SCaN Testbed), the rapid growth of wireless communications in today's society may require solutions leveraging these intelligent communication systems.

This paper presents the results of an SDR platform whose radio characteristics are controlled by different CEs. The prototype system was developed to showcase the capabilities of different cognitive engines, and the effects on data throughput in ideal and noisy channel conditions. Our goal is to continue developing our methods and to port and test our methods on the SCaN testbed.

II. Background

A. The SCaN Testbed

The SCaN Testbed is an experimental communications system that uses software-defined radio (SDR) technology and provides the capability for S-Band, Ka-Band, and L-Band communication with space and ground assets. Launched in 2012, the SDR testbed has logged over 2500 hours of investigation into SDR waveform software development, reconfiguration, and on-orbit operations. The focus of the SCaN Testbed has matured from investigation of SDR technology and the Space Telecommunications Radio System (STRS) radfio architecture to the application of adaptive and cognitive technologies for NASA space science and exploration missions. The SCaN Testbed flight system consists of three software defined radios provided by government and industry partners. Each of the software defined radios has an STRS Operating Environment (OE), which includes an operating system and infrastructure services to applications and waveforms in accordance with the STRS Standard (Ref: NASA-STD-4009). The OE middleware abstracts the SDR hardware from the waveform application software. In addition to the OE, each SDR runs waveform applications, also compliant to STRS, which implement the unique capabilities of the radio to receive and transmit radio frequency (RF) signals. The flight system communicates with NASAs space network infrastructure of orbiting relay satellites or can communicate directly to ground station compatible with its frequency plan. The cognitive application experiments proposed and described in this report will exercise communication links to the relay satellites and to the ground station. Adaptive and cognitive responses to atmospheric effects between space system and ground station can be studied using the space to ground connection at S-band, while relay satellite connections (at S-band and Ka-band frequency), which last longer and can be tailored to emulate direct to ground passes, are better suited to characterize cognitive engine behavior and performance. As a completely reconfigurable testbed, the SCaN Testbed provides experimenters an opportunity to develop, test, and demonstrate advanced communications, networking, and navigation technologies and to advance the understanding of operating SDRs in space. The prosed cognitive experiments align well with NASAs interests in applying intelligent system behavior to operational systems to reduce cost and human intervention, and better manage the system and communications complexity introduced with flexible and reconfigurable software defined radios

B. Cognitive Radio Engines

The scope of Cognitive Radio (CR) research is to develop radios with adaptive capabilities that are facilitated using artificial intelligence (AI) techniques and other computer science concepts. According to the CR pioneer, Joe Mitola, an ideal CR is capable of not only optimally utilizing its own wireless capabilities, but also self-determining its goals by observing its human operator's behavior [6]. Current CR devices are capable of efficient spectrum utilization and optimized performance in challenging conditions. To make CR possible, communication engineers have, in the last few years, borrowed ideas from machine learning and AI [7]. A cognitive engine (CE) is an intelligent agent who enables the radio to have the desired learning and adaptation abilities. This intelligent agent [1] senses its environment (the wireless channel), acts by using a communication method based on its experience, and observes its own performance to learn its capabilities, adding to its experience base. The work of Rieser, Rondeau, and Le [8]–[10] propose a CE that deals with the user, policy, and radio domains. Their designs are similar and based on the Genetic Algorithm (GA), cased-based reasoning (CBR), and multi-objective optimization principles. He et al. [11] designed a CBR-based CE for IEEE 802.22 wireless regional area network (WRAN) applications and also investigated the radio and policy domains. Other works have focused only on the radio domain. For example, Newman et al. [12] and Z. Zhao, et al. [13] applied a GA and particle swarm optimization, respectively, to multi-channel radio links. On the other hand, N. Zhao et al. [14] proposed a CE design based on ant colony optimization. Zhang and Weng designed a CE with dynamic resource allocation [15], and Y. Zhao et al. [16] looked into utility function selection for streaming video with a CE testbed. Finally, for learning and optimization of a wireless link, Baldo and Zorzi [17] applied an artificial neural network (ANN) and Clancy et al. [18] used predicate logic.

Different types of aforementioned CEs have their own advantages and disadvantages. Providing predictable and higher confident performance level is the most important aspect of designing various CE algorithms at all times. Therefore, the metacognitive radio engine [3] is being proposed to provide the mentioned level of prediction for distinct types of CEs. The first effort was Gadhiok *et al.* [19], who proposed a very primitive architecture of metacognition. In their work they state, "when using a case-based reasoning (CBR) learning framework, metacognition may classify the CBR as an infant, child, or adult, based on the level of learning achieved. Moreover, more advanced and generic metacognitive engine is proposed by the authors [3], which is able to classify various CE algorithms based on the operating conditions (objective, channel condition, radio capabilities, etc.). The proposed meta-CE employs a generic performance characterization method to evaluate the performance of individual CE algorithms. Also, the meta-CE can identify distinct operating scenario based on the performance level of CEs.

III. System Scope

The proposed system consists of a reconfigurable digital communication layer capable of transmitting using a variety of modulation and coding schemes. The permutations are controlled by different CEs, where each CE is tuned to explore a particular channel scenario. The specific CE algorithm in use will be determined by a Meta-CE that bases it decision on the current channel scenarios, spectrum availability, and geographic location of the communicating peers.

The use of a Meta-CE is what introduces a measure of "intelligence" into the system. Each CE *reacts* to the current channel conditions, adapting the system configuration to maximize throughput; this behavior is, by definition, blind. A Meta-CE is *proactive*, as it chooses a CE that it believes will adapt rapidly to the current conditions. For instance, if transmission was occurring in a region that the Meta-CE had no prior knowledge of, it would be more likely to choose a CE algorithm that rapidly changed its transmission configuration, allowing more exploration of the channel.

The ability of the system to leverage its past experiences (through the CE), and current knowledge of the channel (through the Meta-CE) allows for the system to make balanced decisions. In terms of use, such a system could be deployed in a variety of situations without prior training and knowledge, and after a learning curve, would optimally transmit without human interaction. In other words, the proposed system could be deployed for terrestrial communications in a spectrally noisy region, such as city, and an identical system could be deployed for space communications. In both cases, the system would optimally transmit in the respective regions without considerations from an external operator.

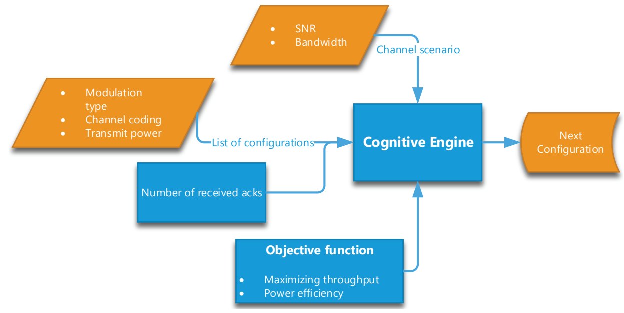

Fig. 2. Cognitive Radio Framework

IV. Cognitive Radio Engine Implementation

This section describes the our implementation of CE methods. First, Figure 2, illustrates our implementation of a CR framework, where the Cognitive Engine block denotes a Meta-CE that selects different CE algorithms. In this architecture, the Meta-CE receives channel scenario metrics and list of all possible configurations based on the radios capabilities, and is aware about the capabilities of different CEs to face various conditions. Additionally, the Meta-CE receives an objective function from operator to be able to evaluate past decisions the defined objective. As an output, the CE selected by the Meta-CE will determine next set of configurations to transmit.

In the rest of this section, we present three different CE algorithms. The algorithms are ϵ-Greedy [20], the Boltzmann exploration [20], and the Gittins Index strategy [21]. All the techniques have two things in common: firstly, they are based on stochastic principles, and secondly, they all have a factor that affects the exploration rate.

A. The ϵ-Greedy Strategy

The ϵ-greedy strategy [20] is a simple strategy that uses (i.e., exploits) the best method (with highest reward) $1-\epsilon(\epsilon \in [0,1])$ of the time (greedy). However, with probability ϵ it explores by using a random method, k, uniformly selected. As $(n \rightarrow \infty)$ by the law of large numbers the mean of rewards is going to converge to the true mean. The ϵ-greedy methods guarantee that all the options are explored as the horizon tends to infinity. The ϵ parameter controls how fast exploration is performed. A higher ϵ will cause a faster exploration and arrive more quickly at an optimal or near-optimal option. However, the high exploration rate may cause reduced overall returns because of the higher exploration cost. The strategy described here is the classic version of the strategy. An issue with this version is that exploration never stops. For this reason, there is a variation that slows down as the number of trials increases. This is especially important when the search space has many significantly underperforming methods. The exploration parameter is updated at every trial n by:

$$\epsilon = \frac{\epsilon_0}{1 + n d_\epsilon} \tag{1}$$

where ϵ_0 is the initial value of ϵ, and d_ϵ the decrease rate. Another variation takes into account the prior knowledge about each method's potential. We know the maximum potential

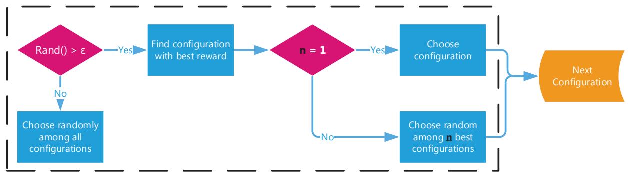

Fig. 3. ϵ-Greedy Algorithm

return of each option (capacity) and we also know the upper bound of the capacity that can be achieved under the current channel. Therefore, we restrict the exploration to machines that potentially can outperform the current k_{greedy}.

Figure 3 presents the flow graph of ϵ-greedy implementation as a CE algorithm. In first step, we create a random variable uniformly between 0 and 1. If the generated random value is greater than ϵ, CE will go to exploitation phase, otherwise, CE will run exploration phase and will choose a configuration randomly. In exploitation phase, if we CE faces more than one configuration as the best one, it will choose one of them randomly.

B. Boltzmann Exploration

Boltzmann exploration [20] weighs its actions based on their estimated value (i.e., methods with a higher value are more likely to be selected.) Each method is selected with probability p_k given by:

$$P_k = \frac{e^{\overline{\mu}_k(n)/T}}{\sum_i e^{\overline{\mu}_i(n)/T}} \tag{2}$$

where T is a positive parameter called the temperature. When the temperature is high (T >1000), the methods are selected probabilistically based on their values $p_k \approx \overline{\mu}_k(n)/\sum_{i=1}^{K}\overline{\mu}_i(n)$. That is, methods with a higher estimated value $\overline{\mu}_k(n)$ are more likely to be selected. However, as $T \to 0$, $p_{k_{max}} \to 1$, where $k_{max} = \text{argmax}_k\,\overline{\mu}_k(n)$. Therefore, as the temperature gets lower in value, the Boltzmann exploration disproportionately selects methods with higher value (i.e., it becomes greedier) and when $T \approx 0$ it only selects the method with the highest estimated value. The temperature T is updated at each trial n using

$$T = \frac{T_0}{1 + nd_T} \tag{3}$$

where T_0 is the initial value of temperature and d_T the decrease rate.

Figure 4 provides a flow graph of Boltzmann strategy implementation. First, CE updates temperature value (T) based on 3. Then, CE will updates all of the probabilities based on the modified value of T and new observations. Finally, CE chooses a configuration randomly by using an the updated probabilities of each configuration.

C. The Gittins Index Strategy

Gittins proved that exploration vs. exploitation can be optimally balanced using a dynamic allocation index-based strategy [39]. This strategy maximizes the total sum of rewards

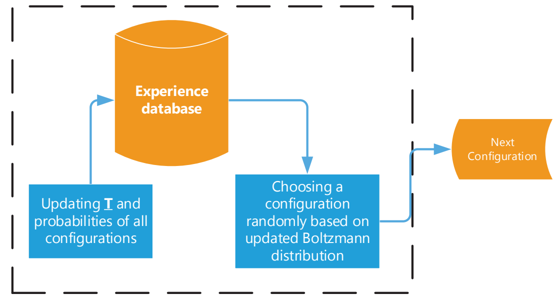

Fig. 4. Boltzmann Exploration Algorithm

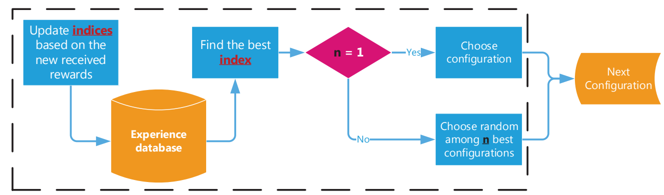

Fig. 5. Gittins Index Algorithm

collected over a long-term horizon. The strategy is simply to use the method with the highest Gittins index, which is based on the reward statistics of each method and must be estimated only when those statistics change (i.e., only when each method is used). We discuss the use of the Gittins indices in more detail in two of our prior publications [4], [5].

The Gittins index is dependent upon the underlying distribution of Formula. In this work, we consider the Gittins index for the normal reward process (NRP) and the Bernoulli reward process (BRP). In the application examined in this work, the underlying process is Bernoulli - a packet is either successful or unsuccessful. For a NRP, the Gittins index is equal to

$$\nu(\overline{\mu}, \overline{\sigma}^2, n', \gamma) \equiv \overline{\mu} + \overline{\sigma}\nu(0, 1, n', \gamma) \tag{4}$$

where $\overline{\mu}$ and $\overline{\sigma}^2$ are the estimates of the mean and the variance of return, respectively, using n' trials; $\gamma \in (0, 1)$ is a discount factor; and $\nu(0, 1, n', \gamma)$ is the Gittins index for a zero mean, unit variance distributed process (tabulated in Gittins' book [21]). For a BRP, the Gittins index is equal to

$$\nu(\alpha, \beta, \gamma, R_k) \equiv R_k\nu(\alpha, \beta, \gamma, 1) \tag{5}$$

where $\nu(\alpha, \beta, \gamma, 1)$ is the Gittins index for a Bernoulli process (again tabulated in Gittins' book [21]), with α successes and β failures, and a reward of 1, if successful. R_k is the reward received when method k is successful. In the BRP case, the belief state is represented by $\{\alpha, \beta\}$. Other works offer more information on the Gittins index [22], [1], [21].

Figure 5 illustrates how the gittins index strategy is implemented. In this process, first, CE updates experience database by index of recent used configuration based on the received rewards. Next, CE chooses a configuration with the highest index. If CE faces more than one configuration with maximum index, It will choose one of the randomly.

V. Platform Development

This section describes the SDR platform developed to test the CE implementations. The plaform and CEs were developed concurrently; a fixed interface was developed to describe the communication between the two modules. The experimental specifics are also discussed in this section.

VI. Radio Platform

To test the CE implementations, a full-duplex radio platform was written using GNU Radio (GR), an open source software development toolkit. The framework provided a variety of signal processing *blocks*, implemented in C++; a chain of blocks (flowgraph) formed a software platform that replaced the functionality of traditional hardware, e.g. an FM demodulator.

During the development of GR, hierarchical blocks were created to accomplish common tasks, such as packet modulation. These blocks were written in Python and made use of the C++ backend through SWIG. The choice of Python allows radio systems to prototyped and modified rapidly, as well as offering the functionality and flexibility of Python.

Additionally, GR was written to leverage external RF hardware, such as the Ettus Research Universal Software Radio Peripheral (USRP) series, to form software defined radio systems. The USRPs provided a generic radio frontend while GR handled the signal processing specific to the radio application.

The radio platform was constructed by compiling GR 3.7.5.1 from source on two PCs that had Fedora 20 installed as the operating system. A pair of USRP N200s were used as the RF frontend. Universal Hardware Driver 3.8 was compiled from source on the PCs allowing communication with the USRPs. The radio platform itself was written in Python.

Phase-shift keying was implemented as the platforms modulation scheme, supporting 2, 4, and 8 constellation points. Differential and Gray coding were available, allowing for a total of twelve transmission configurations. The system was designed to allow the modulation scheme to be switched during operation. The platform also implemented Reed Solomon forward error correction at ratios of $^1/_8, ^1/_4, ^3/_8, ^1/_2, ^5/_8, ^3/_4, ^7/_8$ and 1. However, due to the implementation specifics, the coding scheme was not used during system operation. Signals were fixed to a 400 kHz bandwidth.

To simulate channel scenarios, the radio platform was equipped with an AGWN adder on the receiver side. This allowed noise to be added to the incoming signal, reducing the SNR. This feature was used to force the CE to adapt to noisy conditions.

VII. System

The cognitive engine and radio platform were implemented as two separate applications. A shared SQL database was used to communicate between the two processes during runtime. The database contained the transmission statistics, provided by the radio platform at periodic intervals. The cognitive engine used the statistics to base its decisions. As implemented, the radio platform provided the data throughput as the measurement statistic, consistent with the CE objective. During operation,

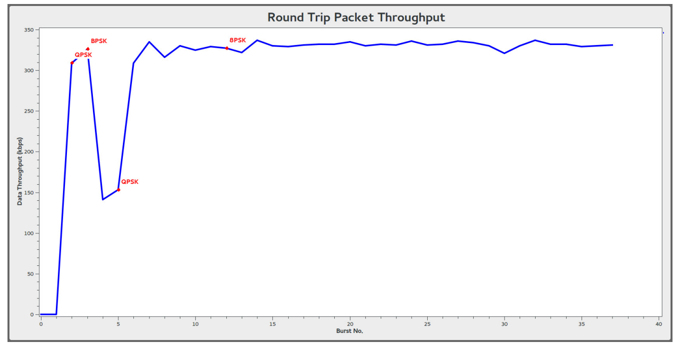

Fig. 6. Throughput vs Frame Number plot for Epsilon-Greedy CE showing convergence to 8PSK after exploration in noiseless channel.

the database was updated by the CE with the next transmission configuration to be used by the radio platform.

The radio platform was written to continuously transmit 1000 packet frames, where each packet had a 376 bit payload. At the end of each frame, if a new transmission configuration was chosen by the CE, the transmitter notified the receiver of the new scheme and both systems automatically reconfigured in preparation for the next frame.

The throughput of the system was defined as the total number of packets acknowledged by the receiver over the time taken for the last acknowledgment to be seen by the transmitter. Only packets that passed a CRC were acknowledged by the receiver. A three second time to live was given to the last packet defining the time window for each frame.

VIII. Results

Each CE starts with no knowledge of the channel, i.e. the experience table in the database is reset each time the system is run. Figure 6 demonstrates the Epsilon-Greedy CE acclimatizing to a noiseless channel. It is seen that after a brief exploration phase, 8PSK is chosen to maximize the data throughput. The exploration phase lasts 10 frames (which corresponds to 10,000 packets). The maximum throughput achieved by the system is 350 kbps. Figure 7 shows the constellation plot at the receiver side with an SNR of 40 dB. The apparent spread of the constellation points in the figure are a result of the automatic gain control and the window zoom.

In a noisy environment, the CEs adapt to more noise-resistant modulation scheme. Figure 8 shows the reduced throughput of the system, roughly 40 kbps, due to the high noise floor, while 9 shows the spread of the constellation points. This is a result of adding noise to the system to reduce the SNR to 10 dB.

The exploitative nature of the CEs in situations where multiple configurations demonstrate similar performance is demonstrated in Figure 10. The figure shows the Gittins CE exploring different two main configurations, QPSK and 8PSK as the channel conditions do not favor either modulation scheme. The two valleys in the figure are a result of the Gittins engine electing to explore a BPSK modulation scheme. Figure 11 shows the noisy channel conditions that favor neither QPSK nor 8PSK. The low SNR increases the BER for 8PSK, reducing its throughput similar to QPSK.

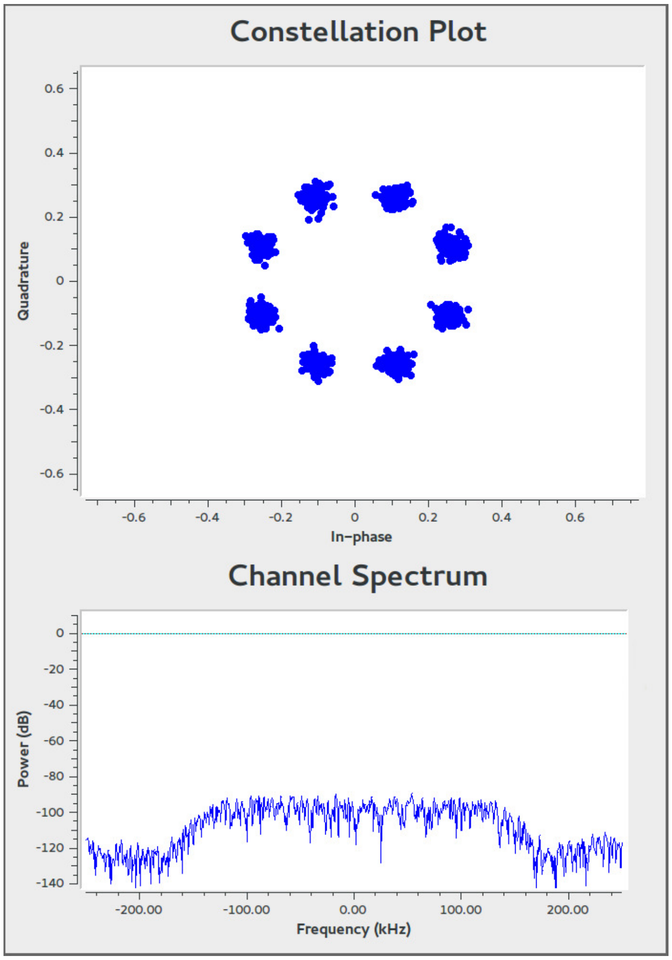

Fig. 7. Constellation plot at receiver for noiseless channel for Epsilon-Greedy 8PSK convergence.

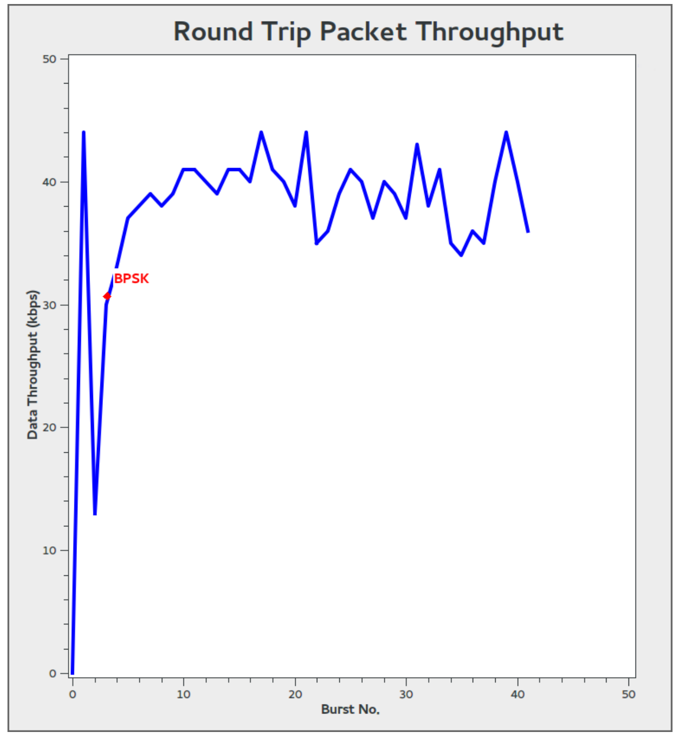

Fig. 8. BPSK convergence in a noisy channel.

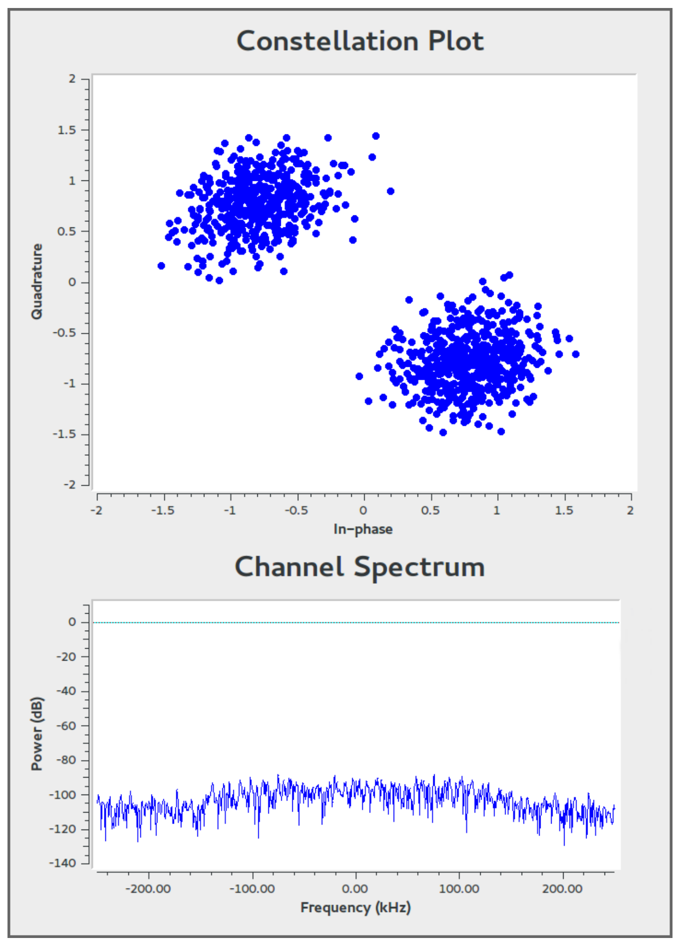

Fig. 9. Constellation plot at receiver for noisy channel.

IX. Future Work

Our immediate plans is to proceed with integrating our Meta-CE [3] work with the radio platform. Based on the observed channel conditions, which define the radio environment, the Meta-Cognition Module sets the operational parameters (channel, BW, power, corresponding station, and applicable link options such as modulation and coding) of the waveform. From Figure 12, the Meta-Cognition Module profiles each CE algorithm's adaptation performance at different operating scenarios, amd chooses the most suitable CE algorithm for the current operating conditions. The Shared Memory stores historical data of the previous adaptations, which is available to each CE algorithm.

Next, we plan to integrate our modulation classification work in order reduce the need for header information in our transmitting packets and for enhancing our radio environment map algorithms that we also plan to integrate to the platform. Modulation classification would allow the system to profile the incoming signal and choose the demodulation scheme that matches the signal profile.

By integrating RF Mapping into the communications the platform, the Meta-CE would be provided a map of spectral usage in the geographic area of operation for the communi-

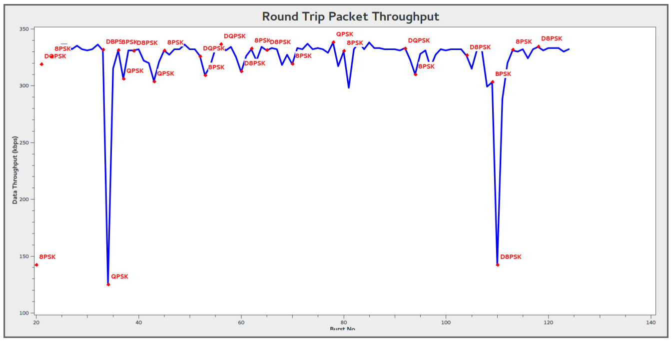

Fig. 10. Gittins CE continuously exploring different configurations when one configuration is not clearly better than others.

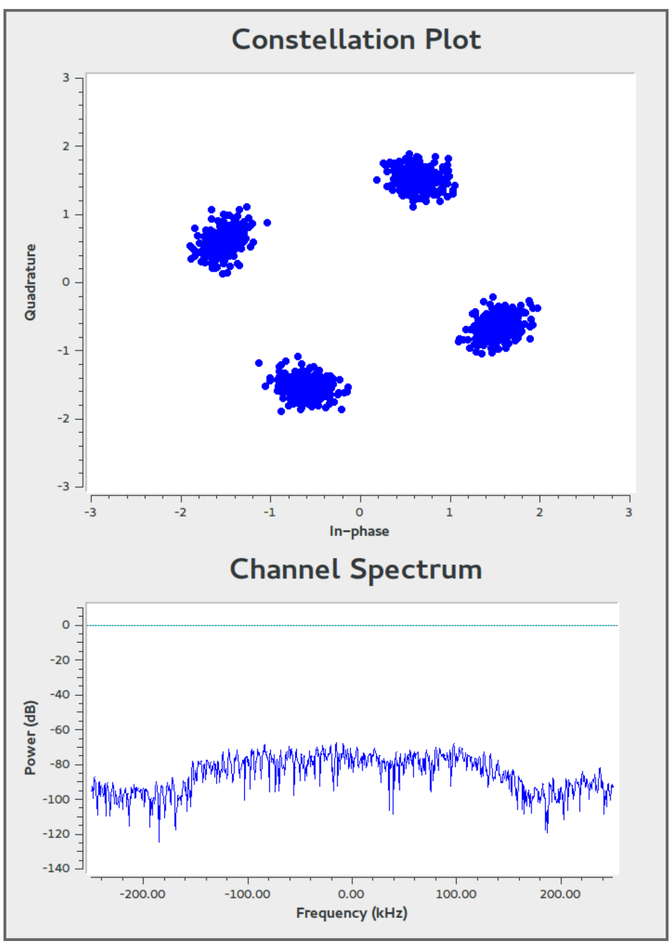

Fig. 11. Receiver side constellation plot during exploration for a noisy channel.

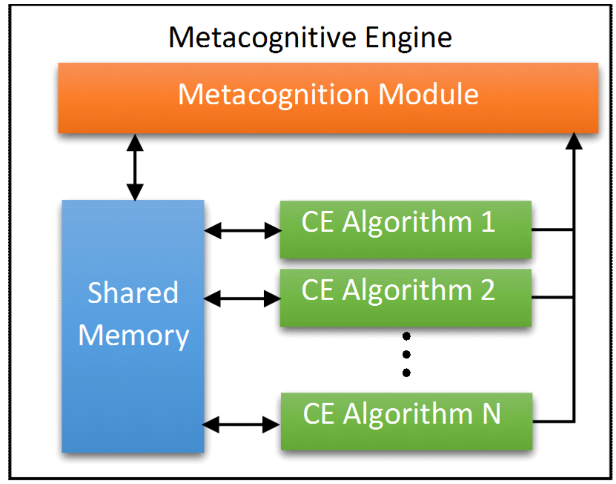

Fig. 12. Metacogntive Engine Diagram

cations system. The RF Mapper will map and estimate the signal availability of the various links taking account the expected orbits. Using this knowledge, the CE will be to more accurately select the appropriate methods for the current and anticipated signal levels and reduce the adaptation convergence time.

We also plan improvements on our current platform such transitioning to C++ (versus Python) and use of a custom packetizer in lieu of the stock GNU Radio packetizer that will be more streamlined for our application and increase flexibility.

Finally, a proposal is being put forward to NASA to showcase the operation of the Meta-CE communications system running on the SCaN Testbed.

X. Conclusions

The prototype communication system demonstrated that the use of a CE with an SDR platform improves the performance of a communication system that uses a fixed modulation scheme with a conservative link margin to cover any anticipated fluctuations. It has been demonstrated that given ideal channel conditions (high SNR) the CEs would maximize throughput. Conversely, with lower SNR channels would cause the CEs to select lower modulation schemes such as BPSK. The ability of the system to adapt to the channel conditions to optimize a specific metric demonstrates the fundamental characteristic desired in an intelligent communication system.

Future work concerning the system will constitute implementing a Meta-CE to control which CE is utilized adding a measure of intelligence to the learning process of the system, allowing the system to habituate rapidly to the channel as the convergence time latency of CEs prevents rapid adaption. Incorporating modulation classification would allow the communicating peers to be *blind*, i.e. no forward declaration of which configuration currently employed by the system would be required, resolving issues of loss of synchronization between peers. RF Mapping would provide the Meta-CE with some prior knowledge of the channel conditions, eliminating an aggressive exploration phase for CEs reducing convergence time.

XI. Acknowledgments

This project was supported the National Aeronautics and Space Administration (NASA) Glenn research center and by the Broadband Wireless Access & Applications Center (BWAC); NSF Award #1265960.

References

[1] H. I. Volos and R. M. Buehrer, "Cognitive Engine Design for Link Adaptation: An Application to Multi-Antenna Systems," *IEEE Transactions on Wireless Communications*, vol. 9, no. 9, pp. 2902–2913, Sept. 2010.

[2] ——, "Cognitive Radio Engine Training," *Wireless Communications, IEEE Transactions on*, vol. 11, no. 11, pp. 3878–3889, 2012.

[3] H. Asadi, H. I. Volos, M. Marefat, and T. Bose, "Learning Characterization Framework and Analysis for a Meta-Cognitive Radio Engine," in *Proceedings of SDRWInnComm 2014 Wireless Innovation Conference on Wireless Communications Technologies and Software Defined Radio*, Mar. 2014, pp. 132–139.

[4] "International Space Station," http://www.nasa.gov/mission_pages/station/main/, accessed: 2014-12-19.

[5] "SCaN Testbed," spaceflightsystems.grc.nasa.gov/SOPO/SCO/SCaNTestbed/, accessed: 2014-12-19.

[6] J. Mitola, III, "Cognitive Radio: Model-Based Competence for Software Radio," Licentiate Thesis, The Royal Institute of Technology (KTH), Stockholm , Sweden, August 1999.

[7] A. He, K. K. Bae, T. R. Newman, J. Gaeddert, K. Kim, R. Menon, L. Morales-Tirado, J. 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, pp. 1578–1592, May 2010.

[8] T. W. Rondeau, "Application of Artificial Intelligence to Wireless Communications," Ph.D. dissertation, Virginia Tech, 2007.

[9] C. J. Rieser, "Biologically Inspired Cognitive Radio Engine Model Utilizing Distributed Genetic Algorithms for Secure and Robust Wireless Communications and Networking," Ph.D. dissertation, Virginia Tech, 2004.

[10] B. Le, T. W. Rondeau, and C. W. Bostian, "Cognitive Radio Realities," *Wiley Journal on Wireless Communications and Mobile Computing*, vol. 7, no. 9, pp. 1037–1048, 2007.

[11] A. He, J. Gaeddert, K. Bae, T. R. Newman, J. H. Reed, L. Morales, and C. Park, "Development of a Case-Based Reasoning Cognitive Engine for IEEE 802.22 WRAN Applications," *ACM Mobile Computing and Communications Review*, vol. 13, no. 2, pp. 37–48, 2009.

[12] T. R. Newman, B. A. Barker, A. M. Wyglinski, A. Agah, J. B. Evans, and G. J. Minden, "Cognitive Engine Implementation for Wireless Multicarrier Transceivers," *Wiley Journal on Wireless Communications and Mobile Computing*, vol. 7, no. 9, pp. 1129–1142, 2007.

[13] Z. Zhao, S. Xu, S. Zheng, and J. Shang, "Cognitive Radio Adaptation Using Particle Swarm Optimization," *Wireless Communications and Mobile Computing*, vol. 9, no. 7, pp. 875–881, 2009.

[14] N. Zhao, S. Li, and Z. Wu, "Cognitive Radio Engine Design Based on Ant Colony Optimization," *Wireless Personal Communications*, pp. 1–10, 2011.

[15] C.-H. Jiang and R.-M. Weng, "Cognitive Engine with Dynamic Priority Resource Allocation for Wireless Networks," *Wireless Personal Communications*, vol. 63, no. 1, pp. 31–43, Mar. 2012.

[16] Y. Zhao, S. Mao, J. Reed, and Y. Huang, "Utility Function Selection for Streaming Videos with a Cognitive Engine Testbed," *Mobile Networks and Applications*, vol. 15, pp. 446–460, 2010.

[17] N. Baldo and M. Zorzi, "Learning and Adaptation in Cognitive Radios Using Neural Networks," in *5th IEEE Consumer Communications and Networking Conference*, Jan. 2008, pp. 998–1003.

[18] C. Clancy, J. Hecker, E. Stuntebeck, and T. O'Shea, "Applications of Machine Learning to Cognitive Radio Networks," *IEEE Wireless Communications*, vol. 14, no. 4, pp. 47–52, August 2007.

[19] M. Gadhiok, A. Amanna, M. J. Price, and J. H. Reed, "Metacognition: Enhancing the Performance of a Cognitive Radio," in *2011 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*. IEEE, Feb. 2011, pp. 198–203.

[20] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. The MIT Press, March 1998.

[21] J. C. Gittins, *Multi-Armed Bandit Allocation Indices*. Wiley, Chichester, NY, 1989.

[22] H. I. Volos and R. M. Buehrer, "On Balancing Exploration vs. Exploitation in a Cognitive Engine for Multi-Antenna Systems," in *Proceedings of the IEEE Global Telecommunications Conference*, Nov. 2009, pp. 1–6.