# An Overview of the Architecture of an Autonomous Radio

J. Hamkins,[1] M. Simon,[1] S. Dolinar,[1] D. Divsalar,[1] and H. Shirani-Mehr[1]

*This article is an overview of the architecture of an autonomous radio that can receive a signal without much a priori knowledge about its defining characteristics. We show that this may be accomplished by estimating and classifying the relevant properties directly from the observed signal, enabling the receiver to configure itself. We describe the function of each estimator/classifier module and how the modules can interact to ultimately recover the transmitted message.*

## I. Introduction

We have embarked on an ambitious project to develop new technology for a radio to receive a signal without much a priori knowledge about its defining characteristics. As a first step in this direction, we have developed a suite of modules that enable autonomous recognition of various signal attributes, including the angle of arrival, data rate, symbol timing, carrier frequency and phase, modulation index, modulation type, signal-to-noise ratio (SNR), code type, and decoded message bits. This article is an overview of the architecture of the autonomous radio, describing what each module does and how the modules interact to produce the desired effect.

The primary application of this technology is in relaying communication signals from multiple deep-space assets. For example, we might desire two or more rovers on a distant planet to relay data through an orbiter, as the two Mars Exploration Rovers have done via Mars Global Surveyor and Mars Odyssey [1]. Multiple landed assets communicating through relays will continue to be an important part of NASA's exploration plans throughout the next two decades [2,3]. Over a period of years, we may expect NASA and other space agencies to launch a set of diverse orbiters and landers,[2] and because technology continues to emerge, it is unlikely that they will all use the same data rates, protocols, error-correcting codes, and modulation types.

The advantage of an autonomous radio in this emerging scenario is that it can communicate to each asset that comes into view, automatically, without having to be reconfigured from Earth for each pass to account for differences in the signal characteristics. The radio would receive whatever each landed asset sent. Since an orbital period may be a few hours, an orbiter may come within view of various landed

[1] Communications Architectures and Research Section.

[2] See `http://www.jpl.nasa.gov/missions/future_missions.cfm`

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

assets several times a day, and the automatic reconfiguration would be this frequent. By comparison, manual radio reconfiguration would be a daunting task.

In addition to easing the scheduling and configuration burden, an autonomous radio also will gracefully handle unpredictable or anomolous events. For example, during entry, descent, and landing (EDL), a spacecraft can undergo large Doppler swings caused by rocket firings, parachute openings, backshell ejection, and a bouncing landing on the surface. Even when all scheduled events occur successfully, there may be Doppler uncertainty due to unpredictable properties of the atmosphere. Ideally, the communication link should operate whether or not each of the EDL events is successful, but the uncertainties involved typically lead to liberal link margins—for example, the Mars Exploration Rovers observed link margins that sometimes exceeded 10 dB. An autonomous radio could substantially reduce this margin because it would handle any Doppler swing near optimally.

Unfortunately, such flexible technology is not available on NASA's currently flying missions. In perhaps the most glaring example of this, NASA engineers discovered in 2000 that a receiver aboard Cassini, launched in 1997, would fail during the Huygens probe descent onto Titan because it did not properly account for the Doppler profile of the probe. Increasing the loop bandwidth of the synchronization loops would have easily fixed the problem, but, unfortunately, these loop bandwidths were hard wired to set values on the spacecraft. NASA/the European Space Agency (ESA) resorted to forming a large and expensive (hundreds of $K) international recovery team to make recommendations on how to overcome the radio's severe limitations.

## II. Preliminaries

### A. Signal Model

This article considers a single channel amplitude and phase modulated signal with or without a residual carrier. In complex baseband, the transmitted signal has the form

$$\tilde{s}(t) = \sum_{n=1}^{N} A_n p\big[t-(n-1)T\big] e^{j[\theta_n+\theta_c(t)]} + \sqrt{P_c}e^{j\theta_c(t)} \tag{1}$$

where $A_n$ is the amplitude of the $n$th symbol; $\theta_n$ is the phase modulation for the $n$th $M$-phase-shift keying ($M$-PSK) symbol; $\theta_c(t)$ is the carrier phase; $p(t)$ is a pulse shape satisfying $T^{-1}\int_0^T p^2(t)dt = 1$; $T$ is the symbol duration; and $P_c$ is the residual carrier signal power.

When the amplitude is not modulated, as is the case with phase-shift keying (PSK), we may write $A_n = \sqrt{P_d}$, a constant equal to the data modulation signal power. In the special case in which $\theta_n = c_n\pi/2$ with $c_n \in \{-1, 1\}$ and $p(t)$ a digital pulse shape taking on values ±1, i.e., a binary PSK (BPSK) signal, Eq. (1) becomes

$$\tilde{s}(t) = \sum_{n=1}^{N} \sqrt{P_t}p\big[t-(n-1)T\big] e^{j[\beta c_n+\theta_c(t)]} \tag{2}$$

where $P_t = P_c + P_d$ is the total signal power and $\beta = \tan^{-1}\sqrt{P_d/P_c}$ is the modulation index.

At the receiver, the timing and carrier phase are initially unknown, and noise is present. If we also assume that the carrier frequency $f_c$ is imperfectly estimated as $\hat{f}_c$ at the front-end of the receiver, then a residual frequency component $f_r = f_c - \hat{f}_c$ will remain after conversion to baseband, and the resulting signal will have the form

$$\tilde{r}(t) = \tilde{s}(t) + \tilde{n}(t) \tag{3}$$

$$= \sum_{n=1}^{N} A_n p\big[t - (n-1-\varepsilon)T\big] e^{j[2\pi f_r t + \theta_n + \theta_c(t)]} + \sqrt{P_c} e^{j[2\pi f_r t + j\theta_c(t)]} + \tilde{n}(t) \tag{4}$$

where $\varepsilon$ is the unknown symbol timing and $\tilde{n}(t)$ is an additive white Gaussian noise (AWGN) process with one-sided power-spectral density $N_0$ W/Hz. A priori, $\varepsilon$ is uniformly distributed on $[0, T)$ and $\theta_c$ is[3] uniformly distributed on $[0, 2\pi)$.

## B. Anatomy of the Received Signal

Figure 1 graphically indicates the dependence of the received signal on several factors. We group the signal dependence graph into three primary components: the forward error-correcting (FEC) code encoder, the modulator/amplifier, and the channel. Each of these is affected by several subfactors, including the ones shown in Fig. 1 as well as others that we call out in italics in the more detailed discussion below.

**1. FEC Code.** The FEC code can be one of several *code types*. The code types standardized by the Consultative Committee for Space Data System (CCSDS) for deep-space [4] or in situ [5] communications include Reed–Solomon (RS) codes, convolutional codes, turbo codes, Bose–Chaudhuri–Hocquenghem (BCH) codes, and cyclic redundancy-check codes (CRC). Work is also progressing rapidly both academically and in various standards (Digital Video Broadcast/Satellite, IEEE 802.{11n,15.3a,16e}, and CCSDS deep space and in situ) on low-density parity-check codes and progressive-parity-type codes, such as tornado and raptor codes.

Associated with each FEC code is its *code rate*, which is the fraction of symbols carrying information, and its *code length*, which indicates the number of symbols in each codeword. For some code types, these parameters alone are nearly enough to completely identify the code. For example, the best-performing convolutional codes for a given rate and constraint length are tabulated in textbooks (e.g., [6]), and applications using convolutional codes nearly always use codes from these tables. CRC codes of a given length also typically use standard generator polynomials [5]. RS codes are specified by their blocklength, rate, field generator polynomial, and code generator polynomials. The latter two can be one of several possibilities, but in practice space communication systems have primarily used the one that is specified in the CCSDS standard [4].

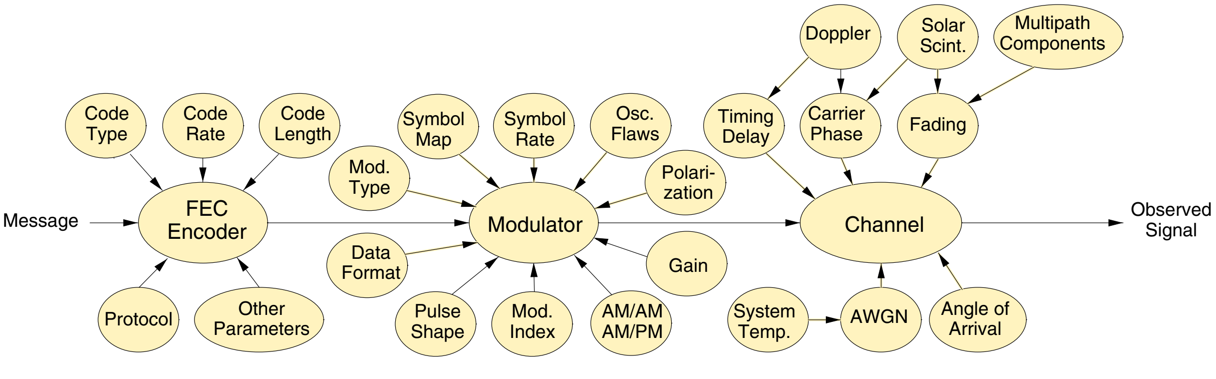


**Fig. 1. Signal dependency graph.**

[3] For our purposes here, we assume that the carrier phase $\theta_c(t)$ is slowly varying with respect to the data rate; thus, we shall hereafter drop the dependence on time in the notation.

The techniques of *puncturing*, *shortening*, and *expurgating* are commonly used to modify a code. Puncturing raises the code rate, and it is the standard technique to obtain the CCSDS turbo codes of rate 1/4, 1/3, and 1/2 from the rate 1/6 mother code [4]. Shortening is often used with the standard RS (255,223) code—the full-length code with interleaving depth 5 has input length $223 \times 8 \times 5 = 8920$, but missions often shorten this to 8800, a multiple of 32, which is a convenient quantity for spacecraft processors to handle. The (63,56) BCH code used for uplink commanding [7] is an expurgated (63,57) Hamming code, with odd-weight codewords removed.

In addition to the substantive factors mentioned above, there are a number of superficial factors that determine the FEC encoder output. The precise *bit ordering*, use of *trellis termination*, and *placement of frame headers, synchronization bits, and filler bits* are examples of these factors.

**2. Modulator and Amplifier.** The modulator uses the coded binary sequence from the output of the FEC encoder to modulate a carrier signal. This process depends on several factors. The *modulation type* identifies the signal constellation from which the transmitted symbols are chosen. BPSK, quadrature PSK (QPSK), quadrature amplitude modulation (QAM), and Gaussian-filtered minimum-shift keying (GMSK) are commonly used modulation types [8]. In the case of GMSK, and other filtered modulation types, the bandwidth time ($BT$) product is also needed to fully specify the modulation.

The assignment of FEC-encoded bits to symbols is defined by a *mapping*, which may be a static mapping—such as a natural ordering, Gray code, or anti-Gray code, which maps each block of bit(s) to a symbol—or the mapping may be dynamically controlled through a state machine, as it is with trellis-coded modulation [9].

The *symbol rate*, or baud, defines the number of discrete signal constellation elements transmitted per second. Within each symbol epoch, a *pulse shape* (rectangular, raised-cosine, etc.) is applied. With BPSK signaling, the *data format* may be non-return to zero (NRZ) or Manchester encoded. The *modulation index* determines the fraction of total power that is allocated to an unmodulated carrier signal.

The carrier signal to be modulated is generated by an imperfect oscillator, whose quality can be measured by its spectrum, or by distilling its spectrum to a single quantity such as *Allan deviation*, *phase noise* at a given offset, or *drift rate*. Ultrastable oscillators can achieve a phase noise of $-100$ dBc/Hz at a 1-Hz offset [10], although not all missions have the mass budget to carry one onboard the spacecraft.

There are several signal-dependent factors in the amplifier as well. Nominally, the amplifier output is larger than the input by the *gain* of the amplifier. However, depending on the input, distortion may affect the amplitude or phase. *Amplitude modulation-to-amplitude modulation (AM/AM) distortion* occurs when the amplitude of the amplifier output is not proportional to the amplifier input. *AM-to-phase modulation (AM-to-PM) conversion* occurs when variations in the input amplitude result in unwanted phase modulation. Additionally, the *group delay* is the rate of change of the total phase shift with respect to angular frequency, and the *polarization* (right- or left-handed, circular or elliptical) describes the time-varying direction and amplitude of the electric field vector propagated from the transmitter.

**3. Channel.** Typically, deep-space communications channels are quite benign, with *additive white Gaussian noise* (AWGN) being the dominating impairment. If *fading* is present, it may be due to *multipath interference* or *solar scintillations* caused by a small Sun–Earth–probe angle. *Doppler* affects carrier and timing parameters. The *angle of arrival*, *symbol timing*, and *carrier phase* also are modeled in the channel component of the dependency graph shown in Fig. 1.

# III. Radio Receiver Architectures

## A. A Conventional Radio Receiver

A functional block diagram of a radio receiver and decoder is shown in Fig. 2. Factors that are known a priori in a conventional radio are shown in ellipses, while the tasks it performs are shown in rectangles. A conventional radio receiver has complete a priori knowledge of the signal-dependent factors relating to the FEC encoder and modulator/amplifier components shown in Fig. 1. Only the channel-related factors are not completely known—although even those may be partially known through the use of predicts.

Knowledge of the transmitted signal parameters greatly simplifies the design and implementation of the receiver. For example, if a residual carrier is present, then the carrier phase tracking loop may be a simple phase-locked loop (PLL), and no Costas loop need be implemented. Or, if the modulation type is known to be BPSK, then the receiver need not include any processing of the quadrature component of the signal. Every rectangular block in Fig. 2 is similarly simplified by knowing the basic properties of the transmitted signal.

On the other hand, a conventional radio usually does not have much capability to receive signal types different from the single signal type for which it was primarily designed, or, when it does have such capability, it requires specific pre-configuration according to a predetermined schedule. For example, if the radio can receive both suppressed-carrier and residual-carrier BPSK signals, it typically would carry both a PLL and a Costas loop, or a hybrid loop that incorporates both components, and would have to be preconfigured to use the appropriate loop (or to set the relative gains in the hybrid loop) based on knowledge of when each type of signal will be arriving. This is the approach taken by the highly capable advanced receiver design [11] that eventually became the Deep Space Network Block V Receiver.

## B. Electra

Electra is NASA's first highly capable software-defined radio [12]. Unlike other flexible radios, the Electra radio is an elegant, compact design based around a reprogrammable radiation-tolerant field programmable gate array (FPGA). The FPGA performs all the baseband processing for reception, including carrier tracking, timing recovery, and demodulation. It also includes all the baseband processing necessary for transmission. A functional block diagram of the receiver portion of the baseband processor module is shown in Fig. 3.

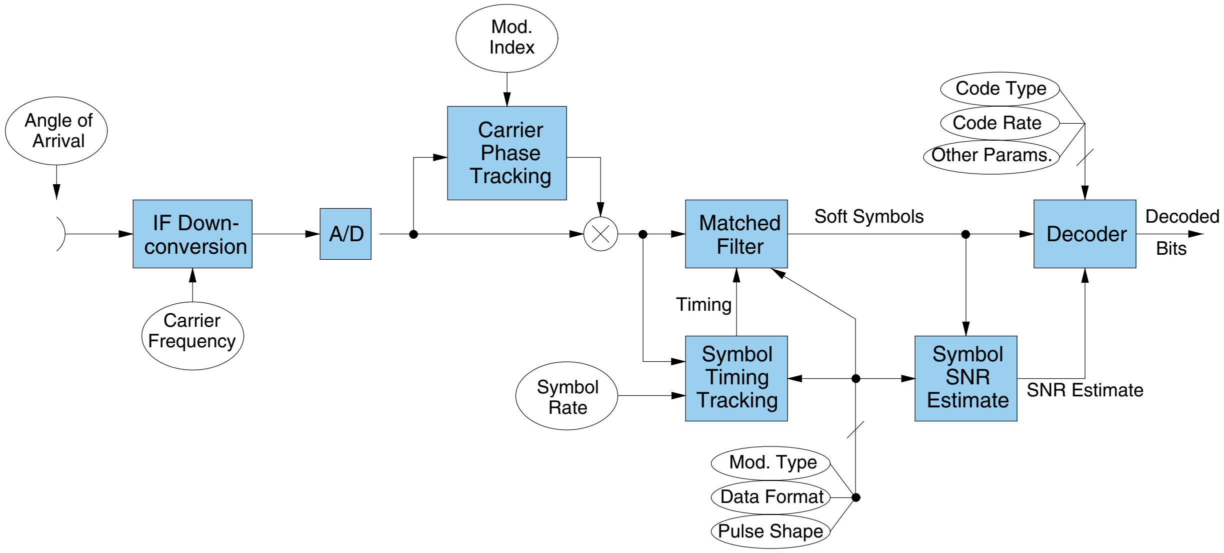


**Fig. 2. Functional block diagram of a radio receiver. A conventional radio assumes a priori knowledge of all parameters shown in the ellipses, while an autonomous radio estimates them based on the incoming signal.**

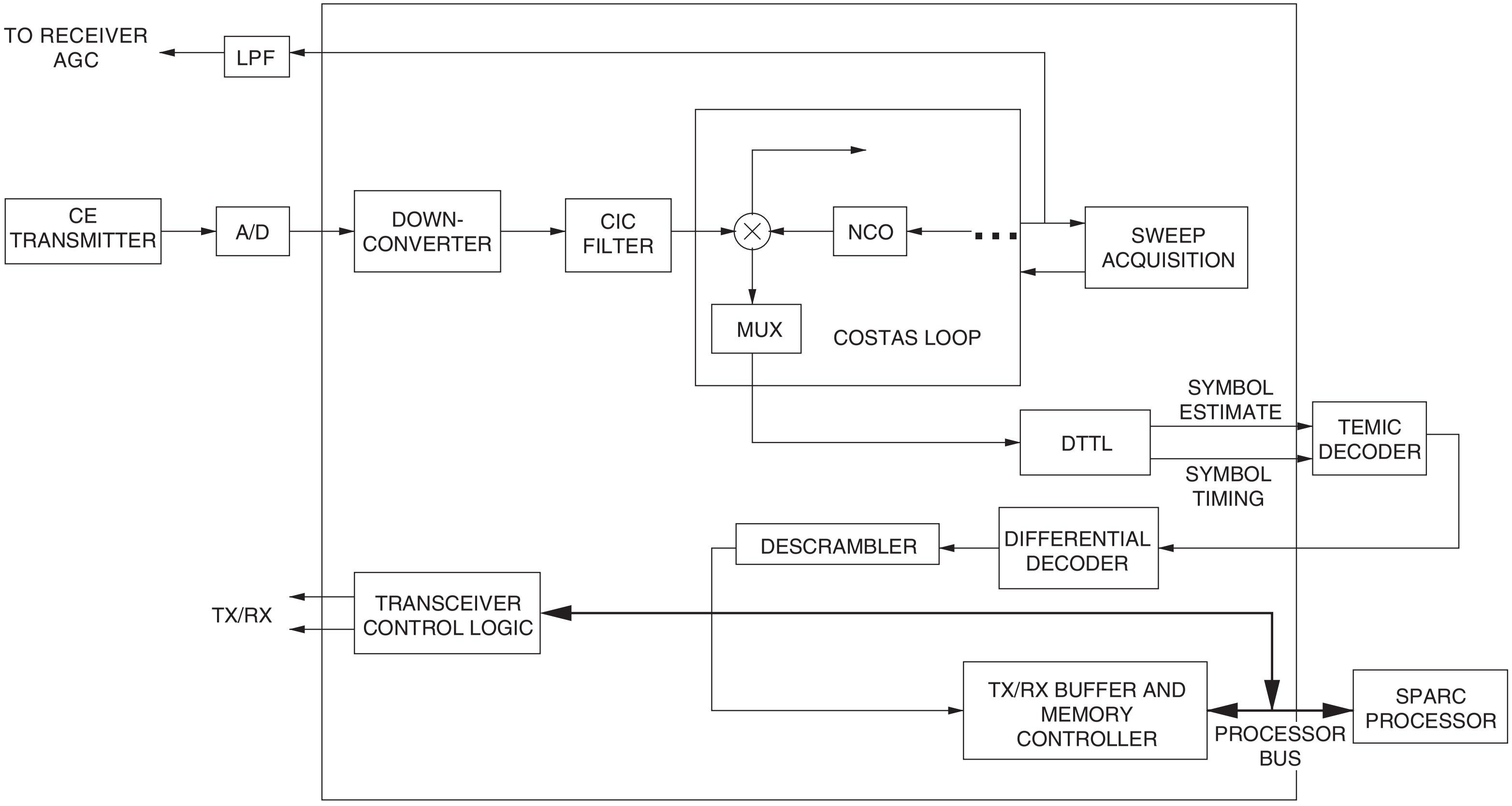


**Fig. 3. Functional block diagram of the receiver portion of the baseband processor module of the Electra radio.**

Unlike the Block V Receiver, the massive capability of this radio is not achieved through multiple simultaneous implementations of tracking loops and demodulators for all the various signal types it might encounter in its lifetime. Rather, the radio is simply redefined in the same small footprint by reprogramming the baseband processor module. This compact, flexible design makes it ideally suited for in situ radios, and, in fact, it is now the NASA standard in situ radio and will fly on the Mars Reconnaissance Orbiter, Mars Telecommunications Orbiter, and Mars Science Laboratory missions, among others.

### C. An Autonomous Radio

The fundamental difference between a conventional radio, or even a software-defined radio such as Electra, and a truly autonomous radio is that an autonomous radio has the ability to recognize features of an incoming signal and to respond intelligently, without explicit pre-configuration or reprogramming to define the functions of the radio.

In an autonomous radio, the parameters shown in ellipses in the functional block diagram in Fig. 2 are assumed unknown a priori and must be determined based on the incoming signal. The quality of each of the estimators and classifiers of the autonomous radio is limited by its lack of knowledge of any of the other parameters. As such, the order in which the estimations/classifications are performed is critical. For example, it would not be feasible to classify the modulation type prior to classifying the data rate and obtaining the symbol timing. Using conventional estimation and tracking designs, one quickly gets into a chicken and egg problem, with nearly every estimator needing the output of the other estimators before it can make a maximum-likelihood (ML) estimate.

To resolve this problem, we have arranged the estimators/classifiers in the partially ordered set shown in Fig. 4, which defines the order in which they may be operated, at least sub-optimally, during the first iteration of estimation. In this ordering, solid arrows indicate a strict dependency, meaning that the estimator at the head of the arrow cannot proceed without the input of the estimator at the tail of the arrow. The estimators are arranged in the minimum number of levels for which some level $i$ estimators/classifiers must be performed before level $i + 1$ classifiers. The number of such levels is the length of the longest solid-arrow path.

The dashed arrows in Fig. 4 indicate additional estimator dependencies that may result in improved performance, but which are not strictly required. For example, the modulation classifier can operate noncoherently, without input from the phase tracking loop, but if it waits for that input, its performance is improved. Using the dashed-arrow ordering, the latency of estimation may increase.

This approach yields a workable boot-strapping approach to estimating/classifying all of the parameters necessary for the proper operation of the entire receiver.

## IV. Estimators and Classifiers of the Autonomous Radio

After identifying the proper order for the estimator modules shown in Fig. 4, we embarked on a design effort for the individual modules. At first glance, it may seem that some of these estimator modules are simply long-established, conventional designs. For example, phase tracking loops have been designed and analyzed for nearly every reasonable signal type. However, we were unable to find any literature for the design of a phase tracking loop for suppressed-carrier signals in which the modulation type is unknown. We need a loop that works adequately for any phase-modulated signal, and which can improve its performance by later taking input from the modulation classifier when it starts producing an output.

The other seemingly standard modules had similar design challenges because of unknown signal attributes. Conventional implementations of frequency estimators, symbol-timing loops, and SNR estimators also assume the modulation type is known.

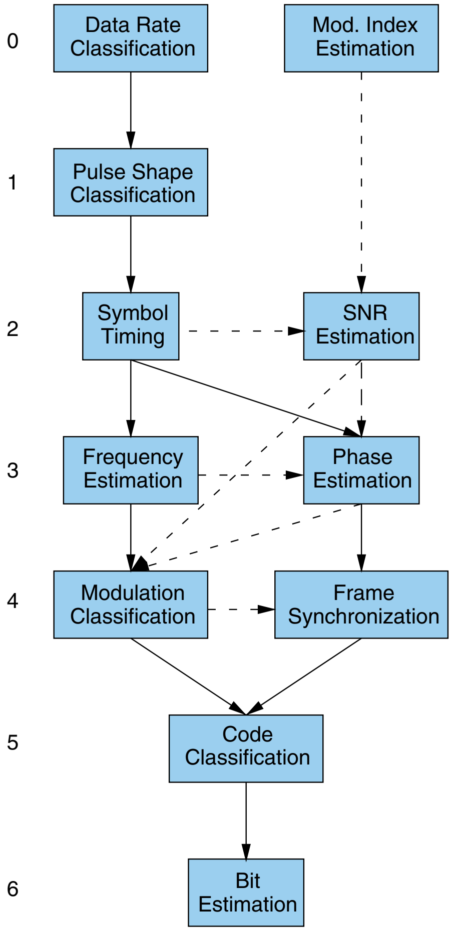


**Fig. 4. Estimator dependency graph.**

In addition, there are a number of estimators that are not conventional and only occur in an autonomous radio. These include the blocks that estimate or classify the data rate, modulation index, modulation type, and code type. We have designed each of these from scratch, in most cases by formulating the ML criterion for the estimator and attempting to solve it analytically. This led to excellent solutions for modulation classification, SNR estimation, and frequency tracking. In some other cases, the maximum-likelihood solution was not tractable, and promising ad hoc schemes were identified.

Because the design effort for the individual modules can be quite involved, we anticipate that a separate *IPN Progress Report* article may be written for each estimator and classifier. Several of these articles are already included in this issue. We briefly summarize the status of the design and analysis of some of these estimators below.

## A. Carrier Phase Tracking

In autonomous radio operation, it is desired that the receiver contain a carrier synchronization structure that is generic in the sense that it is capable of tracking the carrier phase independent of the modulation. If the modulation is restricted to the $M$-PSK class, then it is possible to construct a universal structure that performs the carrier synchronization function for all values of $M$. This structure is derived by first determining the maximum a posteriori (MAP) estimate of carrier phase based on an observation of the received signal, namely, $M$-PSK plus AWGN, and then using this to motivate a closed-loop carrier synchronization loop. Such a structure, referred to as the *MAP estimation loop*, has been previously proposed in the literature for cases where the modulation is known beforehand. The most convenient form for use in the autonomous radio application is its simplification based on low-SNR approximations applied to the nonlinearities inherent in the MAP phase estimate. When this is done, the error signal in the loop for $M$-PSK is of the form $\sin(M\phi)$, where $\phi$ is the phase error, which from simple trigonometry can be written as $\sin(M\phi) = 2\sin[(M/2)\phi]\cos[(M/2)\phi]$. Thus, it is seen that the error signal in the loop for $M$-PSK is formed from the product of the error signal $\sin[(M/2)\phi]$ and the lock detector signal $\cos[(M/2)\phi]$ in the loop for $M/2$-PSK modulation. This simple relationship forms the basis for implementing a universal structure.

### B. Modulation Classification

The autonomous radio determines the modulation type from the incoming signal. We derived new approximations to the ML classifier to discriminate between $M$-ary and $M'$-ary PSK transmitted on an AWGN channel and received noncoherently, partially coherently, or coherently, and when symbol timing is either known or unknown. A suboptimum classifier can be shown to be ten times less complex than the ML classifier and have less than 0.1-dB performance loss for symbol signal-to-noise ratios (SNRs) in the range $(-10,10)$ dB and any number of observed symbols (see Fig. 2). Other methods reduce complexity by a factor of 100 with less than 0.2 dB of performance loss. We also developed a classifier that does not require an estimate of the symbol SNR, and a new threshold optimization technique that improves the high-SNR performance of a previously published classifier. We have shown that a classification error floor exists for any $M$-PSK classifier on any memoryless channel, even a noiseless one, by deriving a lower bound on the misclassification probability as a function of the number of observed samples.

### C. Signal-to-Noise Ratio Estimation

In the design of receivers for autonomous operation, it is desirable that the estimation of SNR take place with as little known information as possible regarding other system parameters, such as carrier phase and frequency, order of the modulation, data symbol stream, data format, etc. While the ML approach to the problem will result in the highest quality estimator, it typically results in a structure that is quite complex unless the receiver is provided with some knowledge of the data symbols, typically obtained from data estimates made at the receiver (which themselves depend on knowledge of the SNR). Instead, we focused our attention on estimators that perform their function without any data symbol knowledge and, despite their ad hoc nature, maintain a high level of quality and robustness with respect to other system parameter variations. One such ad hoc SNR estimator is the so-called split-symbol moments estimator (SSME) that forms its SNR estimation statistic from the sum and difference of information extracted from the first and second halves of each received data symbol. Our initial investigations focused on demonstrating that the scheme, which was previously investigated only for BPSK modulations, is readily applicable to the class of $M$-PSK ($M \geq 2$) modulations and furthermore showed that its performance is independent of the value of $M$! Even more generally, it was pointed out that the complex symbol version of the SSME structure also could be used to provide SNR estimation for two-dimensional signal sets such as quadrature amplitude modulation (QAM). Performance results were obtained for a variety of different scenarios related to the degree of knowledge assumed for the carrier frequency uncertainty and to what extent it is compensated for in obtaining the SNR estimate.

Following this, a modification of the conventional SSME architecture was discovered that provides significant improvement in performance (as measured by the variance of the estimator). The reconfiguration consists of partitioning the symbol interval into a larger (but even) number of subdivisions than the two that characterize the conventional SSME where the optimum number of subdivisions depends on the SNR region in which the true SNR lies. It also was shown that these SNR regions can be significantly widened with very little loss in performance. Most important is the fact that, with this reconfiguration, the SNR estimator tracks the Cramer–Rao bound (with a fixed separation from it) on the variance of the estimator over the entire range of SNR values, whereas the conventional SSME deviates considerably from this bound at high SNR.

### D. Frequency Tracking

We proposed a robust frequency tracking loop for a residual carrier system that is capable of tracking the offset frequency without knowledge of received SNR. The proposed frequency tracking loop can operate robustly not only over an AWGN channel but also over a Rayleigh fading channel. This loop does not require knowledge of carrier phase. We obtained the structure of the tracking loop starting with optimum estimation of frequency criterion, namely ML estimation of frequency offset. First we obtained the likelihood of the received observations given the offset frequency. Call this the ML metric for the frequency estimation. The error signal for closed-loop tracking is obtained by taking the derivative of the ML metric. To reduce the implementation complexity with a small loss in the optimality, we simplified the derivation of the error signal.

## V. An Iterative Message-Passing Architecture

As mentioned above, the autonomous radio begins by producing estimates at the highest level in Fig. 4 and then proceeding to progressively deeper levels. Initially, no estimator at an upper level can make use of any signal attribute estimated at a level beneath it. This limitation significantly impacts performance and is inherent to any non-iterative autonomous signal parameter estimation algorithm.

A fundamental innovation of the autonomous radio envisioned here is that, after each estimator completes its first estimate in the proper boot-strap order, the deeper-level estimators send soft information to the upper estimators. A second iteration then begins, wherein each estimator makes use of this additional extrinsic information to improve its performance. After several iterations, the message-passing system will reach a reasonable convergence. We have shown that such coupled systems are typically quite robust and can provide near-ML joint estimation–decoding [13–15].

We now informally describe a nonexhaustive list of the type of soft information that can be passed upward during the estimation iterations.

### A. Messages from the Symbol-Timing Estimator

The symbol-timing module estimates the boundaries of symbol epochs and can produce a signal that indicates whether it is in lock. The lock indicator, which may be a soft value, can be fed up to the data rate classifier. For example, if the symbol-timing tracker is unable to lock onto symbol timing at one data rate, the data rate classifier can make use of that knowledge in reclassifying the data rate.

### B. Messages from the Frequency Tracker

If the frequency tracker identifies any residual frequency, this may be used to remove it from the received signal. The SNR estimator performance is affected by residual frequency, and thus it will be improved by feedback of its estimate from the frequency tracking loop.

### C. Messages from the Phase Tracker

The phase tracking loop output can be used to generate a coherent reference that can be used to improve the symbol-timing and SNR estimators, effectively improving the noncoherent performance to coherent performance.

### D. Messages from the Modulation Classification

Estimates from the modulation classifier can assist in improving SNR, data rate, and symbol-timing estimators. The likelihood functions for each modulation have expected values that obey a known relationship to the SNR and symbol timing—for example, the modulation classification becomes more certain with increasing SNR and number of symbol observations. If the observed modulation-type likelihoods are inconsistent with the estimates from the SNR and symbol-timing modules, the likelihoods can be fed back to those modules so they can revise their estimates.

### E. Messages from the Decoder

The output of the decoder includes likelihoods for each message bit. Depending on the code, it is usually simple to hard limit these likelihoods and test if the result is a codeword. Typically, codes are designed so that the undetected probability of codeword error is $10^{-10}$ or lower, which implies that if the output is a codeword, it is nearly certain to be the correct one, and no further iterations of the radio are necessary.

If the correct codeword is not obtained, then the bit likelihoods can be used to generate a soft data wipe of the received signal. This makes the signal more like a continuous wave (CW) signal, which will allow the SNR, frequency, and phase estimates to be substantially improved, which will in turn produce better inputs for the decoder to operate on in its second iteration.

## VI. A Demonstration Testbed

We are developing a software demonstration testbed of the autonomous radio. The testbed contains two parts. In the first part, the attributes of the signal may be configured, including the data rate, pulse shape, data format, modulation type, code type, and so forth. Channel effects such as SNR and Doppler also can be configured. Based on these settings, a simulated signal is generated.

The second part of the testbed implements the autonomous receiver estimators and classifiers. In most cases, these are either ML or motivated by low-complexity approximations to ML estimation or hypothesis testing. The testbed produces a graphical output that illustrates the performance of the various estimators and compares them to performance bounds, if such bounds are available.

The autonomous radio algorithms are a work in progress, as is the testbed. Thus, we have not implemented the full suite of estimators and classifiers, and the message-passing architecture is not completely in place. However, a number of pieces are fully implemented, including a residual frequency tracking loop, carrier phase tracking loop, modulation classifier, and SNR estimator.

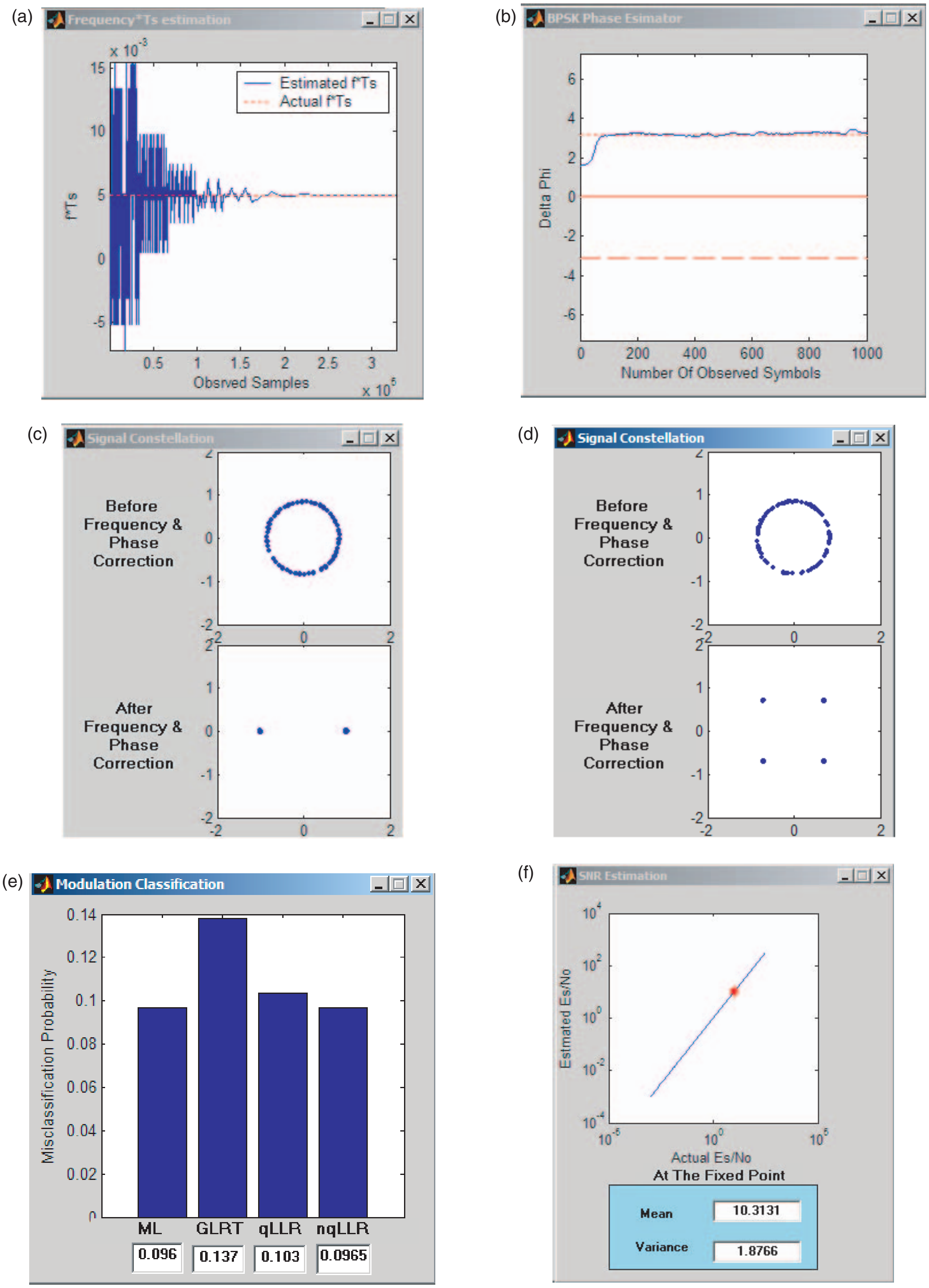
The testbed currently operates on a phase-modulated signal with a residual carrier in complex baseband. For simplicity, we assume rectangular pulse shaping, perfect symbol timing, unit power, and constant carrier phase.

The residual carrier and data-modulated signal are extracted from the received signal by utilizing a low-pass and a high-pass filter, respectively. The frequency tracking loop estimates $f_r$ from the residual carrier. Figure 5(a) shows the actual value (dotted red line) and the estimation result (solid blue line).

After frequency tracking is done and the residual frequency component is removed from the received signal, phase tracking is performed. As in a conventional implementation, the error signal in the loop for $M$-PSK is of the form $\sin(M\varphi)$, where $\varphi$ is the phase error. This results in a phase ambiguity of $\pi$ rad in BPSK and $\pi/2$ rad in QPSK. Figure 5(b) shows a sample phase estimation trajectory when tracking a BPSK signal. The blue line is the actual phase error of the loop, the solid red line corresponds to zero phase error, and the two dashed red lines correspond to the two mentioned ambiguous cases, $\varphi = -\pi$ and $\varphi = \pi$.

Thereafter, the initial raw received signal constellation and the corrected signal constellation can be compared. Figures 5(c) and 5(d) illustrate them for BPSK and QPSK modulations, respectively.

In the modulation classification block, four different methods were implemented: ML, generalized likelihood ratio test (GLRT), quasi-log likelihood ratio (qLLR), and normalized quasi-log likelihood ratio (nqLLR). The comparison of these different methods for a BPSK signal with $E_s/N_0 = -4$ dB is shown in Fig. 5(e).



| ML | GLRT | qLLR | nqLLR |
|---|---|---|---|
| 0.096 | 0.137 | 0.103 | 0.0965 |

Mean 10.3131

Variance 1.8766

**Fig. 5. Sample graphical output from the testbed: (a) frequency tracking, (b) phase tracking, (c) BPSK constellation view, (d) QPSK constellation view, (e) modulation classification performance, and (f) SNR estimator performance.**

To estimate the SNR, the split-symbol SNR estimator is used. Figure 5(f) shows a sample graphical depiction of its performance. The blue line is the reference line ($y = x$), and the red dots represent the estimator output.

## VII. Conclusions

We have presented an overview of the architecture of an autonomous radio that can receive a signal without much a priori knowledge about its defining characteristics. A number of future articles will delve into the detailed design and analysis of each estimator and classifier summarized here.

## Acknowledgments

The authors wish to thank Kenneth Andrews, Ann Devereaux, Fabrizio Pollara, Edgar Satorius, and Victor Vilnrotter for their many helpful discussions on modulation classification, Electra, and other topics related to the autonomous radio.

## References

[1] C. D. Edwards, A. Barbieri, E. Brower, P. Estabrook, R. Gibbs, R. Horttor, J. Ludwinski, R. Mase, C. McCarthy, R. Schmidt, P. Theisinger, T. Thorpe, and B. Waggoner, "A Martian Telecommunications Network: UHF Relay Support of the Mars Exploration Rovers by the Mars Global Surveyor, Mars Odyssey, and Mars Express Orbiters, IAC-04-M.5.07," *International Astronautical Congress 2004*, pp. 1–11, October 2004.

[2] J. Guinn, "Phoenix Plans for Relay Use," *Mars Network Workshop*, June 2004.

[3] R. Barry, "MSL Plans for Relay Use," *Mars Network Workshop*, June 2004.

[4] Consultative Committee for Space Data Systems, *CCSDS 101.0-B-6: Telemetry Channel Coding*, Blue Book, issue 6, October 2002.

[5] Consultative Committee for Space Data Systems, *CCSDS 211.0-B-1: Proximity-1 Space Link Protocol*, Blue Book, issue 1, October 2002.

[6] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*, New Jersey: Prentice-Hall, 1983.

[7] Consultative Committee for Space Data Systems, *CCSDS 201.0-B-3: Telecommand Part 1—Channel Service*, Blue Book, issue 3, June 2000.

[8] J. G. Proakis, *Digital Communications*, third ed., New York: McGraw Hill, Inc., 1995.

[9] G. Ungerboeck, "Channel Coding with Multilevel/Phase Signals," *IEEE Transactions on Information Theory*, vol. IT-28, pp. 55–67, January 1982.

[10] S. W. Asmar, "Characteristic Trends of Ultrastable Oscillators for Radio Science Experiments," *The Telecommunications and Data Acquisition Progress Report 42-129, January–March 1997*, Jet Propulsion Laboratory, Pasadena, California, pp. 1–5, May 15, 1997.
http://tmo.jpl.nasa.gov/tmo/progress_report/42-129/129F.pdf

[11] S. Hinedi, "A Functional Description of the Advanced Receiver," *The Telecommunications and Data Acquisition Progress Report 42-100, October–December 1989*, Jet Propulsion Laboratory, Pasadena, California, pp. 131–149, February 15, 1990.
http://tmo.jpl.nasa.gov/tmo/progress_report/42-100/100K.PDF

[12] C. D. Edwards, T. C. Jedrey, E. Schwartzbaum, A. S. Devereaux, R. DePaula, M. Dapore, and T. W. Fischer, "The Electra Proximity Link Payload for Mars Relay Telecommunications and Navigation," IAC-03-Q.3.A06, *International Astronautical Congress 2003*, September–October 2003.

[13] D. Divsalar and F. Pollara, "Turbo Trellis Coded Modulation with Iterative Decoding for Mobile Satellite Communications," *IMSC*, 1997.

[14] J. Hamkins and D. Divsalar, "Coupled Receiver-Decoders for Low Rate Turbo Codes," *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, June 2003.

[15] J. Gunn, K. Barron, and W. Ruczczyk, "A Low-Power DSP Core-Based Software Radio Architecture," *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 574–590, April 1999.