

**Precious Bits:  
Frame Synchronization in Jet Propulsion Laboratory's  
Advanced Multi-Mission Operations System (AMMOS)**

**Elizabeth (Betsy) Wilson  
Jet Propulsion Laboratory**

**ABSTRACT**

The Jet Propulsion Laboratory's (JPL) Advanced Multi-Mission Operations System (AMMOS) system processes data received from deep-space spacecraft, where error rates are high, bit rates are low, and every bit is precious. Frame synchronization and data extraction as performed by AMMOS enhance data acquisition and reliability for maximum data return and validity. Unique aspects of data phase determination, sync acquisition and sync loss and other bit-level topics are covered.

**KEY WORDS**

frame synchronization, data validity, sync marker, pseudo-noise code, packet extraction

**INTRODUCTION**

Many people think frame synchronization algorithms for CCSDS packet telemetry are standard and quite similar to each other. These common algorithms may lose data unnecessarily or create invalid or corrupted data. Because data from deep-space missions is so precious, AMMOS attempts to recover as much data as possible, while delivering only good quality data. This paper describes some aspects of the frame synchronization and packet extraction algorithms used by AMMOS.

## FRAME SYNCHRONIZATION ALGORITHMS AND DATA VALIDITY

Many frame synchronization algorithms give priority to speed, not data return, primarily because they must process high-rate data. Usually chip-based, these algorithms do not buffer data, but stream it through the chip. They do not have the ability to look upstream, downstream, or inside the data. Without these capabilities, synchronized data may leave the user to either resolve errors or process incorrect data. For instance, if data is identified as being in "verify" or "flywheel" mode, the users must either:

- 1) determine for themselves whether the data is valid enough to be processed,
- 2) not do the validity determination and process potentially invalid data, or
- 3) call all questionable data invalid, thus losing potentially valid data.

To determine if data in "verify" or "flywheel" mode is valid, a user must look upstream to determine the final resolution of the uncertain state. The user must therefore retain data in these modes until the validity question has been resolved. AMMOS frame synchronization only creates a frame if it is valid enough to be passed on to packet extraction, and does not create frames with questionable validity. Groups of bits not used for packet extraction are archived, as are all other bits. They are identified and routed such that packets are not extracted from them, but are still available for later human analysis. Succeeding paragraphs give the details of these validity checks.

### DETAILS OF AMMOS FRAME SYNCHRONIZATION

How a synchronizer handles failed sync acquisition attempts can affect the number of frames produced. Frame synchronizers typically first search for the sync marker (also called pseudo-noise or PN code). Once the first sync marker is found within a specified bit error tolerance, the algorithm looks for the next marker the appropriate number of bits upstream. If the next one is not found in the expected location, less thorough synchronizers bypass intervening bits. They generally examine upstream bits for another sync marker only after the point in the data stream where the expected second sync marker was supposed to be found, but wasn't. This results in one frame's worth of data being discarded. AMMOS' algorithm begins searching for the next sync marker beginning with the second bit of the first found sync marker. Sometimes sync starts within this otherwise discarded data. This technique is especially helpful if embedded sync markers may normally be within a frame, such as having recorded or relayed frame-level data within packets. Similarly, when sync is lost, sync marker searching begins anew with the second bit after the last good frame produced, as opposed to the first bit after the first failed frame.

Once two sync markers are found the correct number of bits apart, further checks ensure the frame is good enough for downstream processing. Some algorithms decide either that sync is found and packet extraction may begin, or sync is ready to be verified by a specified number of correct sync markers upstream (sometimes called the verification number). AMMOS makes Reed-Solomon (RS) decoding an integral part of the sync algorithm. If sync is either imminent or being maintained, then RS decoding is performed. If RS decoding fails, then AMMOS considers sync to be respectively either not yet found, or lost, thus avoiding further processing of frames with bit errors. If the frames are not RS encoded, then the CCSDS

mandated 16-bit Frame Error Control Field is checked, with similar results for this check's failure.

If RS decoding is successful, the frame still has more hurdles to pass. Subsequent packet extraction processing should not have to verify a frame's worthiness. The methods defined above also give a better measure of usable data return because frames not passing the decode or check process do not produce packets and are thus not useful for most data accounting purposes.

## **BEYOND REED-SOLOMON DECODING**

Because sync markers are not usually RS encoded, they are not the best set of bits to use to determine sync. In missions where the sync marker itself is not RS encoded (alas, all too often), once sync is found successive sync markers are no longer considered a valid sync maintenance method. The marker itself could be corrupted beyond the bit error tolerance specification, yet the frame may easily be corrected by RS decoding. For this reason, once sync has been acquired, AMMOS no longer uses the sync marker to maintain synchronization, but uses only RS decoding results. That is, once sync has been acquired, each potential frame is RS decoded, and if it passes, it is considered a frame (pending further checks, see below), regardless of sync marker errors.

RS decoding is transparent, meaning it works for both true and complemented data phases; therefore complemented data exits the RS decoding process still complemented. (After RS decoding a frame may be of either phase.) Phase changes will appear to the RS decoder as bit errors and may be corrected if close enough to the frame beginning or end to enable correction. Thus, examination of the phase of the sync marker, if the sync marker is not RS encoded, is not sufficient to determine the phase of the frame; some other, constant-value field is needed. Therefore, AMMOS examines the transfer frame primary header field "Version ID" to see if it meets the mission's expected value(s). If it is complemented, then the entire frame is considered complemented, the phase is reversed, and frame processing continues with true phase data. If the Version ID is not exactly correct for either phase, the frame is considered corrupted and does not go on for further processing. As with all invalid data, the bits are archived, but not passed on for packet extraction.

Once a frame has passed the previous hurdles, AMMOS has further checks in store, because experience has shown that occasionally something can still be wrong. For instance, a "frame" shifted by one byte may still pass the AMMOS sync maintenance checks, may be properly RS decoded, and may accidentally pass the Version ID check. But the data, of course, is still incorrect. (This is possible because, as explained above, we do not examine the sync marker once sync is acquired.) To avoid this error, AMMOS compares the value of the 10-bit Spacecraft ID field in the CCSDS transfer frame primary header against one or more possible values. If it matches no specified value, the frame is considered invalid, and no further processing is performed on it. If this happens, sync acquisition is begun anew, starting with the second bit of the sync marker of the failed frame.

## **INSIDE THE FRAME**

Sometimes spacecraft flight software makes data that appears valid at the frame level, but is not valid enough for packet extraction. For instance, zero- or garbage-filled frames may not have the proper CCSDS identification of their idle, or filler, status. (Yes, sad but true, flight software does not always follow the rules, but even if the flight software violates the CCSDS standards, it is AMMOS' job to make sense of the data.). Usually, the Spacecraft ID or Version ID check will find these frames. If the frame is garbage, but the frame primary header passes all previous tests, then packet extraction will be attempted. Before packet extraction, the First Header Pointer (FHP) field is examined.

The FHP is a crucial field for packet extraction. Because a mission may have many valid Application Process Identifier (APID) values, many bit strings could mistakenly be used as a valid APID, with resulting data chaos. AMMOS checks the FHP, and if the FHP does not match the value expected, then no frame bytes prior to the FHP-designated byte are made into a packet. The bytes that might have been spliced onto a frame-spanning packet are discarded (archived and marked as invalid), and packet extraction starts again where designated by the FHP. Because flight software may fill frames with zeroes, AMMOS strongly recommends against using a value of zero (0) as a valid Application Process ID (APID) for any packet type.

## **CONVOLUTIONAL DECODING**

All deep-space data is convolutionally encoded prior to transmission. The convolutional decode process is always performed prior to frame synchronization. This decode process may produce bit slips, i.e., a bit stream may contain several bits more or fewer than the spacecraft actually transmitted. While maintaining convolutional decode output through bit slips may aid in antenna lock accountability, it wreaks havoc on data. After all, any size bit slip corrupts all upstream (following) data, and because bit slip locations cannot be known, they cannot be corrected. The AMMOS frame sync algorithm does not allow bit slips. The sync markers must be positioned at the exact location to be considered for further processing. If a sync marker is not separated by the correct number of bits from the previous marker, the bits in between the markers are not considered a valid frame. Bit slips of any size cause failed RS decode and failed cyclic redundancy checks, thus causing sync to be lost when they occur in sync maintenance mode.

## **DATA OUTPUT AND IDENTIFICATION**

Once frames are considered valid for further processing, additional care is taken to extract as much data as possible. Scientists looking at data from a planetary encounter, for instance, are eager to receive every possible data bit, because data from that moment in time and space will never be duplicated.

All input bits not put into a valid frame are output and flagged with a code representing the problem encountered. All data input to the frame synchronization process is also output as "raw" data, so it may be reprocessed with different parameters if need be. AMMOS usually receives data from the

Deep Space Stations that has been initially frame synced by a chip-based algorithm. This initial synchronization allows the stations visibility into data quality, although some missions take this data as is, and do not take advantage of AMMOS' further refinements. The usual agreement between AMMOS and the Deep Space Stations is for AMMOS to receive all bits, in order, despite the station's frame synchronization status. AMMOS then treats this data as a pure bit stream, without regard to the marked sync status. AMMOS is saved some bit shifting if the two sync alignments are the same, the usual case. Where AMMOS pulls out more good data or removes bad data is generally at a streams "edges," where frames are marked as being in a grey area, such as search, verification or flywheel modes.

## **PACKET EXTRACTION**

The effort to use only frames containing all good bits pays off during packet extraction. If a frame contains incorrect bits, then incorrect or invalid packets may be extracted. An incorrect packet is one whose Application Process Identifier (APID) and other CCSDS packet header fields appear valid and thus the packet appears valid, but has corrupted data. Invalid packets are those with an unrecognized APID or other packet header errors. Incorrect packets create problems when they are processed. Invalid packets are not used. In extreme circumstances, incorrect packets may corrupt storage databases or take a valid packet's place. AMMOS also creates invalid packets from bits extracted from a valid frame but unable to be put into a valid packet. For example, a first synced frame's beginning may contain the end of a packet spanning two frames. Those bits cannot be applied to a packet because the packet header is missing; they are thus put into an "invalid packet" and flagged with the reason for their invalidity. They are available for examination, but no other processing.

## **TURBO-CODES**

In the coming "turbo-code" era, frame sync is done in the symbol domain, as opposed to the bit domain. The process produces frames, but any one frame's validity is a statistical probability. Therefore, a Cyclic Redundancy Check (CRC, also called "Frame Error Control Field" by CCSDS) is important to verify that the frame is actually the same as when transmitted from the spacecraft. Turbo-coded frames not passing the CRC check are not considered valid frames by AMMOS. (JPL is not currently supporting any turbo-coded telemetry missions.)

## **FLYWHEELING, ETC.**

For Time Division Multiplexed (TDM) telemetry, or for packetized frames either not RS decoded or not CRC checked, different methods of sync acquisition and loss are used. Sync is acquired without any verification period. If the first frame meets the criteria, it is in sync. These criteria include proper positioning of two sync markers of the same phase and other checks of frame fields. For TDM, at a minimum the format ID, an identification of frame type, must be valid and matching for both frames. During periods of bad data, generally at a track's start or end, sync may be acquired and lost several times in rapid succession. Having no verification period allows retention of data

that is valid, but not succeeded by valid data. During potential sync loss, when the sync marker is outside of the bit error tolerance, flywheeling is done, but the flywheeled frames are held onto until it is determined that either sync will be lost at the end and all flywheeled frames are then declared as out of sync, or that sync can be maintained and all flywheeled frames are output as valid, although flagged as flywheeled. By holding onto frames until the flywheel period is over, the validity of synced frames is ensured, and users do not need to apply further data validity checks. Holding onto frames causes a small slow-down and succeeding burst of data, but the flywheel factor maximum is kept to 4, and the burst does not cause downstream problems.

## **CONCLUSION**

In applications where both data quality and quantity are important, new frame synchronization and packet extraction implementations should attempt to incorporate some of the algorithms discussed above. Although the errors and problems discussed may be rare, it is not good practice for software to allow bad data to be processed.

## **NOTES**

AMMOS personnel usually refer to data as a singular noun. Although in Latin it is the plural of datum, we treat it as a collective, like "sugar". This allows us to construct our sentences more clearly and less awkwardly. When we do this, we can refer to data as flowing through pipelines and processors.

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