Earl Bellinger

JPL SIP Research Fellow

Cassini Science Planning and Sequence Team

Systems and Software Division (J)

Planning and Execution Systems Section (317)

NASA Jet Propulsion Laboratory

Mentors: Diane Conner and David Mittman

August 6, 2012

CASSIUS: The Cassini Uplink Scheduler

The Cassini Uplink Scheduler (CASSIUS) is cross-platform software used to generate a radiation sequence plan for commands being sent to the Cassini spacecraft. Because signals must travel through varying amounts of Earth's atmosphere, several different modes of constant telemetry rates have been devised. These modes guarantee that the spacecraft and the Deep Space Network agree with respect to the data transmission rate. However, the memory readout of a command will be lost if it occurs on a telemetry mode boundary. Given a list of spacecraft message files as well as the available telemetry modes, CASSIUS can find an uplink sequence that ensures safe transmission of each file. In addition, it can predict when the two on-board solid state recorders will swap. CASSIUS prevents data corruption by making sure that commands are not planned for memory readout during telemetry rate changes or a solid state recorder swap.

Table of Contents

1 BACKGROUND 3

2 SCHEDULING UPLINKS 4

2.1 TELEMETRY MODES 4

2.2 PREDICTING SOLID STATE RECORDER SWAPS 5

2.2.1 Data Frames Remaining 5

2.2.2 Effective Telemetry Rate 6

2.2.3 Swap Times 7

3 ALGORITHMIC SCHEDULE GENERATION 8

3.1 CIUT 8

3.2 CASSIUS 8

3.3 A RANDOM WALK 9

3.4 IMPROVING SWAP PREDICTIONS 10

4 INTEGRATION 11

5 ACKNOWLEDGEMENTS 11

6 REFERENCES 12

1 Background

The Cassini robotic spacecraft began orbiting Saturn on July 1, 2004 after traveling over three billion kilometers for about seven years in its journey from Earth. In order to get to its destination, Cassini had to swing by Venus twice, Earth, and Jupiter (VVEJ) for gravitational assistance. Equipped with instruments for studying dust, plasma, electromagnetic waves, and other phenomena, the orbiter quickly proved to be indispensable in understanding Saturn, its rings, and its satellites (Matson et al. 2002).

Roughly each month, the Cassini Science Planning and Sequence Team (SPST) organizes a list of goals and tasks based on what will occur during orbit. In order to properly execute these goals, the team must prepare a list of commands to be radiated to the spacecraft. These messages contain instructions that tell Cassini to reposition, perform maneuvers, calibrate instruments, collect data, etc.

Each day, Cassini goes through an observation phase and a transmission phase. During this latter phase, the spacecraft downlinks its observations back to Earth, and every so often, the Deep Space Network (DSN) uplinks instructions for the upcoming sequence. Since Saturn is over a billion kilometers from Earth, a message takes more than an hour to go to or from the spacecraft.

Cassini is now a year into its second extended mission ("Solstice") and has roughly five years until the remaining fuel is exhausted (Roatsch et al. 2012). The mission is planned to end in September of 2017 with a Saturn impact (Yam et al. 2008). Until the mission ends, hundreds of message files will need to be radiated to the spacecraft.

Scheduling the transmission of message files to Cassini is highly nontrivial. The inherent complication in passing messages to the spacecraft is that the data transmission rate frequently changes, and messages cannot be read out from memory when a change transpires or else the readout will be terminated. Furthermore, care must be taken when radiating files that will pull flight software loads off of Cassini's on-board solid state recorders if the recorders are expected to reach maximal capacity during transmission. The following sections detail how uplink schedules for spacecraft message files are algorithmically generated and verified to conform to these constraints.

2 Scheduling Uplinks

2.1 Telemetry Modes

Due to Earth's rotation, signals sent to the spacecraft travel through a variable amount of atmosphere. The maximum transmission bandwidth is a function of time and has a shape that resembles a normal distribution curve. The DSN and Cassini need to be in agreement with respect to the rate at which data is being transmitted, so this curve must be divided into many different periods of constant telemetry rates. This is accomplished in a manner that is reminiscent of the rectangle method from integral calculus, with the width of the rectangle being the duration of the telemetry period and the height of the rectangle being the amount of data. After being radiated, the message files that contain memory readouts must only be read out from memory within a single 'rectangle'; that is to say, between the start and end of a particular telemetry period. In other words, for every spacecraft message file S, the memory readout beginning time T_B

and ending time T_E must come either both before or both after the starting time t of every telemetry mode M:

$$\forall i \forall j \left(\left[S_i(T_B) < S_i(T_E) < M_j(t) \right] \oplus \left[M_j(t) < S_i(T_B) < S_i(T_E) \right] \right) \quad [1]$$

For the sake of caution, each telemetry mode change is padded on both sides by three minutes due to the uncertainty of the actual telemetry rate change, which can vary by up to 68 seconds.

2.2 Predicting Solid State Recorder Swaps

Cassini has two on-board solid state recorders that are capable of recording and playing back data. If one recorder reaches its maximal capacity, the spacecraft will swap to the other. It is important that the recorders do not swap during the execution of files that write to or pull flight software from the recorders, or else the data might become corrupted. The range of time for possible swaps between Cassini's solid state recorders is a function of how many data frames D remain on the active recorder and the effective telemetry rate R at which data can be transmitted.

2.2.1 Data Frames Remaining

Cassini reports the position of the playback pointer P and the record pointer R of its primary solid state recorder. If the record pointer is greater than the playback pointer, then the amount of data frames remaining is proportional to their difference:

$$D = \frac{P-R}{F} \leftrightarrow P > R \qquad [2]$$

where F is the frame conversion factor with a constant value of 550. Otherwise, the amount of data frames remaining before the swap is proportional to the difference subtracted from the total amount of available space S:

$$D = S - \frac{R-P}{F} \leftrightarrow P \leq R \qquad [3]$$

In order to compute the amount of time it will take to transmit the remaining data frames, it is also necessary to calculate the effective telemetry rate at which they radiate.

2.2.2 Effective Telemetry Rate

The effective telemetry rate R, expressed in bits per second, comes from summing the product of the duration d and rate r of every telemetry mode available after a given snapshot time T_0 and dividing by the total time of use:

$$R = \frac{\sum M_j(d) M_j(r)}{\sum M_j(d)} \ \forall j: M_j(t) > T_0 \qquad [4]$$

With the effective telemetry rate and the amount of remaining data frames, it is now possible to compute the range of swap times during which files cannot be radiated.

2.2.3 Swap Times

The earliest possible swap time S_0, expressed as the number of seconds after the snapshot time, is proportional to the ratio of remaining data frames to the effective telemetry rate:

$$S_0 = B\frac{D}{R} \qquad [5]$$

where B is the number of bits per frame with a constant value of 8,800. Likewise, the latest possible swap time S_1 is some number of seconds after the earliest possible swap time, and is proportional to the ratio of the amount of Magnetosphere and Plasma Science (MAPS) and Engineering data in bits being collected while in playback mode, denoted m, to the effective telemetry rate:

$$S_1 = S_0 + \frac{m}{R} \qquad [6]$$

Together, these times estimate the entire range in which a solid state recorder swap can occur. Recently, a spacecraft message file named Primo Repair has been developed that will correct errors resulting from files that write data to the recorder when a swap occurs. However, Primo Repair cannot help in the case where a file pulls flight software loads from the solid state recorders during a swap. It must be ensured that the execution beginning time E_0 and ending time E_1 for these files occur either both before or both after a predicted solid state recorder swap:

$$E_0 < E_1 < S_0 < S_1 \oplus S_0 < S_1 < E_0 < E_1 \quad [7]$$

It is important that these files execute safely, so it is best to be cautious when finding a time to transmit this file.

3 Algorithmic Schedule Generation

3.1 CIUT

Until the present, radiation times for spacecraft message files were determined manually with aid from a spreadsheet called the **C**assini **I**nstrument Expanded Block **U**plink **T**ool, or CIUT. With this tool, a user specifies telemetry modes and rearranges message files until an uplink queue conforming to [1] is found. The user can then press a button to generate an extensible markup language (XML) document to give to autoRAD, which automatically radiates the files to the spacecraft at the specified times. Unfortunately, CIUT requires the use of Visual Basic Macros for generating XML. These macros are only available in Microsoft Excel 2004, which is no longer supported by Mac OS X. Hence CIUT is both laborious and platform dependent.

3.2 CASSIUS

Replacing CIUT is CASSIUS, the **Cassi**ni **U**plink **S**cheduler. The principal objectives for CASSIUS are to duplicate the functionality of CIUT into a platform independent language and to automate its use entirely. With an automated routine, the likelihood of errors introduced due to human use decreases dramatically, and productivity

increases because users no longer have to dedicate their time to accomplish this task. Java was the language chosen for this project because it is widely popular, easily maintained, and platform independent. CASSIUS can automatically generate uplink schedules and has improved on CIUT's swap prediction algorithm.

3.3 A Random Walk

After a list of message files has been created and is ready to be transmitted to the spacecraft, it is necessary to rearrange and delay the uplink time for each file in the sequence accordingly to ensure that the files may be radiated without risk of being lost. In other words, the list needs to be adjusted until it conforms to [1]. Rearranging a list until it meets certain criteria is a classically intractable problem in computing. Indeed, this problem has factorial complexity, meaning that the solution space explodes with even a very small input size. For instance, if the list contains only ten items, there are 10! = 3,628,800 possible permutations. If one tried to generate all of the possible permutations of a list containing 150 items at a rate of one reordering per second, our neighbor galaxy Andromeda would collide with the Milky Way before completion!

It is possible that messages might have to be divided up into multiple lists of varying lengths, such as in the case where the total uplink time of all of the messages exceeds the total available transmission time. The computational complexity of finding all permutations of all lengths for a list of size *n* is

$$O\left(\sum_{k=0}^{n} \frac{n!}{k!}\right) \qquad [8]$$

This complicates matters even further than the ordinary factorial case, so brute forcing the solution is clearly not an option here. Instead, a Markov chain Monte Carlo method was implemented: rather than trying every possible permutation of spacecraft message files, a fixed number of random permutations are generated. This effectively takes a random walk through the solution space and converges on the best sequence stumbled upon in a short, fixed amount of time without biasing the solutions. Experimenting with this functionality has successfully unveiled optimal solutions that went undiscovered by human operators.

3.4 Improving Swap Predictions

Data travelling from the spacecraft to the DSN comes in formatted units called packets that contain not only the payload itself but also header information for the purpose of identification. In CIUT, the calculation of the effective telemetry rate is based on the rate at which the data alone can be transmitted, so each telemetry rate *r* needs adjustment in order to account for packet overhead. Each rate was mapped to its corresponding adjusted rate *a*, and then interpolated with an exponential power series. That function was then used to transform each telemetry rate *r* into a modified rate *r'*:

$$r_i' = 0.46965 * r_i^{1.053} \qquad [9]$$

This predictor has a residual sum of square errors (RSS) value of:

$$RSS = \sum (a_i - r_i')^2 \cong 4.5962 * 10^7 \qquad [10]$$

A large RSS value such as this indicates a very poor fit. CASSIUS improves the swap time calculations by simply using the adjusted rates directly, which reduces the RSS by over forty million and gives a perfect value of exactly 0. This is an improvement of seven orders of magnitude!

CASSIUS has also corrected a previously undetected error in the CIUT's calculations of the latest possible swap time. Unlike the estimations of the earliest possible swap time, these calculations failed to account for packet overhead and did not adjust the telemetry rates. CASSIUS has improved this calculation on the order of minutes to hours, and has successfully predicted actual swaps that CIUT failed to catch.

4 Integration

CASSIUS is software for generating and verifying uplink schedules for safely transmitting messages to Cassini in its mission to Saturn. CASSIUS was delivered on Monday, August 6 of 2012. CASSIUS is replacing the existing tool, CIUT, which could only verify schedules and not generate them. SPST plans to use CASSIUS to generate radiation sequence plans and predict solid state recorder swaps for every forthcoming sequence until the 2017 end of mission and Saturn impact.

5 Acknowledgements

The author thanks Diane Conner, David Mittman, and Kari Magee for initiating and managing this project. The author thanks SPST for dedicating their time to testing and providing useful suggestions. Most of all, the author thanks SPST Manager Bill

Heventhal, whose guidance, knowledge, and careful eyes made this project come to fruition.

6 References

Roatsch, Th., Kersten, E., Hoffmeister, A., Wählisch, M., Matz, K.D., Porco, C.C., 2012, *Recent improvements of the Saturnian satellites atlases: Mimas, Enceladus, and Dione*, Planetary and Space Science, DOI: 10.1016/j.pss.2012.02.016.

Matson, D. L., Spilker, L. J., and Lebreton, J. P., 2002, *The Cassini/Huygens Mission to the Saturnian System*, Space Science Reviews, 104, 1-58.

Yam, C.H. Davis, D.C. Longuski, J.M. Howell, K.C., 2008, *Saturn Impact Trajectories for Cassini End-of-Life*, American Astronautical Society, Advances in the Astronautical Sciences, 129, 93-116.

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, and was sponsored by [name of program] and the National Aeronautics and Space Administration.