

SpaceWire ネットワーク内時間の同期精度向上手法

中島 啓太^{1,a)} 高田 広章¹

概要: 宇宙機内コンポーネントの通信ネットワークとして SpaceWire が普及している。SpaceWire では基準となるマスターコンポーネントが自身の時刻情報をそれ以外のスレーブコンポーネントに送信し、スレーブコンポーネントがその時刻に同期することでコンポーネント間の時間同期を実現している。しかし、時刻情報がある確率分布に従って遅延するため同期精度が低下する。物理現象を観測する宇宙機ミッションでは同期精度が低下すると物理現象を正確に把握できないことがある。そこで本論文では、時刻情報の遅延が従う確率分布の特徴を利用した時間の同期精度向上手法を提案する。提案手法ではスレーブコンポーネントのみに独自のハードウェアユニットを追加する。独自のユニットを模倣するシミュレータを開発し、そのシミュレータを使って遅延緩和実験を行った。実験の結果、ASTRO-H で用いられている手法に比べて、提案手法を用いることで、時刻のズレを約 15% に緩和できることを確認した。

キーワード: SpaceWire, Time-Code

Time Difference Alleviation Method in SpaceWire Network

NAKAJIMA KEITA^{1,a)} TAKADA HIROAKI¹

Abstract: SpaceWire is the most popular standardized network connecting components of a spacecraft. In SpaceWire, one of the components called master sends its Time-Code to the others called slaves. The slaves synchronize its internal time with the received Time-Code. This time synchronization way generates delay in the arrival of the information. The delay causes time difference between the components. In some space missions, the time difference need to be mitigated for accurate observation of data from the universe. We propose a new method to alleviate the time difference. In the method, one additional hardware unit is needed only in a slave node, and uses features of the probability distributions deciding the amount of the delay. To confirm effectiveness of the method, we developed a simulator of the unit. The results of the simulations show that the method can alleviate the time difference to about 15% of the ASTRO-H's method.

Keywords: SpaceWire, Time-Code

1. はじめに

宇宙機はネットワークに接続された複数のコンポーネントとルータで構成される。以前は独自のネットワークを用いていたが、近年は標準化されたネットワークを用いることが多い。標準化によってインタフェースを共通化するとコンポーネントを再利用しやすくなる。コンポーネントを再利用できれば、開発期間短縮、コスト削減、信頼性向

上のメリットが得られる。そのため、標準化されたネットワークが積極的に用いられている。標準化された宇宙機向けネットワークとして SpaceWire[1] が普及している。

SpaceWire は仕様でコンポーネント間の時間の同期方法を定めている。基準となるマスターコンポーネント（マスター）が自身の時刻をそれ以外のスレーブコンポーネント（スレーブ）に送信し、スレーブがその時刻に同期することで時間同期を実現している。この同期方法には時刻情報が遅延する問題がある [2]。時刻情報が遅延することでコンポーネント間の時間の同期精度が低下する。時刻情報の遅延（時刻遅延）は SpaceWire の構成で決まる固定長の固

¹ 名古屋大学大学院情報科学研究科
Graduate School of Information Science, Nagoya University
^{a)} nakajima@ertl.jp

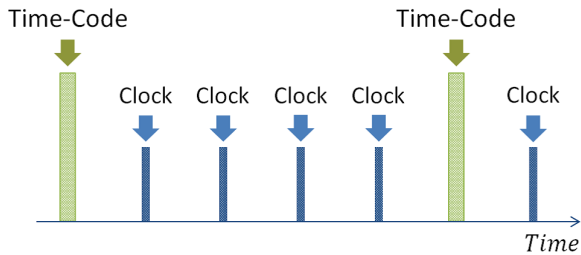


図 1 SpaceWire ネットワーク内の時間のイメージ

定遅延と、ある確率分布に従って変動する変動遅延で構成される [3]. 固定遅延は実機の測定から把握できる. よって、スレーブが固定遅延を見込んで届いた時刻に同期すれば、固定遅延は同期精度に影響しない. 時間の同期精度を低下させるのは変動遅延であると言える. また、スレーブが届いた時刻に同期しても、スレーブ・クロック源のドリフトによる誤差（ドリフト誤差）が時間の同期精度を低下させる. つまり、変動遅延とドリフト誤差がコンポーネント間の時間の同期精度を低下させる要因である. 物理現象を観測する宇宙機ミッションでは 100nsec の同期精度が必要になることがある [4]. 先行研究で変動遅延に対処する手法 [5] が提案されているが、マスターと経由するすべてのルータに独自のハードウェアを追加する必要がある. また、変動遅延の確率分布を利用した手法は知られていない.

本論文では、変動遅延の確率分布の特徴を利用した時刻遅延の緩和手法、すなわち、時間の同期精度向上手法を提案する. 提案手法ではスレーブのみに独自のハードウェアユニットを追加する. ユニット内で時刻情報の到着タイミングを観測し、変動遅延の平均が持つ偏りから変動遅延を推定する. 提案手法を用いることで同期精度の低下を抑制できる. 本論文ではマスターとスレーブの時間同期のみを考える. マスターと宇宙機外部の時間同期は別研究である.

本論文の構成を示す. 2 章で SpaceWire ネットワーク内時間について述べ、3 章で時刻遅延を緩和する DEDA (Drift Estimation and Delay Alleviation) ユニットについて述べる. 4 章で DEDA ユニットを模倣するシミュレータを用いた実験について述べ、5 章でまとめを述べる.

2. SpaceWire ネットワーク内時間

本章ではまず、SpaceWire ネットワーク内の時間のイメージを示す. その後、コンポーネント間の時間の同期精度を低下させる要因について述べ、既存の同期精度向上手法について簡潔に述べる.

2.1 時間のイメージ

SpaceWire ネットワーク内の時間のイメージを図 1 に示す. 大きな時刻をマスターの Time-Code と呼ばれる時刻情報で刻み、小さな時刻をクロック源のクロックで刻む. 普段利用する時計で例えると、Time-Code が長針に、ク

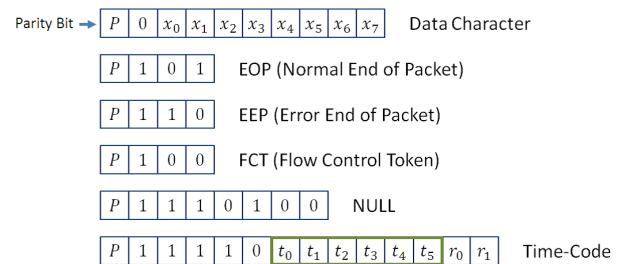


図 2 SpaceWire における通信の最小単位

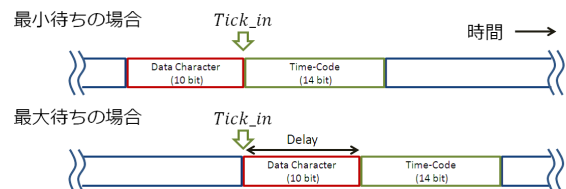


図 3 Time-Code 送信依頼時に発生する待ち

ロック源のクロックが秒針にあたる. 短針にあたる時刻は OS 等のソフトウェアで管理されることが多い. 主に時刻遅延 (Time-Code の遅延) とスレーブ・クロック源のドリフト誤差によってコンポーネント間の時間がズれる.

2.2 時刻遅延

時刻遅延は固定遅延と変動遅延で構成される. 固定遅延は Time-Code を送信すると毎回発生するほぼ固定長の遅延である. 固定遅延はマスター、ルータ、スレーブの内部処理や信号の伝搬によって生じる.

図 2 に SpaceWire における通信の最小単位を示す. データは Data Character の $x_0 \sim x_7$ に格納される. 複数の Data Character をまとめてパケットを構成することでデータが送信される. パケットの最後尾には EOP、もしくは、EEP が付加される. FCT は Data Character, EOP, EEP を合計 8 つ受け取れる時、つまり、8 つ分のバッファ領域を確保するたびに通信相手に送信される. NULL は何も通信をしていない時にリンクを維持するために送信される. 時刻情報は Time-Code の $t_0 \sim t_5$ に格納してマスターからスレーブに送信される. Time-Code の $r_0 \sim r_1$ は次期仕様で割込みに利用予定の予約ビットである [6]. SpaceWire の通信は必ず最小単位で行う.

変動遅延はこの仕様上の決まりが原因で発生する. Time-Code の送信依頼をしても、依頼時に送信している通信の最小単位を送りきるまで Time-Code の送信は待たされる. 図 3 に最小待ち、最大待ちを示す. ちょうど最小単位を送り終えたタイミングで Time-Code の送信依頼 Tick.in が入力されると、何も待たされずに Time-Code が送信される. 逆にちょうど最小単位を送り始めたタイミングで Tick.in が入力されると、Time-Code の送信を最大で Data Character の 10bit 分待たされる. 変動遅延はマスターと各ルータで発生する Time-Code 送信待ちによって生じる.

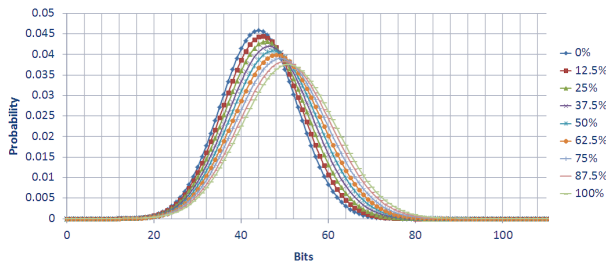


図 4 変動遅延の確率分布 (ルータ段数 10)

先行研究 [3] からこの待ちは、Time-Code 送信時のトラフィックに応じてランダムに発生することが分かっている。図 4 に、ルータ段数 10 の場合に発生する変動遅延の確率分布を示す。横軸は待ちの合計ビット数を表し、縦軸は発生確率を表す。0%のグラフはデータ送信率 0%を意味し、Time-Code 送信依頼時のトラフィックが NULL のみである場合を表す。100%のグラフはデータ送信率 100%を意味し、Time-Code 送信依頼時のトラフィックが Data Character, EOP, FCT のみである場合を表す。Data Character:EOP:FCT=88:1:11 の割合でパケットが送信されると想定している。0%~100%の間のグラフは、Time-Code 送信依頼時のトラフィックが NULL とデータ送信率に対応するパケットである場合を表す。トラフィックが変化しても最頻値があまり変化しないことが見て取れる。

2.3 ドリフト誤差

スレーブ・クロック源のドリフト誤差もコンポーネント間の時間の同期精度を低下させる。主に温度変化、電圧変化、経年劣化によってクロック源のクロックの刻み幅 (周波数) が変動する。宇宙機では特に温度変化による刻み幅の変動が大きい。本論文では時間 T (時刻 $t_1 \sim t_2$) に対するドリフト誤差 D_T を以下の式で定義する。

$$D_T = T \times \int_{Temp_{t_1}}^{Temp_{t_2}} FA(x) dx \quad (2.1)$$

$$FA(x) = FA_0 + (1 + FA_0) \cdot FS(x) \quad (2.2)$$

$Temp_{t_1}$ は時刻 t_1 における温度を、 $Temp_{t_2}$ は時刻 t_2 における温度を表す。 $FA(x)$ は温度 x によって変化する周波数偏差関数を表す。周波数偏差関数は温度 x で発生する理想の周波数からのズレを表す。 FA_0 は常温 (25°C) 時の理想の周波数からのズレを表す常温偏差である。また、 $FS(x)$ は温度 x で発生する常温時の周波数からのズレを表す周波数安定度関数である。常温時の周波数には常温偏差 FA_0 も含まれるため $(1 + FA_0)$ に掛けている。

2.4 既存の同期精度向上手法

Time-Code 送信時にマスターと各ルータでは 0~10bit の待ちがランダムに発生する。このビット数をハードウェアを追加することで取得し、毎回 10bit 分の遅延になるように故意的に遅延を追加する手法が提案されている [5]。変

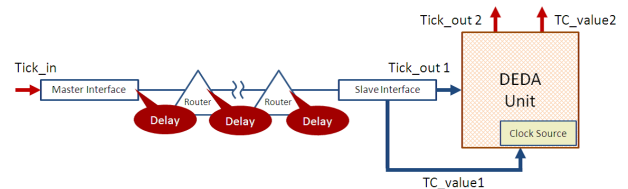


図 5 DEDA ユニットのコンセプト

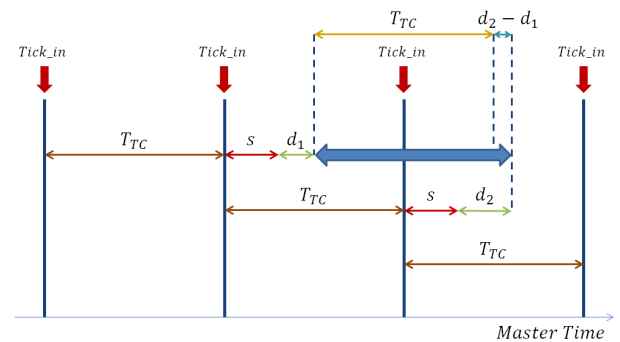


図 6 DEDA ユニットで観測できる差分

動遅延を固定遅延化する手法と言える。ただし、マスターと各ルータにハードウェアを追加する必要がある。

また X 線天文衛星 ASTRO-H では、実機の測定から時刻遅延の平均 (固定遅延+変動遅延の平均) を把握して、その平均分だけ時刻を補正する手法が用いられている [7]。

3. DEDA ユニット

本章では、DEDA ユニットの動作を説明する。

図 5 に DEDA ユニットのコンセプトを示す。図 5 の Interface は SpaceWire インタフェースを表す。マスターが Time-Code 送信依頼 Tick_in を入力しても、スレーブの Time-Code 到着信号 Tick_out1 と時刻情報 TC_value1 は遅れて出力される。基本的に Time-Code の送信周期は一定であり、Time-Code の時刻情報は 1 ずつ増加する。DEDA ユニットでは変動遅延を推定することで次の Time-Code の送信依頼タイミングを予想し、時刻遅延の影響を抑えた Tick_out2 と TC_value2 を出力する。

DEDA ユニットで行う処理を以下に列挙する。

- Time-Code 到着時刻の差分を観測
 - 観測した差分からドリフト誤差 $2D_{TC}$ を推定
 - 観測した差分から変動遅延を推定
 - Tick_out2 と TC_value2 の出力タイミングを調整
- 本章の以後の節でこれらの処理を順に説明する。

3.1 観測差分

DEDA ユニットではユニット内の Clock Source の刻みを使って、Time-Code 到着信号 Tick_out1 の到着時刻の差分を観測できる。観測できる差分 (観測差分) を図 6 に示す。図 6 の値はすべてマスター時間で表現されている。 T_{TC} は Time-Code の周期を、 s は固定遅延を、 d_1, d_2 は

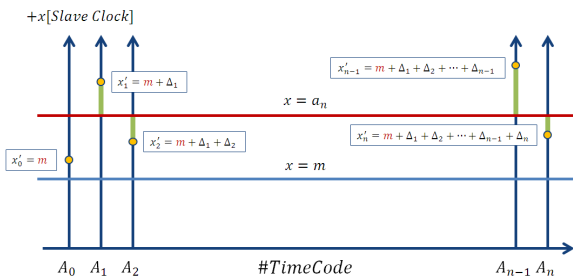


図 7 復元した遅延郡とその平均 ($x'_0 = m$)

変動遅延を表す。DEDA ユニットでは青い矢印の差分を 2.3 節で定義したドリフト誤差を含んだ形で観測できる。青い矢印の T_{TC} 分の時間に対するドリフト誤差を D_{TC} , $(d_2 - d_1)$ 分の時間に対するドリフト誤差を D_{NonTC} とすると、観測差分は以下の式で表せる。

$$T_{TC} + (d_2 - d_1) + D_{TC} + D_{NonTC} \quad (3.1)$$

通常 Time-Code の周期 T_{TC} は前もって定めた固定値になるので、DEDA ユニット内の時間における Time-Code 周期 $T_{TC(Slave)} = T_{TC} - D_{TC}$ を除去することができる。よって、観測差分は最終的に以下の式で表せる。

$$T_{TC} + (d_2 - d_1) + D_{TC} + D_{NonTC} - T_{TC(Slave)} \\ = (d_2 - d_1) + 2D_{TC} + D_{NonTC} \quad (3.2)$$

DEDA ユニットでは Time-Code が到着するたびに 3.2 式の観測差分のサンプルを算出する。なお、Clock Source の分解能未満の観測差分の情報は切り捨てられる。

3.2 ドリフト推定

DEDA ユニットでは算出した観測差分のサンプルからドリフト誤差 $2D_{TC}$ を推定する。推定は観測差分のサンプルの平均を計算することで行う。

3.2 式の観測差分には変動遅延の変化量 $(d_2 - d_1)$ が含まれる。変動遅延の変化量はトラフィックが変化しない時、中心 0 の y 軸対称な分布に従う。トラフィックが変化する時は中心が 0 から少しズレた、ほぼ y 軸対称な分布に従う。中心からのズレはトラフィックが変化し続けると必ず逆側に移動する。そのため、変動遅延の変化量をサンプルして平均を取る時、その平均はサンプル数を増やすほど 0 に収束する。また変動遅延の変化量が 0 に収束するなら、変動遅延の変化量に対するドリフト誤差 D_{NonTC} も 0 に収束する。よって、変動遅延の変化量が 0 に収束するのに十分な観測差分のサンプルが得られれば、観測差分の平均を計算することで $2D_{TC}$ の平均を抽出できる。サンプル区間において Clock Source の刻み幅の変動が微小であれば、抽出した $2D_{TC}$ の平均値をサンプル区間全域における $2D_{TC}$ の推定値として採用できる。 $2D_{TC}$ の推定値は次節の変動遅延推定で利用する。

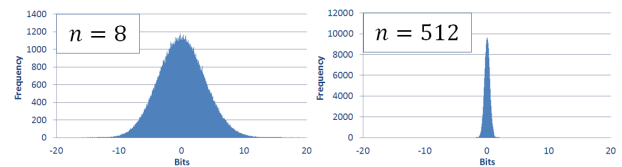


図 8 x_n の推定誤差の分布の一例 (ドリフト誤差の発生なし時)

3.3 変動遅延推定

Clock Source でドリフト誤差が発生しない理想的な場合の変動遅延の推定方法を説明してから、ドリフト誤差が発生する場合の DEDA ユニットで行う推定方法を説明する。

マスターが $A_0 \sim A_n$ までの $n+1$ 個の Time-Code を送信し、それぞれに対して変動遅延 $x_0 \sim x_n$ が発生したとする。ドリフト誤差が発生しない場合の k 番目の観測差分は $\Delta_k = (d_k - d_{k-1})$ で表せる。 $\Delta_1 \sim \Delta_n$ が情報落ちせずに得られた時、直近の変動遅延 x_n の推定方法を説明する。

x_n の推定は x_0 を推定してから間接的に行う。具体的には x_0 の推定値に観測差分 $\Delta_1 \sim \Delta_n$ を足して x_n を推定する。 x_0 を推定するため、 x_0 が $x_0 \sim x_n$ の平均の最頻値 m に等しいと仮定する。 x_0 の仮定値を $x'_0 = m$ とすると、この仮定値と観測差分 $\Delta_1 \sim \Delta_n$ から以下のような遅延郡 $x'_1 \sim x'_n$ を復元できる。

$$x'_1 = m + \Delta_1 \\ x'_2 = m + \Delta_1 + \Delta_2 \\ \dots \\ x'_{n-1} = m + \Delta_1 + \Delta_2 + \dots + \Delta_{n-1} \\ x'_n = m + \Delta_1 + \Delta_2 + \dots + \Delta_{n-1} + \Delta_n \quad (3.3)$$

次に、復元した遅延郡の平均 a_n を算出する。 a_n が m より大きい場合を図 7 に示す。仮定値 x'_0 を $m+1$ に変えると、復元した遅延郡の平均が a_n+1 になることが分かる。 $x'_0 = m$ を $x'_0 = m - (a_n - m) = 2m - a_n$ に変えると、復元した遅延郡の平均値が m になる。 $x_0 \sim x_n$ の平均が最も高い確率で m になる状況において、 $2m - a_n$ が x_0 の推定値としてもっともらしい。この推定方法は a_n が m 以下の場合にも有効である。 x_0 の推定値が得られたので、観測差分を足すことで以下のように x_n を推定できる。

$$2m - a_n + \Delta_1 + \Delta_2 + \dots + \Delta_{n-1} + \Delta_n \quad (3.4)$$

図 8 に x_n の推定誤差の分布の一例を示す。図 8 は 4.3 節の実験と同じ SpaceWire の構成で、ドリフト誤差が発生しない場合の x_n の推定誤差の分布を表す。横軸は推定誤差のビット数を、縦軸は誤差の発生頻度を表す。ドリフト誤差が発生しない場合は、観測差分のサンプル数 n を増やすほど x_n の推定誤差は小さくなる。 n を増やすほど $x_0 \sim x_n$ の平均が最頻値 m に偏るためである。仮定値 m を $x_0 \sim x_n$ の平均の最頻値と異なる値にした場合は、その差だけ x_n の推定誤差の分布が平行移動する。

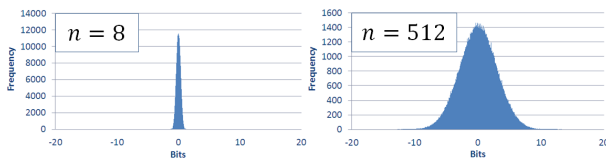


図 9 量子化誤差の分布の一例

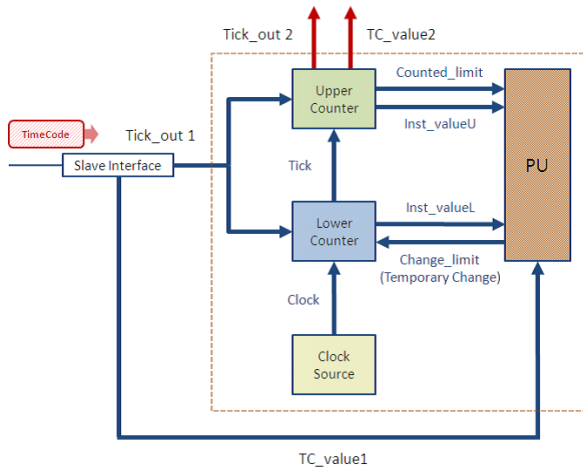


図 10 DEDA ユニットの内部構造

ドリフト誤差が発生する場合の k 番目の観測差分は $\Delta_k = (d_k - d_{k-1}) + 2D_{TCk} + D_{NonTCk}$ で表せる。ドリフト誤差が発生する場合、 Δ_k は変動遅延の変化量以外の項を含む。そこで Δ_k の代わりに、 Δ_k から前節で説明した $2D_{TC}$ の推定値を減算した δ_k を用いて、ドリフト誤差が発生しない場合と同じ手順で x_n の推定を行う。

δ_k を用いると n を増やすほど大きくなる追加の誤差が生じる。図 9 にその誤差の分布の一例を示す。図 9 は 4.3 節の実験と同じ SpaceWire の構成で発生する追加の誤差を表す。横軸は推定誤差のビット数を、縦軸は誤差の発生頻度を表す。DEDA ユニットでは Clock Source の分解能未満の Δ_k の情報は切り捨てられる。そのため、 Δ_k から $2D_{TC}$ の推定値を減算した δ_k に量子化誤差が発生する。この量子化誤差は n を増やすほど大きくなる。ドリフト誤差が発生する場合にできる限り少ない推定誤差で変動遅延を推定するには、図 8 と図 9 の誤差を考慮してサンプル数 n を決める必要がある。

3.4 出カタイミングの調整

図 10 に DEDA ユニットの内部構造を示す。DEDA ユニットは Clock Source, Lower Counter, Upper Counter, PU (Processing Unit) で構成される。Lower Counter は Clock が入力されるとカウントアップし、Clock Source の周波数 ÷ Time-Code の周波数に設定された上限値に達すると Tick を出力して 0 に戻る。Upper Counter は Tick が入力されると Tick_out2 を出力してカウントアップし、カウンタ値を TC_value2 に出力する。上限値は Time-Code の時刻情報と同じ $2^6 = 64$ に設定され、上限値に達したら PU

に Counted_limit を出力して 0 に戻る。Time-Code が到着し Tick_out1 が入力されると、その瞬間の Lower Counter, および, Upper Counter の瞬間の値を PU に出力する。PU はその瞬間の値を使って観測差分を算出する。観測差分からドリフト誤差 $2D_{TC}$ の推定値を求め、届いた Time-Code の変動遅延を推定したら、次の Time-Code の送信依頼タイミングを算出する。算出したタイミングに Tick_out2 と TC_value2 が出力されるように出力タイミングを調整する。出力を早める場合は Change_limit を使って Lower Counter の上限値を一時的に下げ、出力を遅める場合は上限値を一時的に上げる。上限値の変更は次に Lower Counter が上限値に達するまで一時的に実施される。DEDA ユニットはこのようにして時刻遅延を緩和している。

4. 実験

本章では、実験に用いるシミュレータの説明をした後、ドリフト推定実験と遅延緩和実験について述べる。

4.1 TCsim

TCsim はマスターとスレーブの時間、および、DEDA ユニットの処理を模倣するシミュレータである。Time-Code がスレーブに到着するイベントによって内部状態が変化するイベントドリブン型のシミュレータとして実装してある。時刻遅延とドリフト誤差による時間のズレを模倣するので、DEDA ユニットによって時刻遅延をどの程度緩和できるか確認できる。パラメータを変更することで様々な SpaceWire の構成をシミュレートできる。

4.2 ドリフト推定実験

ASTRO-H を意識した SpaceWire の構成でドリフト誤差 $2D_{TC}$ をどの程度推定できるか確認するため、4.1 節で述べた TCsim のシミュレーションによる実験を行った。実験で用いた TCsim のパラメータを表 1 に示す。表 1 のパラメータで、ドリフト推定のサンプル区間を 64, 4096, 65536 の 3 パターンに設定してシミュレーションを行った。実験の周波数安定度モデルに DSB321SD モデル [8] を用いた。また、温度モデルに IDEA 衛星内部の高温時を想定した 6000sec 周期のモデル [9] を用いた。常時変動モデルは 12.5%刻みで、データ送信率 0%~100% を往復するモデルである。シミュレーションで得られるドリフト誤差 $2D_{TC}$ の推定値を 1sec 間隔で温度モデル 1 周期分グラフにまとめ、TCsim 内部で生成された真値と比較した。

実験結果を図 11 に示す。横軸はシミュレーション時間を、縦軸はドリフト誤差の大きさを表す。各線の横にある S はドリフト推定のサンプル区間を意味する。Answer はドリフト誤差 $2D_{TC}$ の真値を表す。サンプル区間を広げても Answer から約 15nsec 小さい値に収束している。Clock Source の分解能 31.25nsec 未満の情報が切り捨てられ、そ

表 1 ASTRO-H を意識した SpaceWire の構成

パラメータ	パラメータ値
シミュレーション時間 [sec]	8000
擬似乱数シード	29
ルータ数	10
Time-Code 周波数 [Hz]	64
SpaceWire リンクレート [MHz]	25
固定遅延 [nsec/hop]	700
Clock Source 周波数 [MHz]	32
常温偏差 [ppm]	-100
周波数安定度モデル (対温度)	DSB321SD モデル
温度モデル	IDEA 衛星高温モデル
トラフィックモデル	常時変動モデル

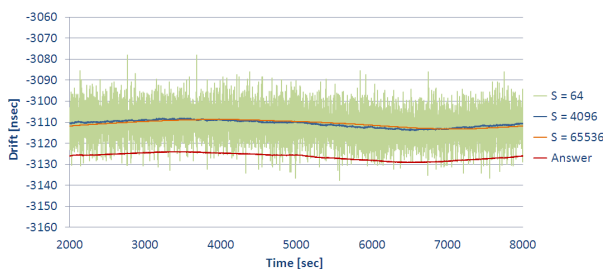


図 11 ドリフト推定実験の結果

の平均が約 15nsec になったと考えられる。情報の切り捨ては分解能に対してランダムに発生していたと推測できる。Clock Source の分解能の半分程度の推定誤差でドリフト誤差 $2D_{TC}$ を推定できることが確認できた。

4.3 遅延緩和実験

ASTRO-H を意識した SpaceWire の構成で時刻遅延をどの程度緩和できるか確認するため、4.1 節で述べた TCsim のシミュレーションによる実験を行った。前節と同じ表 1 のパラメータでドリフト推定のサンプル区間を 4096 に、変動遅延推定のサンプル区間を 64 に、仮定値を 47.5bit に設定してシミュレーションを行った。結果を 2.4 節で述べた ASTRO-H の手法適用時と比較した。比較には時刻のズレのヒストグラム、マイナス方向の最大値、プラス方向の最大値、標準偏差を用いた。

実験結果を図 12 と表 2 に示す。図 12 の横軸は時刻のズレを、縦軸は頻度を表している。薄い赤が提案手法適用時を、濃い赤 ASTRO-H の手法適用時を表している。ヒストグラムの階級間隔は 0.1bit に設定した。表 2 は小数点第 1 位を四捨五入した値を表している。図 12 から、時刻遅延による時刻のズレを緩和できていることが確認できた。また表 2 から、プラス・マイナス両方向の時刻のズレを約 15% に緩和できること、時刻のズレの標準偏差を約 17% に減らせることが確認できた。

5. おわりに

本論文では、変動遅延の確率分布の特徴を利用した時刻

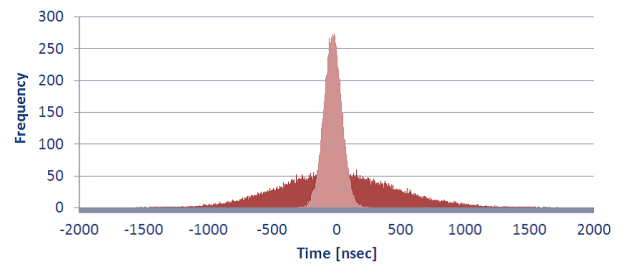


図 12 時刻のズレのヒストグラム

表 2 遅延緩和実験の結果

	提案手法	ASTRO-H の手法
マイナス方向の最大値 [nsec]	-297	-1552
プラス方向の最大値 [nsec]	251	1713
標準偏差	65	393

遅延の緩和手法、すなわち、時間の同期精度向上手法について述べた。時刻遅延の緩和はスレーブに DEDA ユニットの追加することで実現した。実験から、DEDA ユニットのよって時刻遅延を緩和できることが確認できた。

今後の課題として FPGA での実装・評価が挙げられる。一般的な FPGA のクロック源は実験で用いたものより刻み幅の変動が大きい。観測差分の平均によるドリフト推定では変化に追従できないと予想される。そのため、直近の観測差分に重きを置いた平均でない統計処理によってドリフト推定を行う必要がある。FPGA でも提案手法が有効であれば、DEDA ユニットの IP コアとして提供できる。

参考文献

- [1] ECSS: ECSS-E-ST-50-12C SpaceWire – Links, nodes, routers and networks, European Cooperation for Space Data Standardization (online), available from (<http://www.ecss.nl>) (accessed 2014-02-11).
- [2] Parkes, S.: The Operation and Uses of the SpaceWire Time-Code, *ISWS International SpaceWire Seminar*, pp. 223–230 (2003).
- [3] Suess, M. and Siegle, F.: SpaceWire Time Code Latency and Jitter, *SpaceWire Fifth International Conference*, pp. 357–362 (2013).
- [4] Pinsard, F. and Cara, C.: High Resolution Time Synchronization over SpaceWire Links, *Aerospace Conference, IEEE*, pp. 1–8 (2008).
- [5] Cook, B. M.: Reducing SpaceWire Time-code Jitter, 4Links Limited (online), available from (<http://www.4links.co.uk/>) (accessed 2014-02-11).
- [6] Gorbachev, S., Koblyakova, L. et al.: Distributed Interrupt Signalling for SpaceWire, *SpaceWire Fifth International Conference*, pp. 35–41 (2013).
- [7] 岩瀬かほり: ASTRO-H 衛星における SpaceWire を利用した時刻情報配信の精度検証, 埼玉大学 (オンライン), 入手先 (<http://astro-h.isas.jaxa.jp/>) (参照 2014-02-11).
- [8] 株式会社大真空: 小型 SMD 温度補償水晶発振器の最新技術, 株式会社大真空 (オンライン), 入手先 (<http://www.kds.info/index.htm>) (参照 2014-02-11).
- [9] IDEA Project: IDEA The Project for In-situ Debris Environmental Awareness, 九州大学 (online), available from (<http://idea.aero.kyushu-u.ac.jp/>) (accessed 2014-02-11).