

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4097891号
(P4097891)

(45) 発行日 平成20年6月11日(2008.6.11)

(24) 登録日 平成20年3月21日(2008.3.21)

(51) Int.Cl.		F I			
H04L	12/28	(2006.01)	H04L	12/28	200Z
H04L	7/00	(2006.01)	H04L	7/00	Z

請求項の数 20 (全 30 頁)

(21) 出願番号	特願2000-360279 (P2000-360279)	(73) 特許権者	000006013
(22) 出願日	平成12年11月27日(2000.11.27)		三菱電機株式会社
(65) 公開番号	特開2002-164903 (P2002-164903A)		東京都千代田区丸の内二丁目7番3号
(43) 公開日	平成14年6月7日(2002.6.7)	(74) 代理人	100089118
審査請求日	平成17年10月21日(2005.10.21)		弁理士 酒井 宏明
		(72) 発明者	仲井 勲
			東京都千代田区丸の内二丁目2番3号 三
			菱電機株式会社内
		(72) 発明者	鈴木 健司
			東京都千代田区丸の内二丁目2番3号 三
			菱電機株式会社内
		(72) 発明者	茅野 眞一郎
			東京都千代田区丸の内二丁目2番3号 三
			菱電機株式会社内

最終頁に続く

(54) 【発明の名称】 I E E E 1 3 9 4 を用いた同期システム

(57) 【特許請求の範囲】

【請求項 1】

I E E E 1 3 9 4 を用いて接続されたネットワーク内の各ノード間の同期をとる I E E E 1 3 9 4 を用いた同期システムにおいて、

前記各ノードが有する各サイクルタイム同士の遅延ずれ量を計測するサイクルタイム遅延計測手段と、

前記サイクルタイム遅延計測手段が計測した遅延ずれ量をもとに前記各サイクルタイム同士の遅延ずれを補正するサイクルタイム遅延補正手段と、

を備え、

前記サイクルタイム遅延計測手段は、要求ノードから応答ノードに対してReadパケットを送信し、該応答ノードのサイクルタイム値を読み取り、この読み取ったサイクルタイム値を含むReadResponseパケットを前記要求ノードが受信し、この受信したサイクルタイム値と要求ノードのサイクルタイム値との差をもとに前記遅延ずれ量を算出する、

ことを特徴とする I E E E 1 3 9 4 を用いた同期システム。

【請求項 2】

前記要求ノードは、

前記Readパケットを送信し、前記ReadResponseパケットが返送される一連のシーケンスが、同一のアシクロナス転送が行われている時間帯である同一のAsynchronous転送時間帯で完了するか否かを判定する判定手段を備え、

前記判定手段が、同一のAsynchronous転送時間帯で完了すると判定した場合に、Readパ

10

20

ケットを前記応答ノードに送出することを特徴とする請求項 1 に記載の I E E E 1 3 9 4 を用いた同期システム。

【請求項 3】

前記応答ノードの上位層は、

アシンクロナス転送が行われている時間帯であるAsynchronous転送時間帯であるか否かを判定する判定手段を備え、

前記判定手段がAsynchronous転送時間帯であると判定した場合に、前記上位層が前記ReadResponseパケットを送信バッファに蓄えることを特徴とする請求項 1 に記載の I E E E 1 3 9 4 を用いた同期システム。

【請求項 4】

前記応答ノードのリンク層は、

パケットの送出直前に該パケットにデータを書き込む書込手段を備え、

前記ReadResponseパケットがリンク層バッファから送出される直前に前記書込手段によって該ReadResponseパケットに当該応答ノードのサイクルタイム値を書き込むことを特徴とする請求項 1 に記載の I E E E 1 3 9 4 を用いた同期システム。

【請求項 5】

各ノードは、前記遅延ずれ量を記憶するサイクルタイム遅延記憶手段を備えたことを特徴とする請求項 1 ~ 4 のいずれか一つに記載の I E E E 1 3 9 4 を用いた同期システム。

【請求項 6】

前記サイクルタイム遅延記憶手段は、全てのノード間の遅延ずれ量をデータテーブル形式で記憶することを特徴とする請求項 5 に記載の I E E E 1 3 9 4 を用いた同期システム。

【請求項 7】

前記サイクルタイム遅延記憶手段は、サイクルマスタとしてのノードに設けられたことを特徴とする請求項 6 に記載の I E E E 1 3 9 4 を用いた同期システム。

【請求項 8】

前記ネットワークの構成の変化を判別するネットワーク構成変化判別手段をさらに備え、

前記ネットワーク構成変化判別手段がネットワークの構成に変化があったと判別した場合、該ネットワークの再構成後、前記サイクルタイム遅延計測手段がサイクルタイム同士の遅延ずれ量を計測し、前記サイクルタイム遅延補正手段が該遅延ずれ量をもとにサイクルタイム同士の遅延ずれを補正する処理を一回行うことを特徴とする請求項 1 ~ 7 のいずれか一つに記載の I E E E 1 3 9 4 を用いた同期システム。

【請求項 9】

前記ネットワーク構成変化判別手段は、サイクルマスタとしてのノードに設けられたことを特徴とする請求項 8 に記載の I E E E 1 3 9 4 を用いた同期システム。

【請求項 10】

前記サイクルタイム遅延計測手段および前記サイクルタイム遅延補正手段は、ノード間の同期を用いる所定のノードに設けられたことを特徴とする請求項 1 ~ 9 のいずれか一つに記載の I E E E 1 3 9 4 を用いた同期システム。

【請求項 11】

前記遅延ずれ量を伝送するサイクルタイム遅延伝送手段をさらに備え、

同期の基準となるサイクルタイムを有するサイクルマスタのノードは、前記サイクルタイム遅延計測手段および前記サイクルタイム遅延伝送手段を有し、

前記サイクルマスタのノード以外の所定のノードは、前記サイクルタイム遅延補正手段を有し、

前記サイクルマスタのノードが、該ノードが計測した遅延ずれ量を前記所定のノードに伝送し、あるいは該ノードが計測した遅延ずれ量をデータテーブルに記憶し、前記所定のノードが前記サイクルマスタのノードから該遅延ずれ量を取得し、その後所定のノードに伝送された遅延ずれ量あるいは取得した遅延ずれ量をもとに該ノードのサイクルタイムの

10

20

30

40

50

遅延ずれを補正することを特徴とする請求項 1 ~ 9 のいずれか一つに記載の I E E E 1 3 9 4 を用いた同期システム。

【請求項 1 2】

前記サイクルタイム遅延補正手段は、サイクルスタートパケットを受信した際に、遅延ずれ量をもとにサイクルタイム値を設定することを特徴とする請求項 1 0 または 1 1 に記載の I E E E 1 3 9 4 を用いた同期システム。

【請求項 1 3】

前記サイクルタイム遅延補正手段は、上位層がサイクルタイム値の使用時に、前記遅延ずれ量を用いて遅延ずれを補正することを特徴とする請求項 1 0 または 1 1 に記載の I E E E 1 3 9 4 を用いた同期システム。

10

【請求項 1 4】

前記所定のノードは、第 2 のサイクルタイムをさらに備え、

前記サイクルタイム遅延補正手段は、任意のタイミングで遅延ずれ量を用いて前記遅延ずれを補正したサイクルタイム値を前記第 2 のサイクルタイムに設定することを特徴とする請求項 1 0 または 1 1 に記載の I E E E 1 3 9 4 を用いた同期システム。

【請求項 1 5】

前記サイクルマスタのノードは、タイムスタンプを生成するタイムスタンプ生成手段を備え、

前記所定のノードは、タイムスタンプが示す時刻を監視するタイムスタンプ時刻監視手段を備え、

20

前記サイクルタイム遅延補正手段は、前記遅延ずれ量をもとに、前記サイクルマスタのノードから送られた前記タイムスタンプのタイムスタンプ値を更新し、

前記タイムスタンプ時刻監視手段は、前記更新されたタイムスタンプ値が示す時刻を監視することを特徴とする請求項 1 0 または 1 1 に記載の I E E E 1 3 9 4 を用いた同期システム。

【請求項 1 6】

前記サイクルタイム遅延計測手段および前記サイクルタイム遅延補正手段は、サイクルマスタとしてのノードに設けられたことを特徴とする請求項 1 ~ 9 のいずれか一つに記載の I E E E 1 3 9 4 を用いた同期システム。

【請求項 1 7】

30

前記遅延ずれ量を伝送するサイクルタイム遅延伝送手段をさらに備え、

サイクルマスタのノードは、前記サイクルタイム遅延補正手段を有し、

前記サイクルマスタのノード以外の所定のノードは、前記サイクルタイム遅延計測手段および前記サイクルタイム遅延伝送手段を有し、

前記所定のノードが、該所定のノードで計測した遅延ずれ量を前記サイクルマスタのノードに伝送し、あるいは前記所定のノードから遅延ずれ量を取得し、その後該所定のノードに伝送された遅延ずれ量あるいは取得した遅延ずれ量をもとに前記サイクルマスタのノードのサイクルマスタの遅延ずれを補正することを特徴とする請求項 1 ~ 9 のいずれか一つに記載の I E E E 1 3 9 4 を用いた同期システム。

【請求項 1 8】

40

前記サイクルマスタのノードは、タイムスタンプを生成するタイムスタンプ生成手段を備え、

前記所定のノードは、タイムスタンプが示す時刻を監視するタイムスタンプ時刻監視手段を備え、

前記サイクルタイム遅延補正手段は、前記遅延ずれ量をもとに、前記タイムスタンプ生成手段が生成したタイムスタンプのタイムスタンプ値を更新することを特徴とする請求項 1 6 または 1 7 に記載の I E E E 1 3 9 4 を用いた同期システム。

【請求項 1 9】

請求項 1 ~ 1 8 のいずれか一つに記載した I E E E 1 3 9 4 を用いた同期システムは、サーボドライブ装置とこれを制御するコントローラとからなる制御システムであることを

50

特徴とする I E E E 1 3 9 4 を用いた同期システム。

【請求項 2 0】

前記コントローラは、サイクルマスタであることを特徴とする請求項 1 9 に記載の I E E E 1 3 9 4 を用いた同期システム。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

この発明は、I E E E 1 3 9 4 をネットワークのデータ伝送手段として使用しているシステムにおける各ノード間のサイクルタイム遅延を測定し補償することができる I E E E 1 3 9 4 を用いた同期システムに関し、特に、1 つ以上のコントローラと、センサやサーボ
ドライブなどの 1 つ以上の被制御対象機器とによって構成される制御システムにおいて、
I E E E 1 3 9 4 をネットワークのデータ伝送手段として使用している場合におけるコント
ローラと被制御対象機器との間のサイクルタイム遅延を測定し補償することができる I
E E E 1 3 9 4 を用いた同期システムに関するものである。

10

【0 0 0 2】

【従来の技術】

従来から、コントローラとサーボドライバとの間の通信方式は、各メーカー独自の方式で行
われていた。一方、近年、このような F A 用コントローラの分野においても、オープン化
に対応した製品への要求が高まっている。オープン化コントローラの普及によって、ユー
ザ独自の組込み、汎用パソコン資産の利用が可能になり、メーカーの異なる製品間の
接続も容易になるなど、ユーザの使いやすさの向上が実現される。

20

【0 0 0 3】

このようなコントローラとサーボドライバとの間のオープンなインタフェースとして、I
E E E 1 3 9 4 や U S B を利用することが検討され始めている。特に、ホームオートメー
ション用のネットワーク規格 I E E E 1 3 9 4 を F A 用途に利用しようとする動きが本格
化してきている。I E E E 1 3 9 4 は、

- (1) 1 0 0 M バイト / 秒以上の高速のデータ転送が可能
- (2) アイスクロナス転送方式が可能で速い通信周期の同期通信が可能
- (3) 電源を切ることなく接続や切り離しが可能 (活線挿抜)
- (4) 6 3 台までの機器が接続できる

30

といった特徴を持つ。さらに量産効果によってインタフェース部分を低コストに製造でき
る可能性がある。また、インタフェース基板は、イーサネット (Ethernet) に比較して小
型なので、各種の機器に組み込みやすい。

【0 0 0 4】

図 2 3 は、I E E E 1 3 9 4 の通信レイヤの構成を示す図である。図 2 3 において、I E
E E 1 3 9 4 の通信レイヤは、物理層 (Phy レイヤ) 1 0 1、リンク層 (Link レイヤ) 1
0 2、トランザクション層 (Transaction レイヤ) 1 0 3 の 3 層と、シリアルバスマネー
ジメント 1 0 4 とから構成される。

【0 0 0 5】

Phy レイヤ 1 0 1 は、送受信する電気信号と Link レイヤとの間で信号処理を行うレイヤで
ある。Phy レイヤ 1 0 1 は、コネクタ、ケーブルなどの機械的インターフェースの規定、L
ink レイヤ 1 0 2 が使用する論理信号のアナログディジタル変換を行うエンコードおよび
デコード、通信信号の電気レベルを決めるシグナルレベルなどの電氣的インターフェース
の規定、通信ノード決定のためのアービトレーション、通信ロックの再同期、バスの初期
化検出などを行っている。

40

【0 0 0 6】

Link レイヤ 1 0 2 は、Phy レイヤ 1 0 1 と Transaction レイヤ 1 0 3 との間で信号処理を行
うレイヤである。Link レイヤ 1 0 2 は、アドレス割当、データのチェック、データのフレ
ーム分けを行うパケット送受信、サイクルのコントロールを行っている。なお、Isochron
ous データは、Transaction レイヤを経由せずに、データが処理される。

50

【 0 0 0 7 】

Transactionレイヤ 1 0 3 は、上位アプリケーションとLinkレイヤ 1 0 2 との間で信号処理を行うレイヤである。Transactionレイヤ 1 0 3 は、データの読み出し、書き込みなどの処理を行う。すなわち、Transactionレイヤ 1 0 3 は、Linkレイヤ 1 0 2 の処理を利用して、リクエストパケットの送信とレスポンスパケットの受信を行うことによって、指定されたノードとアドレスに対する 1 つの通信処理を行っている。また、逆に、リクエストパケットの受信とレスポンスパケットの送信という、他のノードから自ノードへの通信処理を行っている。

【 0 0 0 8 】

シリアルバスマネージメント 1 0 4 は、3 つのレイヤ全体の制御を行っている。なお、典型的な通信レイヤの実装としては、Phyレイヤ 1 0 1 とLinkレイヤ 1 0 2 とをハードウェアで構成し、Transactionレイヤ 1 0 3 とシリアルバスマネージメント 1 0 4 をファームウェアで構成する。

【 0 0 0 9 】

I E E E 1 3 9 4 のAsynchronous通信は、非同期に、データの通信を行う際に用いられる。Asynchronous通信では、相手ノードに確実にパケットを送信することを保証し、送信の遅延時間は保証しない。送信ノードは、ヘッダ情報と実データとを指定先のノードに送信し、受信ノードは、Acknowledgeパケットを返信することで、パケットを受け取ったことを知らせる。

【 0 0 1 0 】

I E E E 1 3 9 4 のIsochronous通信は、一種の同期通信であり、動画や音声などの伝送に適している。Isochronous通信では、1 2 5 μ sec毎にデータ転送が完了することを保証する。Isochronousパケットの送信ノードは、特定のノードに送信するのではなく、0 ~ 6 3 までのチャンネル番号を使用してバス全体に送信し、受信ノードは、Acknowledgeパケットを返信しない。Isochronousパケットのヘッダは、6 ビットのチャンネル番号を使用することで、パケットを識別できるため、Asynchronousパケットのヘッダのように、6 4 ビットのアドレス空間を使うのに比べて単純化されており、ヘッダ情報を少なくすることができる。受信ノードは、自ノードが受信したいチャンネル番号のIsochronousパケットを選択して取り込む。送信ノードと受信ノードとが同じチャンネル番号を使ってデータの送受信を行うことで、Isochronous通信が確立できる。

【 0 0 1 1 】

図 2 4 に示すように、Isochronous通信とAsynchronous通信とは混在できる。1 サイクル 1 2 5 μ secのうち、最大で 1 0 0 μ secをIsochronous通信に使用し、残りをAsynchronous通信に使用する。なお、パケット転送速度によって、Asynchronousパケットのデータペイロードの最大値が決定されているのは、Asynchronousデータの伝送時間が長すぎて、Isochronous通信を保証できなくなるのを回避するためである。

【 0 0 1 2 】

Isochronous転送時間帯は、Isochronous通信を行う帯域であり、Asynchronous転送時間帯は、Asynchronous通信を行う帯域である。図 2 4 に示すように、Isochronous通信が常にAsynchronous通信よりも先に開始され、Isochronous通信を保証することができる。1 サイクルでは、サイクルスタートパケット D S が送信された後に、図 2 4 では、Isochronous転送時間帯において、チャンネル C H 1 のデータパケット D 1 およびチャンネル C H 3 のデータパケット D 2 が伝送される。その後、Asynchronous転送時間帯において、データパケット D 3 , D 4 が伝送されるが、各データパケット D 3 , D 4 には、Acknowledgeパケット D A が返送される。

【 0 0 1 3 】

I E E E 1 3 9 4 の各ノードのうち、アイソクロナス (Isochronous) 能力のあるノードが搭載するLinkレイヤは、バスとクロック同期するためのCYCLE__TIMEレジスタを実装し、このCYCLE__TIMEレジスタの内容を最新にするためのタイマを有している。このタイマをサイクルタイマと呼ぶ。このサイクルタイマの時刻は、サイクルスタート (Cycle Star

10

20

30

40

50

t) パケットによって設定される。また、I E E E 1 3 9 4では、このサイクルスタートパケットを発信するノードのことを、サイクルマスタと呼び、このサイクルマスタ以外のノードは、サイクルスレーブと呼ばれる。

【 0 0 1 4 】

すなわち、図 2 5 および図 2 6 に示すように、サイクルマスタ 1 1 1 , 1 3 1 が有するサイクルタイマ 1 1 2 , 1 3 2 の時刻をサイクルスタートパケット「CycleStart」に乗せて各ノード(サイクルスレーブ) 1 2 1 , 1 4 1 , 1 5 1 に伝えることによって、各ノード 1 2 1 , 1 4 1 , 1 5 1 が有するサイクルタイマ 1 2 2 , 1 4 2 , 1 5 2 の時刻がサイクルマスタ 1 1 1 , 1 3 1 が有するサイクルタイマ 1 1 2 , 1 3 2 と同じ時刻に設定される。

10

【 0 0 1 5 】

I E E E 1 3 9 4 を利用したリアルタイムデータ転送は、Isochronousパケットによって伝送されるが、サイクルタイマを基準にして、送信側と受信側でデータの同期を取っている。同期の方法としてIsochronousパケットにタイムスタンプを入れたりする。

【 0 0 1 6 】

【発明が解決しようとする課題】

しかしながら、実際には伝送遅延があるため、送信側のノードと受信側のノードのサイクルタイマは厳密に同じ時刻にはならない。すなわち、応答ノードがサイクルスタートパケットを受信してサイクルタイマの時刻を設定した瞬間には、要求ノードのサイクルタイマの時刻はその分だけ既に進んでおり、伝送遅延によって、送信側のノードと受信側のノードとの間の同期が精度良くとれないことになる。

20

【 0 0 1 7 】

特に、I E E E 1 3 9 4 によって接続されたネットワークを用いた制御システムを考えた場合、コントローラとサーボドライブとは、これらサイクルタイマを基準として同期制御を実現することになる。しかしながら、この制御システムに対しては、高速高精度、特に同期制御の高速高精度が要求されており、そのようなコントローラとサーボドライブとの間のタイマ遅延が、制御の高速高精度、特に同期制御の高速高精度に大きな影響を与えてしまうという問題点があった。

【 0 0 1 8 】

この発明は上記に鑑みてなされたもので、送信側のノードと受信側のノードとが有するサイクルタイマのタイマ遅延を補償し、特にI E E E 1 3 9 4 をコントローラとサーボドライブとの間のオープンなインタフェースとして使用した場合でも、高速高精度な制御、特に同期制御の高速高精度を実現することができるI E E E 1 3 9 4 を用いた同期システムを得ることを目的とする。

30

【 0 0 1 9 】

【課題を解決するための手段】

上記目的を達成するため、この発明にかかるI E E E 1 3 9 4 を用いた同期システムは、I E E E 1 3 9 4 を用いて接続されたネットワーク内の各ノード間の同期をとるI E E E 1 3 9 4 を用いた同期システムにおいて、前記各ノードが有する各サイクルタイマ同士の遅延ずれ量を計測するサイクルタイマ遅延計測手段と、前記サイクルタイマ遅延計測手段が計測した遅延ずれ量をもとに前記各サイクルタイマ同士の遅延ずれを補正するサイクルタイマ遅延補正手段とを備え、前記サイクルタイマ遅延計測手段は、要求ノードから応答ノードに対してReadパケットを送信し、該応答ノードのサイクルタイマ値を読み取り、この読み取ったサイクルタイマ値を含むReadResponseパケットを前記要求ノードが受信し、この受信したサイクルタイマ値と要求ノードのサイクルタイマ値との差をもとに前記遅延ずれ量を算出することを特徴とする。

40

【 0 0 2 0 】

この発明によれば、I E E E 1 3 9 4 を用いて接続されたネットワーク内の各ノード間の同期をとる際、サイクルタイマ遅延計測手段が、前記各ノードが有する各サイクルタイマ同士の遅延ずれ量を計測し、要求ノードから応答ノードに対してReadパケットを送信し

50

、該応答ノードのサイクルタイム値を読み取り、この読み取ったサイクルタイム値を含むReadResponseパケットを前記要求ノードが受信し、この受信したサイクルタイム値と要求ノードのサイクルタイム値との差をもとに前記遅延ずれ量を算出するようにしていると共に、サイクルタイム遅延補正手段が、前記サイクルタイム遅延計測手段が計測した遅延ずれ量をもとに前記各サイクルタイム同士の遅延ずれを補正するようにしている。

【 0 0 3 1 】

つぎの発明にかかる I E E E 1 3 9 4 を用いた同期システムは、上記の発明において、前記要求ノードは、前記Readパケットを送信し、前記ReadResponseパケットが返送される一連のシーケンスが、同一のアシクロナス転送が行われている時間帯である同一のAsynchronous転送時間帯で完了するか否かを判定する判定手段を備え、前記判定手段が、一連のシーケンスが同一のAsynchronous転送時間帯で完了すると判定した場合に、Readパケットを前記応答ノードに送出することを特徴とする。

10

【 0 0 3 2 】

この発明によれば、前記要求ノードの判定手段が、前記Readパケットを送信し、前記ReadResponseパケットが返送される一連のシーケンスが、同一のアシクロナス転送が行われている時間帯である同一のAsynchronous転送時間帯で完了するか否かを判定し、前記要求ノードが、前記判定手段が同一のAsynchronous転送時間帯で完了すると判定した場合に、Readパケットを前記応答ノードに送出するようにし、ReadResponseパケットが同一のAsynchronous転送時間帯内で返送されるようにしている。

20

【 0 0 3 3 】

つぎの発明にかかる I E E E 1 3 9 4 を用いた同期システムは、上記の発明において、前記応答ノードの上位層は、アシクロナス転送が行われている時間帯であるAsynchronous転送時間帯であるか否かを判定する判定手段を備え、前記判定手段がAsynchronous転送時間帯であると判定した場合に、前記上位層が前記ReadResponseパケットを送信バッファに蓄えることを特徴とする。

【 0 0 3 4 】

この発明によれば、前記応答ノードの上位層の判定手段が、アシクロナス転送が行われている時間帯であるAsynchronous転送時間帯であるか否かを判定し、前記判定手段がAsynchronous転送時間帯であると判定した場合に、前記上位層は前記ReadResponseパケットを送信バッファ (TX_FIFO) に蓄えることで、前記ReadResponseパケットは、送信バッファ長くとどまることなく、直ちに要求ノードに返送されるようにしている。

30

【 0 0 3 5 】

つぎの発明にかかる I E E E 1 3 9 4 を用いた同期システムは、上記の発明において、前記応答ノードのリンク層は、パケットの送出直前に該パケットにデータを書き込む書込手段を備え、前記ReadResponseパケットがリンク層バッファから送出される直前に前記書込手段によって該ReadResponseパケットに当該応答ノードのサイクルタイム値を書き込むことを特徴とする。

【 0 0 3 6 】

この発明によれば、前記応答ノードのリンク層の書込手段が、前記ReadResponseパケットがリンク層バッファから送出される直前に該ReadResponseパケットに当該応答ノードのサイクルタイム値を書き込むようにしている。

40

【 0 0 3 7 】

つぎの発明にかかる I E E E 1 3 9 4 を用いた同期システムは、上記の発明において、各ノードは、前記遅延ずれ量を記憶するサイクルタイム遅延記憶手段を備えたことを特徴とする。

【 0 0 3 8 】

この発明によれば、各ノードのサイクルタイム遅延記憶手段が、前記遅延ずれ量を記憶し、少なくとも他のノードからのアクセスによって遅延ずれ量を獲得できるようにしている。

【 0 0 3 9 】

50

つぎの発明にかかる I E E E 1 3 9 4 を用いた同期システムは、上記の発明において、前記サイクルタイム遅延記憶手段は、全てのノード間の遅延ずれ量をデータテーブル形式で記憶することを特徴とする。

【 0 0 4 0 】

この発明によれば、前記サイクルタイム遅延記憶手段が、全てのノード間の遅延ずれ量をデータテーブル形式で記憶するようにしている。

【 0 0 4 1 】

つぎの発明にかかる I E E E 1 3 9 4 を用いた同期システムは、上記の発明において、前記サイクルタイム遅延記憶手段は、サイクルマスタとしてのノードに設けられたことを特徴とする。

10

【 0 0 4 2 】

この発明によれば、前記サイクルタイム遅延記憶手段を、サイクルマスタとしてのノードに設けるようにしている。

【 0 0 4 3 】

つぎの発明にかかる I E E E 1 3 9 4 を用いた同期システムは、上記の発明において、前記ネットワークの構成の変化を判別するネットワーク構成変化判別手段をさらに備え、前記ネットワーク構成変化判別手段がネットワークの構成に変化があったと判別した場合、該ネットワークの再構成後、前記サイクルタイム遅延計測手段がサイクルタイム同士の遅延ずれ量を計測し、前記サイクルタイム遅延補正手段が該遅延ずれ量をもとにサイクルタイム同士の遅延ずれを補正する処理を一回行うことを特徴とする。

20

【 0 0 4 4 】

この発明によれば、ネットワーク構成変化判別手段が、前記ネットワークの構成の変化を判別し、前記ネットワーク構成変化判別手段がネットワークの構成に変化があったと判別した場合、該ネットワークの再構成後、前記サイクルタイム遅延計測手段がサイクルタイム同士の遅延ずれ量を計測し、前記サイクルタイム遅延補正手段が該遅延ずれ量をもとにサイクルタイム同士の遅延ずれを補正する処理を一回行うようにしている。

【 0 0 4 5 】

つぎの発明にかかる I E E E 1 3 9 4 を用いた同期システムは、上記の発明において、前記ネットワーク構成変化判別手段は、サイクルマスタとしてのノードに設けられたことを特徴とする。

30

【 0 0 4 6 】

この発明によれば、前記ネットワーク構成変化判別手段を、サイクルマスタとしてのノードに設けるようにしている。

【 0 0 4 7 】

つぎの発明にかかる I E E E 1 3 9 4 を用いた同期システムは、上記の発明において、前記サイクルタイム遅延計測手段および前記サイクルタイム遅延補正手段は、ノード間の同期を用いる所定のノードに設けられたことを特徴とする。

【 0 0 4 8 】

この発明によれば、前記サイクルタイム遅延計測手段および前記サイクルタイム遅延補正手段を、ノード間の同期を用いる所定のノードに設けるようにしている。

40

【 0 0 4 9 】

つぎの発明にかかる I E E E 1 3 9 4 を用いた同期システムは、上記の発明において、前記遅延ずれ量を伝送するサイクルタイム遅延伝送手段をさらに備え、同期の基準となるサイクルタイムを有するサイクルマスタのノードは、前記サイクルタイム遅延計測手段および前記サイクルタイム遅延伝送手段を有し、前記サイクルマスタのノード以外の所定のノードは、前記サイクルタイム遅延補正手段を有し、前記サイクルマスタのノードが、該ノードが計測した遅延ずれ量を前記所定のノードに伝送し、あるいは該ノードが計測した遅延ずれ量をデータテーブルに記憶し、前記所定のノードが前記サイクルマスタのノードから該遅延ずれ量を取得し、その後所定のノードに伝送された遅延ずれ量あるいは取得した遅延ずれ量をもとに該ノードのサイクルタイムの遅延ずれを補正することを特徴とする。

50

【 0 0 5 0 】

この発明によれば、同期の基準となるサイクルタイムを有するサイクルマスタのノードが、該ノードが計測した遅延ずれ量を前記サイクルタイム遅延伝送手段によって前記所定のノードに伝送し、あるいは該ノードが計測した遅延ずれ量をデータテーブルに記憶し、前記所定のノードが前記サイクルマスタのノードから該遅延ずれ量を取得し、その後所定のノードに伝送された遅延ずれ量あるいは取得した遅延ずれ量をもとに該ノードのサイクルタイムの遅延ずれを補正するようにしている。

【 0 0 5 1 】

つぎの発明にかかる I E E E 1 3 9 4 を用いた同期システムは、上記の発明において、前記サイクルタイム遅延補正手段は、サイクルスタートパケットを受信した際に、遅延ずれ量をもとにサイクルタイム値を設定することを特徴とする。

10

【 0 0 5 2 】

この発明によれば、前記サイクルタイム遅延補正手段が、サイクルスタートパケットを受信した際に、遅延ずれ量をもとにサイクルタイム値を設定するようにしている。

【 0 0 5 3 】

つぎの発明にかかる I E E E 1 3 9 4 を用いた同期システムは、上記の発明において、前記サイクルタイム遅延補正手段は、上位層がサイクルタイム値の使用時に、前記遅延ずれ量を用いて遅延ずれを補正することを特徴とする。

【 0 0 5 4 】

この発明によれば、前記サイクルタイム遅延補正手段が、上位層がサイクルタイム値の使用時に、前記遅延ずれ量を用いて遅延ずれを補正するようにしている。

20

【 0 0 5 5 】

つぎの発明にかかる I E E E 1 3 9 4 を用いた同期システムは、上記の発明において、前記所定のノードは、第 2 のサイクルタイムをさらに備え、前記サイクルタイム遅延補正手段は、任意のタイミングで遅延ずれ量を用いて前記遅延ずれを補正したサイクルタイム値を前記第 2 のサイクルタイムに設定することを特徴とする。

【 0 0 5 6 】

この発明によれば、所定のノードの前記サイクルタイム遅延補正手段が、任意のタイミングで遅延ずれ量を用いて前記遅延ずれを補正したサイクルタイム値を第 2 のサイクルタイムに設定するようにしている。

30

【 0 0 5 7 】

つぎの発明にかかる I E E E 1 3 9 4 を用いた同期システムは、上記の発明において、前記サイクルマスタのノードは、タイムスタンプを生成するタイムスタンプ生成手段を備え、前記所定のノードは、タイムスタンプが示す時刻を監視するタイムスタンプ時刻監視手段を備え、前記サイクルタイム遅延補正手段は、前記遅延ずれ量をもとに、前記サイクルマスタのノードから送られた前記タイムスタンプのタイムスタンプ値を更新し、前記タイムスタンプ時刻監視手段は、前記更新されたタイムスタンプ値が示す時刻を監視することを特徴とする。

【 0 0 5 8 】

この発明によれば、前記サイクルタイム遅延補正手段が、前記遅延ずれ量をもとに、前記サイクルマスタのノードから送られた前記タイムスタンプのタイムスタンプ値を更新し、前記タイムスタンプ時刻監視手段が、前記更新されたタイムスタンプ値が示す時刻を監視するようにしている。

40

【 0 0 5 9 】

つぎの発明にかかる I E E E 1 3 9 4 を用いた同期システムは、上記の発明において、前記サイクルタイム遅延計測手段および前記サイクルタイム遅延補正手段は、サイクルマスタとしてのノードに設けられたことを特徴とする。

【 0 0 6 0 】

この発明によれば、前記サイクルタイム遅延計測手段および前記サイクルタイム遅延補正手段を、サイクルマスタとしてのノードに設けるようにしている。

50

【 0 0 6 1 】

つぎの発明にかかる I E E E 1 3 9 4 を用いた同期システムは、上記の発明において、前記遅延ずれ量を転送するサイクルタイム遅延伝送手段をさらに備え、サイクルマスタのノードは、前記サイクルタイム遅延補正手段を有し、前記サイクルマスタのノード以外の所定のノードは、前記サイクルタイム遅延計測手段および前記サイクルタイム遅延伝送手段を有し、前記所定のノードが、該所定のノードで計測した遅延ずれ量を前記サイクルマスタのノードに伝送し、あるいは前記所定のノードから遅延ずれ量を取得し、その後該所定のノードに伝送された遅延ずれ量あるいは取得した遅延ずれ量をもとに前記サイクルマスタのノードのサイクルマスタの遅延ずれを補正することを特徴とする。

【 0 0 6 2 】

この発明によれば、前記所定のノードが、該所定のノードで計測した遅延ずれ量を前記サイクルマスタのノードに伝送し、あるいは前記所定のノードから遅延ずれ量を取得し、その後該所定のノードに伝送された遅延ずれ量あるいは取得した遅延ずれ量をもとに前記サイクルマスタのノードのサイクルマスタの遅延ずれを補正するようにしている。

【 0 0 6 3 】

つぎの発明にかかる I E E E 1 3 9 4 を用いた同期システムは、上記の発明において、前記サイクルマスタのノードは、タイムスタンプを生成するタイムスタンプ生成手段を備え、前記所定のノードは、タイムスタンプが示す時刻を監視するタイムスタンプ時刻監視手段を備え、前記サイクルタイム遅延補正手段は、前記遅延ずれ量をもとに、前記タイムスタンプ生成手段が生成したタイムスタンプのタイムスタンプ値を更新することを特徴とする。

【 0 0 6 4 】

この発明によれば、前記サイクルタイム遅延補正手段が、前記遅延ずれ量をもとに、自サイクルマスタのノードの前記タイムスタンプ生成手段が生成したタイムスタンプのタイムスタンプ値を更新し、この更新したタイムスタンプ値を含むデータを所定のノードに伝送するようにしている。

【 0 0 6 5 】

つぎの発明にかかる I E E E 1 3 9 4 を用いた同期システムは、上記の発明において、請求項 1 ～ 2 3 のいずれか一つに記載した I E E E 1 3 9 4 を用いた同期システムは、サーバドライブ装置とこれを制御するコントローラとからなる制御システムであることを特徴とする

【 0 0 6 6 】

この発明によれば、請求項 1 ～ 2 3 のいずれか一つに記載した I E E E 1 3 9 4 を用いた同期システムを、サーバドライブ装置とこれを制御するコントローラとからなる制御システムに適用している。

【 0 0 6 7 】

つぎの発明にかかる I E E E 1 3 9 4 を用いた同期システムは、上記の発明において、前記コントローラは、サイクルマスタであることを特徴とする。

【 0 0 6 8 】

この発明によれば、前記コントローラを、サイクルマスタとし、効率的な遅延ずれ量の計測および補正ができるようにしている。

【 0 0 6 9 】

【 発明の実施の形態 】

以下、添付図面を参照して、この発明にかかる I E E E 1 3 9 4 を用いた同期システムの好適な実施の形態について説明する。

【 0 0 7 0 】

実施の形態 1 .

まず、この発明の実施の形態 1 について説明する。この実施の形態 1 では、ピングパケットを利用してサイクルタイム遅延を推定し補償するようにしている。P1394aでは、最適な Gap_Count 値を計算し設定するために、ピング (Ping) パケットが定義されている。Gap

10

20

30

40

50

(ギャップ)とは、例えば図24に示したデータパケットD2, D3間などの時間であり、バスがアイドル状態となっている時間をいい、このギャップが長くなれば、バスがアイドルとなっている時間が長くなり、実際にパケットが流れる時間が制限され、一方、ギャップを極端に短くすると、トポロジの構成で、物理層のホップ数が多い場合や接続されるノードが多い場合には通信が行えなくなってしまうため、ギャップは最適化される。

【0071】

P1394aでは、あるノード(バスマネージャ)から他のノードにこのピングパケット送信すると、このピングパケットを受信した送り先のノードの物理層(Phyレイヤ)は、Self_IDパケットを送り返す仕様になっており、ピングパケットが送出されてから、Self_IDパケットが返送されて戻ってくるまでの時間を測ることによって、2つのノード間の遅延時間がわかるようになっている。そして、上述したように、P1394aでは、この遅延時間をもとに、最適なGap_Count値を計算し設定することができるようになった。ルート以外は、すべてリンク層(Linkレイヤ)がなくても、このPingパケットに応答できるため、物理層だけで構成されるリピータでも遅延時間を知ることができ、トポロジが変わっても最適なGap_Count値を計算することができる。なお、このPingパケットは、IEEE 1394-1995規格にはないが、P1394a規格で検討中である。

【0072】

さて、この実施の形態1では、Pingパケットを利用してサイクルタイム遅延を推定し、補償する。IEEE 1394の物理層のパケットリピータ回路は、対称なため、図1に示すように、要求ノードN1の物理層から送られたPingパケットが応答ノードN2の物理層に到達するまでの伝送遅延t1は、応答ノードN2の物理層から返送されたSelf_IDパケットが要求ノードN1の物理層に到達するまでの伝送遅延t2とほぼ等しい。すなわち、伝送遅延t1 伝送遅延t2

とみなすことができる。この結果、図1に示すように、要求ノードN1の物理層がPingパケットを送出してからSelf_IDパケットを受信するまでの時間を所要時間tとすると、伝送遅延t1は、

$$t_1 (\text{伝送遅延}) = (t (\text{所要時間}) - \text{RESPONSE_TIME}) / 2$$

として求めることができる。この伝送遅延t1がそのまま、要求ノードN1が保持するサイクルタイムと応答ノードN2が保有するサイクルタイムとの間の遅延として考えることができる。なお、「RESPONSE_TIME」に実際どのぐらいの時間がかかったのかは、この手法では分からないが、「RESPONSE_TIME」の最小値および最大値は、パラメータで指定されており(Cable interface timing constants)、既知である。したがって、たとえば、RESPONSE_TIME (最大値 + 最小値) / 2

として「RESPONSE_TIME」を求めてもよい。実験的に、「RESPONSE_TIME」に実際どのぐらいの時間がかかっているのかがわかっているならば、最大値と最小値の平均の代わりに適当な重み付き平均、たとえば、係数wを用いて

$$\text{RESPONSE_TIME} = (w \times \text{最大値} + (1 - w) \times \text{最小値}) / 2$$

とし、これによって一層精度の良い「RESPONSE_TIME」を求めることができる。

【0073】

また、Pingパケットを送信し、Self_IDパケットが返ってくるまでの所要時間tを計測する代わりに、RemoteAccessパケットを送信し、RemoteReplyパケットが返ってくるまでの時間を計測し、この時間を所要時間tとしてもよい。また、RemoteCommandパケットを送信し、RemoteConfirmationパケットが返ってくるまでの時間を計測し、この時間を所要時間tとしてもよい。これらパケットは、ほぼ同じ振る舞い、すなわち物理層が返送するので、これらのどのパケットを用いても良い。これらのパケットは、サイクルマスタ、サイクルスレーブに関わらず、全てのノードが任意にノードに対して送信することができるので、任意のノード間の遅延を任意のノードが必要に応じて計測することができる。

【0074】

このノード間の遅延(タイム遅延)は各ノード間で各々異なるので、各ノード間の全ての場合についてタイム遅延の補償を行うのが望ましい。しかしながら、タイム遅延計測を頻

繁に行えば、タイマ遅延計測のためのパケットが増えることになり、その分だけネットワークのトラフィックが増えることになり、重要なパケットの伝送の妨げになりかねない。

【 0 0 7 5 】

タイマ遅延は、ネットワークの構成（トポロジ）が変わらない限り同じであるから、タイマ遅延補償は、基本的にネットワークの構成（再構成）が行われた後の最初の1回だけ行えば良いので、バスリセットなどのネットワークの再構成を含むネットワークの構成が行われたことを検出し、かつサイクルスタートパケットによるサイクルタイマ設定が行われた後に、最初に1回だけタイマ遅延計測のためのパケットを流してタイマ遅延補償を行う。

【 0 0 7 6 】

このため、ネットワーク上のある1つノード、たとえばサイクルマスタ、制御システムの場合はコントローラが、ネットワーク構成変化判別手段を保有する必要がある。ネットワーク構成変化判別手段は、ネットワークが再構成を判別した場合、タイマ遅延補償を行う指令を他のノードへ伝える。あるいは、ネットワーク構成変化判別手段を、必ずサイクルマスタ（制御システムの場合はコントローラ）が保有するようにし、サイクルマスタ自体が保有するデータテーブルにタイマ遅延を記憶するなどの一連のタイマ遅延に関する作業をすべてサイクルマスタによって管理するようにしても良い。

【 0 0 7 7 】

この実施の形態1によれば、IEEE 1394の仕様（P1394a既存の仕様）の範囲内で、各ノードが有するサイクルタイマのタイマ遅延を補償することで、IEEE 1394を任意の用途に利用する場合でも、高精度な同期を実現することが可能となる。さらには、IEEE 1394をコントローラとサーボドライバと間のオープンなインタフェースとして使用した場合でも、コントローラとサーボドライバとの間、あるいはサーボドライバ間同士のタイマ遅延を補償することが可能となり、高速高精度な制御を実現することができる。また、タイマ遅延計測に使用されるパケットは、サイクルマスタのみが使用できるものではなく、全てのノードが等しく使用できるため、任意のノード間のタイマ遅延を計測することが可能である。

【 0 0 7 8 】

実施の形態2 .

つぎに、この発明の実施の形態2について説明する。上述した実施の形態1では、RESPONSE_TIMEに、実際どのぐらいの時間がかかったのかが分からないため、およそそのタイマ遅延を推定するようにしていたが、この実施の形態2では、真に正確なタイマ遅延を測定できるようにし、特にタイマ遅延の誤差が大きな影響を及ぼす、コントローラとサーボドライバとを有する制御システムにIEEE 1394を用いる場合であっても、支障のないようにしている。

【 0 0 7 9 】

IEEE 1394では、リード（Read）パケットを使って、応答ノードが有するサイクルタイマのサイクルタイマ値を読み込むことができる。この読み込みによって取得されたサイクルタイマ値と要求ノードが有するサイクルタイマのサイクルタイマ値との差から、タイマ遅延を計測すれば、実際のタイマ遅延を測定することが可能であるように思われる。

【 0 0 8 0 】

しかし、実際には、Readパケットに対するレスポンス（ReadResponse）パケットは、応答ノードの上位層（トランザクション層およびそれより上位の層）がReadResponseパケットを送信バッファ（TX_FIFO）に蓄える際、ReadResponseパケットにサイクルタイマ値を書き込むため、その後、実際に応答ノードのリンク層から送信されるまでの時間は一定値ではなく、Readパケットとこれに対するReadResponseパケットとを用いても、正確なタイマ遅延を測定することはできない。

【 0 0 8 1 】

送信バッファに蓄えられる時間が長くなる理由は、このリード処理が非同期（ASYNC）処理であり、ReadパケットおよびReadResponseパケットが非同期（ASYNC）パケットである

10

20

30

40

50

から、Asynchronous転送時間帯でしか送信することができないために、ReadResponseパケットが送信バッファに蓄える時がAsynchronous転送時間帯であれば、図2に示すように、送信バッファに滞留する時間は一瞬であって、直ちに送信されるのに対し、図3に示すように、ReadResponseパケットが送信バッファに送られた時がIsochronous転送時間帯であれば、Isochronous転送時間帯が終了してAsynchronous転送時間帯になって送信可能となるまで、ReadResponseパケットが送信バッファに滞留するからである。

【0082】

この送信バッファに滞留する時間の差異を解消するには、いくつかの方策がある。その第1の方策は、応答ノードN2がReadパケットを受け付けてReadResponseパケットを送出するまでの一連の動作が、常に図2に示すように、同一のAsynchronous転送時間帯中に行われればよいのであるから、要求ノードN1が予め、Readパケットを送信してReadResponseパケットが返送されるまでの一連の動作が、同一のAsynchronous転送時間帯中に行われるかどうかを判定部1が判定し、同一のAsynchronous転送時間帯中に行われると判断した場合に、要求ノードがReadパケットを応答ノードに送信すればよい。

10

【0083】

基本的に、タイマ遅延計測のためのパケットは、ネットワーク構築後の初期の段階で行われるので、ネットワーク構築後の初期の段階においては、制御システムが制御するためのデータ転送はなく、すなわちトラフィックは少なく、また、応答ノードN2の上位層でも特別なアプリケーションが作動していないので、タイマ遅延計測のためのパケットの要求に対してすばやく応答するであろう。

20

【0084】

したがって、同一のAsynchronous転送時間帯中に上述した一連の動作が行われるネットワーク構築後の初期の段階などに、タイマ遅延計測のためのパケットを要求ノードN1が応答ノードN2に送出すれば、Readパケットを受け付けてからReadResponseパケットを送出するまでの処理時間は、容易に推定することができる。この場合、上位層における処理時間の推定は、上述した実施の形態1による遅延計測を用いることができる。また、この場合、上述した一連の動作が他のトラフィックに邪魔される可能性は少ないため、信頼度の高い測定が可能となる。

【0085】

この第1の方策によれば、既存のプロトコルをそのまま使用し、送信側である要求ノードN1の上位層に、Readパケットを送信してReadResponseパケットが返送されるまでの一連の動作が同一のAsynchronous転送時間帯中に行われるかどうかを判定する判定部1を設けるのみで、この一連の動作が同一のAsynchronous転送時間帯内で行われるので、正確なタイマ遅延補償が可能となる。

30

【0086】

第2の方策としては、図4に示すように、応答ノードN1の上位層が、ReadResponseパケットを送信バッファに蓄えるタイミングをAsynchronous転送時間帯の間に行えば良い。すなわち、応答ノードN2の上位層が、Asynchronous転送時間帯となるまで待って、ReadResponseパケットをリンク層に送出するようにする。なお、上位層は、下位のリンク層からの割り込み信号によって現在の通信伝送路の状態がIsochronous転送時間帯であるかAsynchronous転送時間帯であるかを知ることができる。これによって、Asynchronous転送時間帯待ちの既知の時間を上位層の処理時間に加えるのであるので、要求ノードN1と応答ノードN2との間のタイマ遅延を確実に測定することができる。

40

【0087】

第2の方策によれば、既存のプロトコルをそのまま使用し、受信側ノードである応答ノードN2の上位層に、現在の通信伝送路の状態がIsochronous転送時間帯であるかAsynchronous転送時間帯であるかを判定する判定部を設けるだけで、ReadパケットとReadResponseパケットとが同一のAsynchronous転送時間帯内であろうとなかろうとに関わらず、正確なタイマ遅延補償を行うことができる。

【0088】

50

第3の方策としては、図5に示すように、ReadResponseパケットをリンク層から送出する時に、書込部3がこのReadResponseパケットにサイクルタイム値を書き込むようにする。すなわち、上位層からリンク層にReadResponseパケットが送られたときにサイクルタイム値を書き込むのではなく、ReadResponseパケットがリンク層から送出される瞬間に、このReadResponseパケットにサイクルタイム値を書き込むようにする。

【0089】

IEEE1394において、サイクルスタートパケットによるサイクルタイム設定では、サイクルマスタから送られてくるサイクルスタートパケットを応答ノードN2がリンク層で受け取った瞬間に、該サイクルスタートパケットに記述されているサイクルタイム値を自ノードのサイクルタイム値として設定する機構を既に備えている。

10

【0090】

しかしながら、この機構は、サイクルスタートパケットに固有の機構であって、第3の方策を行う機構は、IEEE1394には定義されておらず、新規にこの機構を追加するものである。なお、第1の方策あるいは第2の方策のように、送信バッファに蓄えられてから直ちに送信されると言っても若干の時間はかかるものであるが、この第3の方策によれば、この若干の時間の誤差すら発生せず、一層、正確な遅延測定が可能になる。

【0091】

第3の方策によれば、既存のプロトコルをそのまま使用し、受信側ノードである応答ノードのリンク層に、ReadResponseパケットがリンク層から送出される瞬間にこのReadResponseパケットにサイクルタイム値を書き込む書込部3を設けるだけで、ReadパケットとReadResponseパケットとが同一のAsynchronous転送時間帯内であろうとなかろうとに関わらず、正確なタイム遅延補償が可能となる。

20

【0092】

この実施の形態2によれば、IEEE1394の仕様（IEEE1394-1995既存の仕様）の範囲内で、各ノードが有するサイクルタイムのタイム遅延を補償することで、IEEE1394を任意の用途に利用する場合でも、高精度な同期を実現することが可能である。さらには、IEEE1394をコントローラとサーボドライバとの間のオープンインタフェースとして使用した場合でも、コントローラとサーボドライバとの間あるいはサーボドライバ間同士のタイム遅延を補償することが可能となり、高速高精度な制御を実現できる。また、タイム遅延計測に使用されるパケットは、サイクルマスタのみが使用できるものではなく、全てのノードが等しく使用できるため、任意のノード間のタイム遅延を計測することが可能である。

30

【0093】

実施の形態3.

つぎに、この発明の実施の形態3について説明する。この実施の形態3では、上述した実施の形態1または実施の形態2によって測定されたタイム遅延をもとに、タイム遅延の補償を行うものである。なお、この実施の形態3では、サイクルスレーブ側がサイクルスタートパケット受信時にタイム遅延を補償する場合について説明する。

【0094】

まず、サイクルスレーブは、サイクルスレーブ自身が有するサイクルタイム値と、サイクルマスタが有するサイクルタイム値との遅延ずれを知っている必要がある。この遅延ずれを取得するためのいくつかの態様としては、たとえば、つぎの9つの遅延ずれ取得態様がある。

40

【0095】

第1の遅延ずれ取得態様は、図6に示すように、サイクルマスタNMが任意のサイクルスレーブNSとの間の遅延ずれを測定し（S11）、この遅延ずれを当該サイクルスレーブNSに伝える（S12）ことによって当該サイクルスレーブNSが遅延ずれを取得する態様である。

【0096】

第2の遅延ずれ取得態様は、図7に示すように、サイクルマスタNMが任意のサイクルス

50

レーブNSとの間の遅延ずれを測定し(S21)、この遅延ずれをサイクルマスタNMが有する遅延記憶手段に記憶し(S22)、任意のサイクルスレーブNSがサイクルマスタNMに対してこの遅延記憶手段をアクセスし、遅延ずれを読み取り(S23)、当該サイクルスレーブNSが遅延ずれを取得する態様である。

【0097】

第3の遅延ずれ取得態様は、図8に示すように、任意のサイクルスレーブNSがサイクルマスタNMとの間の遅延ずれを測定し(S31)、これによって遅延ずれを取得する態様である。

【0098】

この第1～第3の遅延ずれ取得態様によれば、サイクルスレーブNSは、サイクルマスタNMのタイム値に厳密に一致するように補償することができる。

10

【0099】

第4の遅延ずれ取得態様は、図9に示すように、ある特定のサイクルスレーブNS1が、他の任意のサイクルスレーブNS2との間の遅延ずれを測定し(S41)、かつ、サイクルマスタNMとの間の遅延ずれを測定し(S42)、サイクルマスタNMと任意のサイクルスレーブNS2との間の遅延を算出し(S43)、この遅延を任意のサイクルスレーブNS2に伝える(S44)ことによって、当該任意のサイクルスレーブNS2が遅延ずれを取得する態様である。

【0100】

第5の遅延ずれ取得態様は、図10に示すように、ある特定のサイクルスレーブNS1が、他の任意のサイクルスレーブNS2との間の遅延ずれを測定し(S51)、かつ、サイクルマスタNMとの間の遅延ずれを測定し(S52)、サイクルマスタNMと任意のサイクルスレーブNS2との間の遅延を算出し(S53)、この算出した遅延を、特定のサイクルスレーブ自身が有する遅延記憶手段に記憶し(S54)、任意のサイクルスレーブNS2がこの遅延記憶手段をアクセスしてこの遅延を読み取り(S55)、これによって任意のサイクルスレーブNS2が遅延ずれを取得する態様である。

20

【0101】

第6の遅延ずれ取得態様は、図11に示すように、ある特定のサイクルスレーブNS1が、他の任意のサイクルスレーブNS2との間の遅延ずれを測定し(S61)、かつ、サイクルマスタNMとの間の各遅延ずれを測定し(S62)、これらの各遅延を特定のサイクルスレーブNS1自身が有する遅延記憶手段に記憶し(S63)、任意のサイクルスレーブNS2が特定のサイクルスレーブNS1をアクセスしてこれらの遅延を読み取り(S64)、その後サイクルマスタNMとの間の遅延を算出する(S65)態様である。

30

【0102】

上述した第4～第6の遅延ずれ取得態様は、第1～第3の遅延ずれ取得態様と同じ作用効果を有するが、やや複雑な取得態様となっている。

【0103】

第7の遅延ずれ取得態様は、図12に示すように、ある特定のサイクルスレーブNS1が他の任意のサイクルスレーブNS2との間の遅延ずれを測定し(S71)、この遅延ずれを当該任意のサイクルスレーブNS2に伝送し(S72)、任意のサイクルスレーブNS2が遅延ずれを取得する態様である。

40

【0104】

第8の遅延ずれ取得態様は、図13に示すように、ある特定のサイクルスレーブNS1が他の任意のサイクルスレーブNS2との間の遅延ずれを測定し(S81)、この遅延ずれを特定のサイクルスレーブNS1自身が有する遅延記憶手段に記憶し(S82)、任意のサイクルスレーブNS2がこの遅延記憶手段にアクセスして遅延ずれを読み取る(S83)ことによって、任意のサイクルスレーブが遅延ずれを取得する態様である。

【0105】

第9の遅延ずれ取得態様は、図14に示すように、任意のサイクルスレーブNS2が、ある特定のサイクルスレーブNS1との間の遅延ずれを測定し(S91)、これによって任

50

意のサイクルスレーブNS2が遅延ずれを取得する態様である。

【0106】

上述した第7～第9の遅延ずれ取得態様によれば、ある特定のサイクルスレーブのタイマ値に厳密に一致するように他の任意のサイクルスレーブのタイマ値を補償することが可能となる。この場合、同じタイマ値を有していないのはネットワーク上ではサイクルマスタのみとなる。

【0107】

サイクルスレーブが自信の有するサイクルタイマ値と、基準となるノード（サイクルマスタの場合もあればサイクルスレーブの場合もある）が有するサイクルタイマ値の遅延ずれを、サイクルスレーブ自身で知ることができたならば、サイクルスレーブ側自身でタイマ遅延を補償することができる。このタイマ遅延補償態様には、つぎの3つの態様がある。以下に説明する3つの態様では、基準となるノードがサイクルマスタの場合について説明しているが、基準となるノードがサイクルスレーブの場合でも同様に実施できる。

【0108】

第1のタイマ遅延補償態様は、図15に示されるように、タイマ遅延量 t_{11} を考慮してサイクルタイマTM1のサイクルタイマ値を設定するものである。サイクルマスタNMが、サイクルスレーブNSに対し、サイクルスタートパケットによってサイクルマスタNMが有するサイクルタイマ値 t_{10} を伝送した際に、サイクルスレーブNS側がこのサイクルスタートパケットを受け取った時にサイクルタイマ値 t_{10} をそのままサイクルスレーブNS自身が有するサイクルタイマTM1のサイクルタイマ値 t_{10} と設定せずに、遅延ずれ量 t_{11} を考慮してサイクルタイマ値 t_{12} を設定する。すなわち、
 t_{12} （設定すべきサイクルタイマ値）
 $= t_{10}$ （受け取ったサイクルタイマ値）+ t_{11} （タイマ遅延量）
 となる。

【0109】

第2のタイマ遅延補償態様は、図16に示すように、サイクルスレーブNS側がサイクルスタートパケットを受け取った時にサイクルタイマ値 t_{10} をそのまま自サイクルスレーブNSが有するサイクルタイマ値 t_{10} として設定する点は、従来のタイマ設定と同じであるが、サイクルスレーブNS側の上位層（サイクルスレーブ側のアプリケーション）がサイクルタイマ値を使用する、すなわち読みにきた際におけるサイクルタイマ値 t_{13} に、遅延ずれ量を補正し、この補正したサイクルタイマ値 t_{14} を上位層が用いるようにする。この場合、遅延ずれ量が予め知っていれば、基本的にいつでもサイクルタイマ値を補償でき、この補償したサイクルタイマ値を用いることができる。すなわち、
 t_{14} （使用すべきサイクルタイマ値）
 $= t_{13}$ （リンク層のサイクルタイマ値）+ t_{11} （タイマ遅延量）
 となる。

【0110】

第3のタイマ遅延補償態様は、図17に示すように、サイクルスレーブNSは、サイクルタイマTM1とは別個に第2のサイクルタイマTM2を有する。サイクルタイマTM1には、受け付けたサイクルタイマ値をそのまま設定し、第2のサイクルタイマTM2には、任意のタイミング、たとえば時間的にクリティカルでないときに、タイマ遅延量 t_{11} を加味したタイマ遅延補償されたサイクルタイマ値が設定され、上位層は、このサイクルタイマ値を用いるようにしている。

【0111】

この実施の形態3によれば、IEEE1394の仕様（IEEE1394-1995既存の仕様）の範囲内で、各ノードが有するサイクルタイマのタイマ遅延を補償することで、IEEE1394を任意の用途に利用する場合でも、高精度な同期を実現することが可能である。また、すべてのノードがタイマ補償を必要としている訳ではないので、必要なノードだけが必要に応じて遅延計測補償を行えば良い。さらには、IEEE1394をコントローラとサーボドライバとの間のオープンなインタフェースとして使用した場合でも、

10

20

30

40

50

コントローラとサーボドライブとの間、あるいはサーボドライブ間同士のタイマ遅延を補償することが可能となり、高速高精度な制御を実現できる。また、タイマ遅延補償はサーボ側で行うので、コントローラ部分に何の変更も必要ない。さらには、すべての被制御機器がタイマ補償を必要としている訳ではないので、必要なサーボノードだけが必要に応じて遅延計測補償を行えば良い。

【 0 1 1 2 】

ところで、上述した第1のタイマ遅延補償態様は、単純であり、また、タイマを用いた割込は遅延を補償した時刻における割込となることから、正確である。しかし、IEEE 1394の基幹に関わる部分であるリンク層を変更しなければならない。さらに、IEEE 1394ではサイクルスタートパケットを受け取ってタイマ値を更新する際にタイマ値を戻してはならない、という規約がある。この実施の形態で、サイクルスレーブのサイクルタイマを基準としたならば、遅延ずれが負の値を持ち、タイマ遅延補償をする際にタイマ値を戻すノードが存在することがあるので、このような場合には、IEEE 1394のこの規約を無視するか、あるいは、「基準となるサイクルスレーブのサイクルタイマ値＋一定時間」をネットワーク上での基準時間とすることで、タイマ遅延補償をする際にタイマ値を戻すノードがないようにして、IEEE 1394のこの規約を守るようにする、などの処置を行う必要がある。また、タイマ遅延補償を行うサイクルスレーブの上位層からみて、タイマがいつ更新されるかがわからない、すなわちサイクルスタートパケットがいつ送られてくるのかわからないため、タイマ遅延補償を行うサイクルスレーブの上位層において、時間を計測している最中など、時間に対してクリティカルな最中にタイマの更新が行われる可能性があり、クリティカルな最中にタイマの更新が行われた場合とそうでない場合とで誤差が生じてしまう。

【 0 1 1 3 】

もともと、サイクルマスタが有するサイクルタイマとサイクルスレーブが有するサイクルタイマとは完全に特性が一致するとは限らないので、お互いのサイクルタイマの間に精度の差が存在し、サイクルスタートパケットによってサイクルスレーブのサイクルタイマを、サイクルマスタのサイクルタイマ値に合わせるのは、このサイクルタイマ間の精度差をなくするためである。しかしながら、たとえば、タイマ遅延補償を行うサイクルスレーブの上位層（の任意のアプリケーション）が、何か別の目的で時間計測などを、自身のサイクルタイマを使って行う場合、サイクルタイマの更新が起これなければ、自身のサイクルタイマの精度の範囲内で、その時間計測の精度は保証されるが、その時間計測の途中でサイクルタイマの更新が起こってしまったら、その時間計測の精度は全く保証されないものになってしまう。もちろん、この誤差が問題のない範囲でシステムにIEEE 1394を用いるのであれば、この第1のタイマ遅延補償態様はこのシステムに十分である。

【 0 1 1 4 】

一方、第2のタイマ遅延補償態様では、IEEE 1394に関わる変更は全くなく、上位層（アプリケーション）のみ変更すれば良いので、容易に実現が可能である。しかしながら、このタイマを使った割込は、遅延を補償したタイマが発生する割込とはならない。この代替手段としてアプリケーションからタイマをポーリングして正確な時刻の割込を実現することも可能である。ただし、ポーリングによる負荷が生じることになる。さらに、第1のタイマ遅延補償態様と同様に、サイクルタイマ値の更新がいつ起こるか判らないことによる、タイマの精度の保証の問題が残る。ポーリングによる負荷や、タイマ更新に関わる誤差が問題ない範囲で用いるシステムに適用するのであれば、この第2のタイマ遅延補償態様は、十分用いることができる。

【 0 1 1 5 】

また、第3のタイマ遅延補償態様では、第1および第2のタイマ遅延補償態様で問題となっていたタイマ更新に関わる誤差の保証問題を解決することができる。この第3のタイマ遅延保証態様では、サイクルタイマ値にタイマ遅延量を考慮したサイクルタイマ値を、第2のサイクルタイマに第2のサイクルタイマ値として設定するタイミングを任意とすることができるため、タイマ遅延量 t を計測している最中など、時間にクリティカルな時間

帯を避けたタイミングでタイマ値を設定することができる。また、このタイマを使った割込は、遅延を補償した時刻における割込であるから、正確である。したがって、この第3のタイマ遅延補償態様によるタイマ遅延補償が最も好ましいといえる。

【0116】

実施の形態4

つぎに、この発明の実施の形態4について説明する。この実施の形態4では、上述した実施の形態1, 2に示したタイマ遅延の測定などを用いてタイマ遅延を測定しておき、このタイマ遅延およびタイムスタンプを用いてタイマ遅延補償を行うようにしている。

【0117】

図18は、タイムスタンプを用いて制御を行う同期システムの概要を示す図である。図18において、要求ノードN1と応答ノードN2とによる同期をとる場合、たとえばコントローラと各サーボドライバとによる同期制御を行う場合、ある時刻において同期を取る、すなわち同期制御を行う必要があるが、このような場合には、コマンド内容（制御内容）とともに、このコマンド内容（制御内容）が行われるべき時刻をタイムスタンプとして、要求ノードN1から応答ノードN2に対して送られる。要求ノードN1がコントローラであり、応答ノードN2が各サーボドライバである場合には、コントローラから各サーボドライバに対してタイムスタンプが送られる。応答ノードN2は、受け取った時刻、すなわちタイムスタンプの時刻（タイムスタンプ値） t_{20} になると、受け取ったコマンド内容（制御内容）を実行する。

【0118】

しかしながら、応答ノードN2の時刻は、応答ノードN2自身が有するサイクルタイマによるものであるから、応答ノードN2と要求ノードN1との間に遅延が存在する。応答ノードN2が要求ノードN1からタイムスタンプ値を受け付けると、応答ノードN2は、遅延ずれを補償する。なお、この場合、各応答ノードN2は、各応答ノードN2が有するサイクルタイマ値と、要求ノードN1が有するサイクルタイマ値との遅延ずれを予め知っている必要がある。この遅延ずれを、タイムスタンプを用いて補償する態様には、つぎの2つの態様がある。

【0119】

第1のタイムスタンプによるタイマ遅延補償態様は、まず要求ノードN1が、任意の応答ノードN2との間の遅延ずれを測定し、この遅延を任意の応答ノードN2に伝達する。応答ノードN2が遅延ずれ量を予め取得する態様には、つぎの3つの態様がある。

【0120】

第1の取得態様は、図6に示した第1の遅延ずれ取得態様と同様に、要求ノードN1が任意の応答ノードN2との間の遅延ずれを測定し、この遅延ずれを当該任意の応答ノードN2に伝える態様である。

【0121】

第2の取得態様は、図7に示した第2の遅延ずれ取得態様と同様に、要求ノードN1が任意の応答ノードN2との間の遅延ずれを測定し、要求ノードN1自身が有する遅延記憶手段に記憶し、任意の応答ノードN2が要求ノードN1の遅延記憶手段にアクセスし、この遅延ずれを読み取り、当該任意の応答ノードN2が遅延ずれを取得する態様である。

【0122】

第3の取得態様は、図8に示した第3の遅延ずれ取得態様と同様に、任意の応答ノードN2が要求ノードN1との間の遅延ずれを測定し、これによって、遅延ずれを取得する態様である。

【0123】

その後、図19に示すように、要求ノードN1から応答ノードN2に対して、コマンド内容（制御内容）とともに、このコマンド内容（制御内容）が行われるべき時刻であるタイムスタンプ値 t_{20} が応答ノードに送られてくると、応答ノードN2は、受け取った時刻、すなわちタイムスタンプ値 t_{20} に対して遅延ずれ量 T_{21} を補償し、この補償したタイムスタンプ値 t_{22} が示す時刻になると、受け取ったコマンド内容（制御内容）を実

行する。

【0124】

一方、第2のタイムスタンプによるタイム遅延補償態様では、要求ノードN1が各応答ノードN2の遅延ずれ量 t_{21} をそれぞれ考慮したタイムスタンプを発行する。この場合は、要求ノードN1は、要求ノードN1自身が有するサイクルタイム値と各応答ノードが有するサイクルタイム値との遅延ずれ量 t_{21} を予め知っている必要がある。この遅延ずれ量を取得する態様としては、つぎの3つの態様がある。

【0125】

その第1の取得態様は、要求ノードN1が各応答ノードN2との間の遅延を測定し、要求ノードN1自身が有するデータテーブル形式の遅延量記憶手段に記憶しておく。

10

【0126】

第2の取得態様は、各応答ノードN2が、要求ノードN1との間の遅延ずれ量 t_{21} を測定し、この遅延ずれ量 t_{21} を応答ノードN2自身の遅延量記憶手段に記憶しておき、要求ノードN1が、応答ノードN2の遅延量記憶手段に記憶された遅延ずれ量を読み取る。

【0127】

第3の取得態様は、各応答ノードN2が、要求ノードN1との間の遅延ずれ量を測定し、この遅延ずれ量を要求ノードN1に伝える。

【0128】

その後、図20に示すように、要求ノードN1は、制御内容とともに、この制御内容が行われるべき時刻であるタイムスタンプ値 t_{20} を、各応答ノードN2の遅延ずれ量を考慮して、遅延ずれ量 t_{21} 分だけ時刻を戻して設定し、要求ノードN1から各応答ノードN2に対して送る。これに対し、応答ノードN2側は、受け取ったタイムスタンプ値 t_{22} が示す時刻になると、受け取った制御内容を実行する。この場合、受け取ったタイムスタンプ値 t_{22} は、既に遅延ずれ量を考慮した値となっているので、そのまま制御内容を実行すればよい。

20

【0129】

この実施の形態4によれば、IEEE1394の仕様(IEEE1394-1995既存の仕様)の範囲内で、各ノードが有するサイクルタイムのタイム遅延を補償することで、IEEE1394を任意の用途に利用する場合でも、高精度な同期を実現することが可能である。また、すべてのノードがタイム補償を必要としている訳ではないので、必要なノードだけが必要に応じて遅延計測補償を行えば良い。さらには、IEEE1394をコントローラとサーボドライバとの間のオープンなインタフェースとして使用した場合でも、コントローラとサーボドライブとの間、あるいはサーボドライブ間同士のタイム遅延を補償することが可能となり、高速高精度な制御を実現できる。また、タイム遅延補償はサーボ側で行うので、コントローラ部分に何の変更も必要ない。さらには、すべての被制御機器がタイム補償を必要としている訳ではないので、必要なサーボノードだけが必要に応じて遅延計測補償を行えば良い。

30

【0130】

実施の形態5

40

つぎに、この発明の実施の形態5について説明する。この実施の形態5では、上述した実施の形態1~4で示したIEEE1394を、コントローラとサーボドライバとの間のオープンなインタフェースとして使用した場合について説明する。

【0131】

ここで、コントローラをサイクルスレーブとし、被制御対象機器であるサーボドライバをサイクルマスターとし、被制御対象機器のサーボドライバの1つを、サイクルマスターとしてもよいが、コントローラをサイクルマスターとし、被制御対象機器のサーボドライバの全てをサイクルスレーブとすることが好ましい。

【0132】

コントローラは、サーボドライバに対してIsochronousパケットによるサーボデータなど

50

のデータを送出し、サーボドライバは、コントローラに対してIsochronousパケットによるフィードバックデータなどを返送する。この場合、コントローラがサイクルマスタであれば、Isochronous転送時間帯の中で、コントローラの制御パケットを先に優先し、すなわちIsochronousパケットの先頭に位置させて送出することができる。

【0133】

上述した実施の形態1～4を適宜組み合わせ、コントローラとサーボドライバとのタイム遅延計測補償を行ってもよいが、ここでは、タイム遅延の計測に関する一連の動作をコントローラが管理し、タイム遅延の補償をサーボドライバ側が行う場合のタイム遅延補償処理シーケンスについて説明する。

【0134】

図21は、タイム遅延の補償をサーボドライバ側が行う場合におけるタイム遅延補償処理手順を示すフローチャートである。図21において、まず、サーボドライバをコントローラに接続する(ステップS101)。その後、コントローラが有するネットワーク構成変化判別手段が、ネットワークの構成が変化したことを検出し、ネットワーク再構成後に1回だけ、コントローラが各サーボドライバとの間のタイム遅延を計測し、このタイム遅延量を、コントローラ自身が有するデータテーブルに記憶する(ステップS102)。

【0135】

その後、コントローラは、各サーボドライバにタイム遅延量を伝達し(ステップS103)、このタイム遅延量を受けたサーボドライバは、このタイム遅延量をもとにタイム遅延を補償し(ステップS104)、本処理を終了する。

【0136】

つぎに、コントローラが遅延補償したタイムスタンプを生成する場合のタイム遅延補償処理シーケンスについて説明する。図22は、コントローラが遅延補償したタイムスタンプを生成する場合の遅延補償処理手順を示すフローチャートである。図22において、まず、ステップS101と同様に、サーボドライバをコントローラに接続する(ステップS201)。その後、コントローラが有するネットワーク構成変化判別手段が、ネットワークの構成が変化したことを検出し、ネットワーク再構成後に1回だけ、コントローラが各サーボドライバとの間のタイム遅延を計測し、このタイム遅延量を、コントローラ自身が有するデータテーブルに記憶する(ステップS202)。その後、コントローラは、タイム遅延量を補償したタイムスタンプを生成してサーボドライブ側に伝達し(ステップS203)、本処理を終了する。

【0137】

この実施の形態5では、コントローラとサーボドライブとからなる制御システムのように、高速高精度の同期を必要とするシステムであっても、コントローラとサーボドライブとの間にオープンなインタフェースであるIEEE1394を使用して、コントローラとサーボドライブとの間のタイム遅延補償を確実に精度良く行うことができる。

【0138】

【発明の効果】

以上説明したように、この発明によれば、IEEE1394を用いて接続されたネットワーク内の各ノード間の同期をとる際、サイクルタイム遅延計測手段が、前記各ノードが有する各サイクルタイム同士の遅延ずれ量を計測し、サイクルタイム遅延補正手段が、前記サイクルタイム遅延計測手段が計測した遅延ずれ量をもとに前記各サイクルタイム同士の遅延ずれを補正するようにしているので、IEEE1394の仕様の範囲内で、各ノードが有するサイクルタイムのサイクルタイム同士のタイム遅延が補償され、高速高精度な同期制御が可能なシステムを実現することができるという効果を奏する。また、前記サイクルタイム遅延計測手段が、要求ノードから応答ノードに対してReadパケットを送信し、該応答ノードのサイクルタイム値を読み取り、この読み取ったサイクルタイム値を含むReadResponseパケットを前記要求ノードが受信し、この受信したサイクルタイム値と要求ノードのサイクルタイム値との差をもとに前記遅延ずれ量を算出するようにしているので、サイクルタイム値を読み取ることによって容易にサイクルタイム同士の遅延ずれ量を得る

10

20

30

40

50

ことができるという効果を奏する。

【 0 1 4 4 】

つぎの発明によれば、前記要求ノードの判定手段が、前記Readパケットを送信し、前記ReadResponseパケットが返送される一連のシーケンスが、同一のアシクロナス転送が行われている時間帯である同一のAsynchronous転送時間帯で完了するか否かを判定し、前記要求ノードが、前記判定手段が同一のAsynchronous転送時間帯で完了すると判定した場合に、Readパケットを前記応答ノードに送出するようにし、ReadResponseパケットが同一のAsynchronous転送時間帯内で返送されるようにしているので、信頼度の高い遅延ずれ量を計測することができるという効果を奏する。

【 0 1 4 5 】

つぎの発明によれば、前記応答ノードの上位層の判定手段が、アシクロナス転送が行われている時間帯であるAsynchronous転送時間帯であるか否かを判定し、前記判定手段がAsynchronous転送時間帯であると判定した場合に、前記上位層は前記ReadResponseパケットを送信バッファ(TX_FIFO)に蓄えることで、前記ReadResponseパケットは、送信バッファ長くとどまることなく、直ちに要求ノードに返送されるようにしているので、信頼度の高い遅延ずれ量を計測することができるという効果を奏する。

【 0 1 4 6 】

つぎの発明によれば、前記応答ノードのリンク層の書込手段が、前記ReadResponseパケットがリンク層バッファから送出される直前に該ReadResponseパケットに当該応答ノードのサイクルタイム値を書き込むようにしているので、信頼度の高い遅延ずれ量を計測することができるという効果を奏する。

【 0 1 4 7 】

つぎの発明によれば、各ノードのサイクルタイム遅延記憶手段が、前記遅延ずれ量を記憶し、少なくとも他のノードからのアクセスによって遅延ずれ量を獲得できるようにしているので、効率的な遅延ずれ量の取得を可能にするという効果を奏する。

【 0 1 4 8 】

つぎの発明によれば、前記サイクルタイム遅延記憶手段が、全てのノード間の遅延ずれ量をデータテーブル形式で記憶するようにしているので、遅延ずれ量の取得を迅速に行うことができるという効果を奏する。

【 0 1 4 9 】

つぎの発明によれば、前記サイクルタイム遅延記憶手段を、サイクルマスタとしてのノードに設けるようにしているので、効率的な遅延ずれ量の管理を行うことができるという効果を奏する。

【 0 1 5 0 】

つぎの発明によれば、ネットワーク構成変化判別手段が、前記ネットワークの構成の変化を判別し、前記ネットワーク構成変化判別手段がネットワークの構成に変化があったと判別した場合、該ネットワークの再構成後、前記サイクルタイム遅延計測手段がサイクルタイム同士の遅延ずれ量を計測し、前記サイクルタイム遅延補正手段が該遅延ずれ量をもとにサイクルタイム同士の遅延ずれを補正する処理を一回行うようにしているので、ネットワークのトラヒックに影響を与えず、しかも正確かつ確実な遅延ずれ量の計測および補正を行うことができるという効果を奏する。

【 0 1 5 1 】

つぎの発明によれば、前記ネットワーク構成変化判別手段を、サイクルマスタとしてのノードに設けるようにしているので、迅速かつ効率的な遅延ずれの計測および補正を行うことができるという効果を奏する。

【 0 1 5 2 】

つぎの発明によれば、前記サイクルタイム遅延計測手段および前記サイクルタイム遅延補正手段を、ノード間の同期を用いる所定のノードに設けるようにしているので、ノード間の同期を必要とするノードが確実に遅延ずれ量を計測し、補正することができるという効果を奏する。

10

20

30

40

50

【 0 1 5 3 】

つぎの発明によれば、同期の基準となるサイクルタイムを有するサイクルマスタのノードが、該ノードが計測した遅延ずれ量を前記サイクルタイム遅延伝送手段によって前記所定のノードに伝送し、あるいは該ノードが計測した遅延ずれ量をデータテーブルに記憶し、前記所定のノードが前記サイクルマスタのノードから該遅延ずれ量を取得し、その後所定のノードに伝送された遅延ずれ量あるいは取得した遅延ずれ量をもとに該ノードのサイクルタイムの遅延ずれを補正するようにしているので、遅延ずれ量およびその補正を確実に行うことができるという効果を奏する。

【 0 1 5 4 】

つぎの発明によれば、前記サイクルタイム遅延補正手段が、サイクルスタートパケットを受信した際に、遅延ずれ量をもとにサイクルタイム値を設定するようにしているので、同期がとれた制御処理を確実に行うことができるという効果を奏する。

【 0 1 5 5 】

つぎの発明によれば、前記サイクルタイム遅延補正手段が、上位層がサイクルタイム値の使用時に、前記遅延ずれ量を用いて遅延ずれを補正するようにしているので、適時に、遅延ずれ量が補正された時刻を用いて制御処理を行うことができ、遅延ずれ量の補正時期の不一致による不具合を解消することができるという効果を奏する。

【 0 1 5 6 】

つぎの発明によれば、所定のノードの前記サイクルタイム遅延補正手段が、任意のタイミングで遅延ずれ量を用いて前記遅延ずれを補正したサイクルタイム値を第2のサイクルタイムに設定するようにしているので、適時に、遅延ずれ量が補正された時刻を用いて制御処理を行うことができ、遅延ずれ量の補正時期の不一致による不具合を解消することができるという効果を奏する。

【 0 1 5 7 】

つぎの発明によれば、前記サイクルタイム遅延補正手段が、前記遅延ずれ量をもとに、前記サイクルマスタのノードから送られた前記タイムスタンプのタイムスタンプ値を更新し、前記タイムスタンプ時刻監視手段が、前記更新されたタイムスタンプ値が示す時刻を監視するようにしているので、タイムスタンプを用いた遅延ずれ量の補正を確実に行うことができるという効果を奏する。

【 0 1 5 8 】

つぎの発明によれば、前記サイクルタイム遅延計測手段および前記サイクルタイム遅延補正手段を、サイクルマスタとしてのノードに設けるようにしているので、効率的なサイクルタイム同士の遅延ずれ量の計測および補正を行うことができるという効果を奏する。

【 0 1 5 9 】

つぎの発明によれば、前記所定のノードが、該所定のノードで計測した遅延ずれ量を前記サイクルマスタのノードに伝送し、あるいは前記所定のノードから遅延ずれ量を取得し、その後該所定のノードに伝送された遅延ずれ量あるいは取得した遅延ずれ量をもとに前記サイクルマスタのノードのサイクルマスタの遅延ずれを補正するようにしているので、遅延ずれ量およびその補正を確実に行うことができるという効果を奏する。

【 0 1 6 0 】

つぎの発明によれば、前記サイクルタイム遅延補正手段が、前記遅延ずれ量をもとに、自サイクルマスタのノードの前記タイムスタンプ生成手段が生成したタイムスタンプのタイムスタンプ値を更新し、この更新したタイムスタンプ値を含むデータを所定のノードに伝送するようにしているので、タイムスタンプを用いた遅延ずれ量の計測および補正を効率的かつ確実に行うことができるという効果を奏する。

【 0 1 6 1 】

つぎの発明によれば、請求項1～23のいずれか一つに記載したIEEE 1394を用いた同期システムを、サーボドライブ装置とこれを制御するコントローラとからなる制御システムに適用しているため、高速高精度の同期制御を確実に行うことができるという効果を奏する。

10

20

30

40

50

【 0 1 6 2 】

つぎの発明によれば、前記コントローラを、サイクルマスタとし、効率的な遅延ずれ量の計測および補正ができるようにしているので、高速高精度の同期制御が可能な制御システムを実現することができるという効果を奏する。

【図面の簡単な説明】

【図 1】 この発明の実施の形態 1 である I E E E 1 3 9 4 を用いた同期システムにおける伝送遅延測定処理シーケンスを示す図である。

【図 2】 I E E E 1 3 9 4 を用いた同期システムにおけるタイマ遅延補償の処理シーケンスの概要を示す図である。

【図 3】 この発明の実施の形態 2 である I E E E 1 3 9 4 を用いた同期システムにおけるタイマ遅延補償の処理シーケンスの一例を示す図である。

10

【図 4】 この発明の実施の形態 2 である I E E E 1 3 9 4 を用いた同期システムにおけるタイマ遅延補償の処理シーケンスの一例を示す図である。

【図 5】 この発明の実施の形態 2 である I E E E 1 3 9 4 を用いた同期システムにおけるタイマ遅延補償の処理シーケンスの一例を示す図である。

【図 6】 遅延ずれの取得処理シーケンスの一例を示す図である。

【図 7】 遅延ずれの取得処理シーケンスの一例を示す図である。

【図 8】 遅延ずれの取得処理シーケンスの一例を示す図である。

【図 9】 遅延ずれの取得処理シーケンスの一例を示す図である。

【図 10】 遅延ずれの取得処理シーケンスの一例を示す図である。

20

【図 11】 遅延ずれの取得処理シーケンスの一例を示す図である。

【図 12】 遅延ずれの取得処理シーケンスの一例を示す図である。

【図 13】 遅延ずれの取得処理シーケンスの一例を示す図である。

【図 14】 遅延ずれの取得処理シーケンスの一例を示す図である。

【図 15】 この発明の実施の形態 3 である I E E E 1 3 9 4 を用いた同期システムにおけるタイマ遅延補償の処理シーケンスの一例を示す図である。

【図 16】 この発明の実施の形態 3 である I E E E 1 3 9 4 を用いた同期システムにおけるタイマ遅延補償の処理シーケンスの一例を示す図である。

【図 17】 この発明の実施の形態 3 である I E E E 1 3 9 4 を用いた同期システムにおけるタイマ遅延補償の処理シーケンスの一例を示す図である。

30

【図 18】 この発明の実施の形態 4 である I E E E 1 3 9 4 を用いた同期システムにおけるタイマ遅延補償の処理シーケンスの一例を示す図である。

【図 19】 この発明の実施の形態 4 である I E E E 1 3 9 4 を用いた同期システムにおけるタイマ遅延補償の処理シーケンスの一例を示す図である。

【図 20】 この発明の実施の形態 4 である I E E E 1 3 9 4 を用いた同期システムにおけるタイマ遅延補償の処理シーケンスの一例を示す図である。

【図 21】 この発明の実施の形態 5 である I E E E 1 3 9 4 を用いた同期システムを、コントローラと 1 以上のサーボドライブとからなる制御システムに適用した場合における遅延ずれ補正処理手順の一例を示すフローチャートである。

【図 22】 この発明の実施の形態 5 である I E E E 1 3 9 4 を用いた同期システムを、コントローラと 1 以上のサーボドライブとからなる制御システムに適用した場合における遅延ずれ補正処理手順の一例を示すフローチャートである。

40

【図 23】 I E E E 1 3 9 4 の通信レイヤの構成を示す図である。

【図 24】 Isochronous 通信と Asynchronous 通信とが混在する場合を示す図である。

【図 25】 サイクルスタートパケットを用いたサイクルマスタとサイクルスレーブとの間におけるタイマ同期処理を示すシーケンス図である。

【図 26】 サイクルスタートパケットを用いたサイクルスレーブ間のタイマ同期処理を示すシーケンス図である。

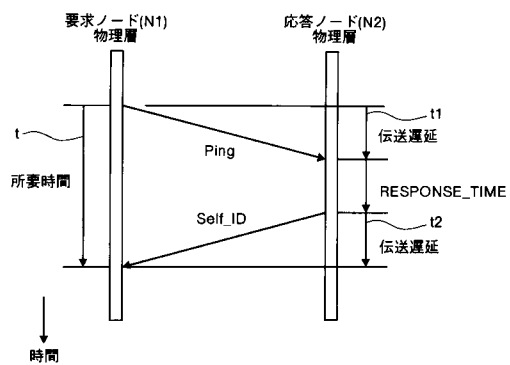
【符号の説明】

1 , 2 判定部、 3 書込部、 N 1 要求ノード、 N 2 応答ノード、 N M サイクルマ

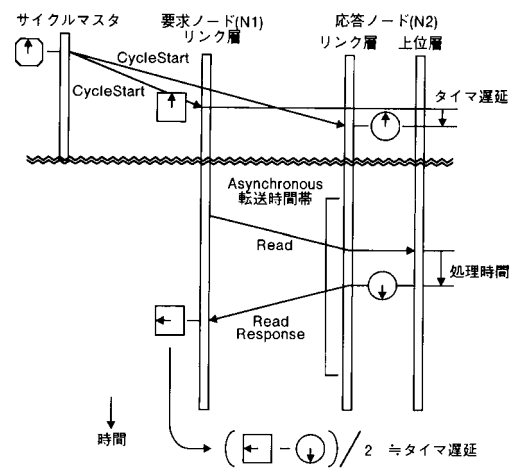
50

スタ、NS, NS1, NS2 サイクルスレーブ、TM2 第2のサイクルタイマ。

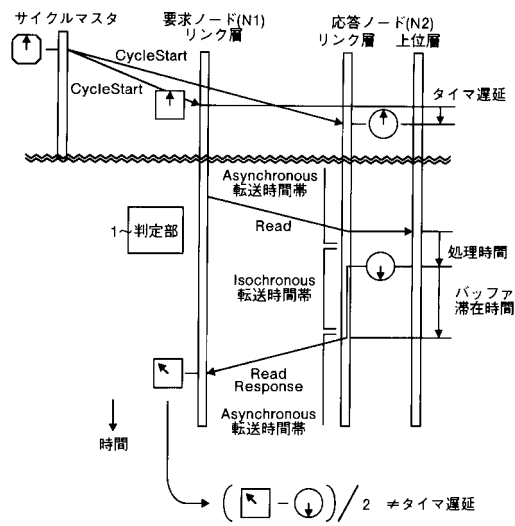
【図1】



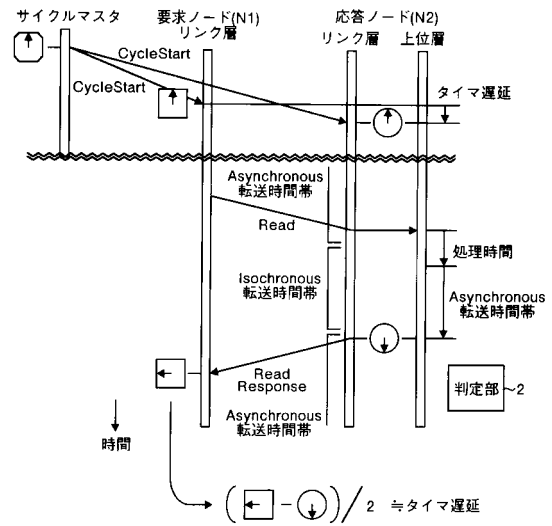
【図2】



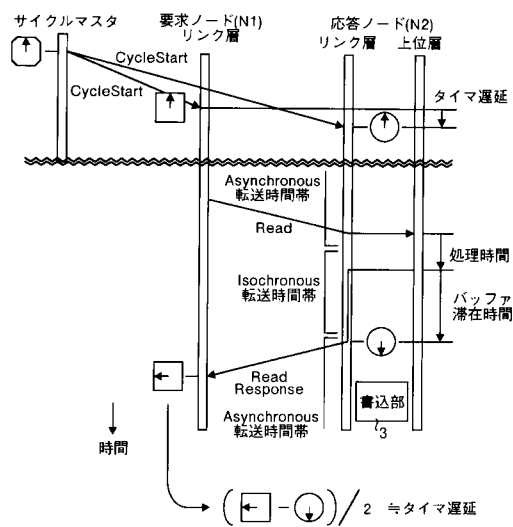
【図 3】



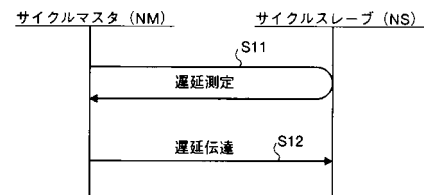
【図 4】



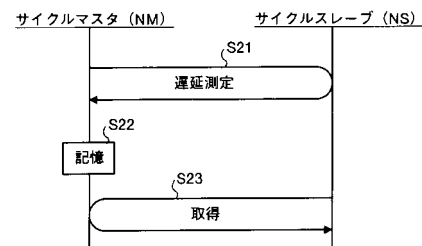
【図 5】



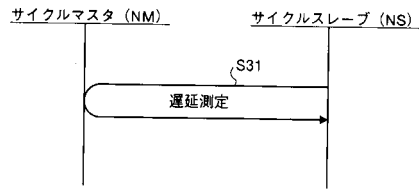
【図 6】



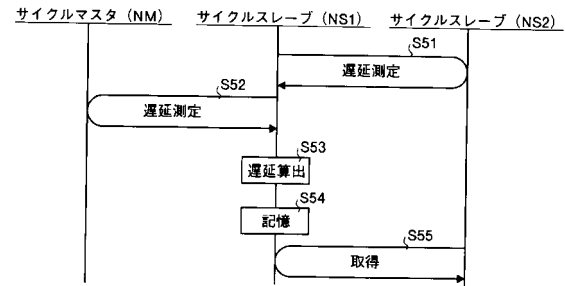
【図 7】



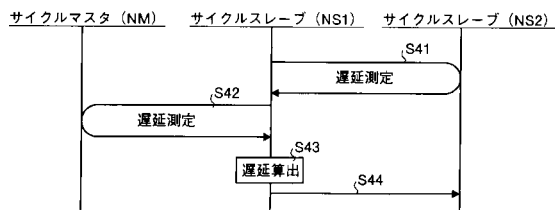
【図 8】



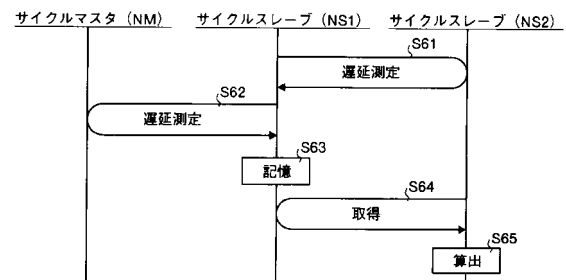
【図 10】



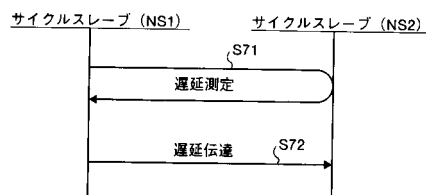
【図 9】



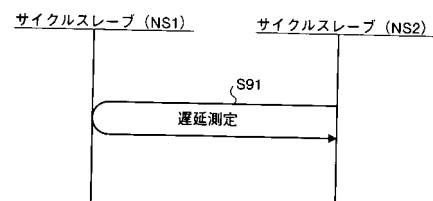
【図 11】



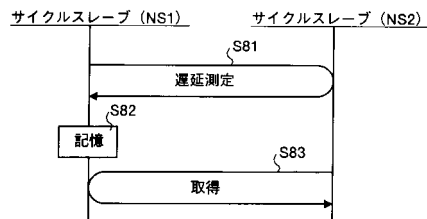
【図 12】



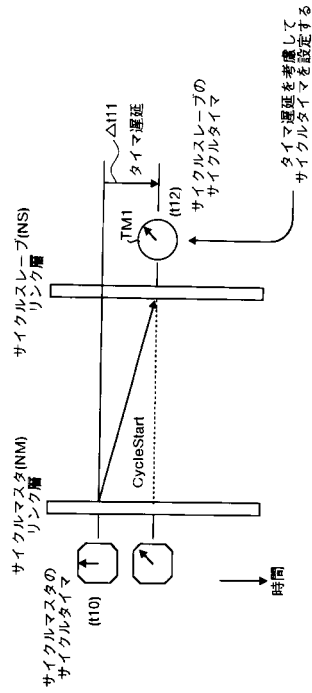
【図 14】



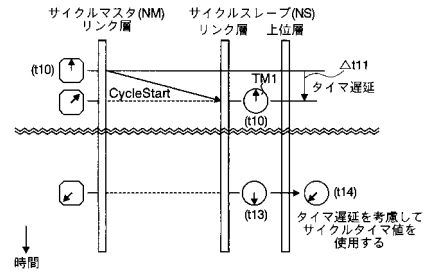
【図 13】



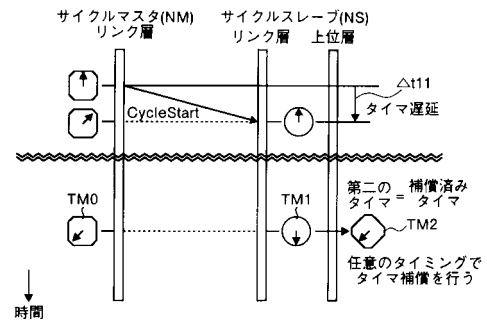
【図 15】



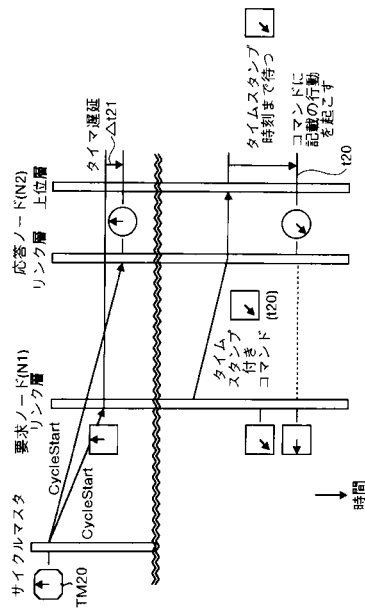
【図 16】



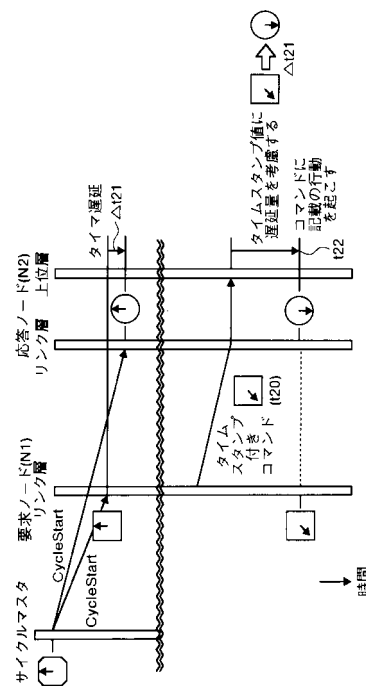
【図 17】



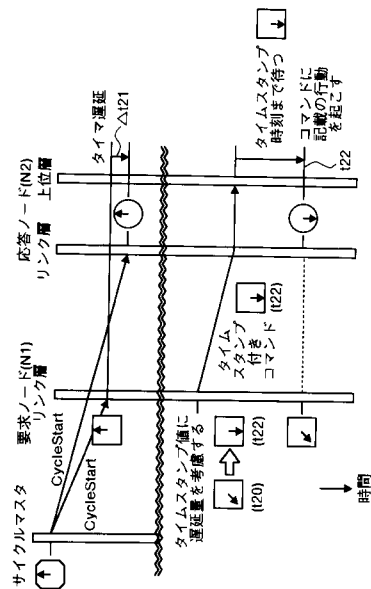
【図 18】



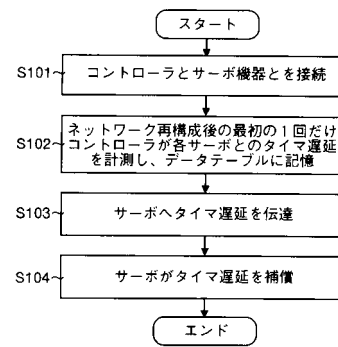
【図 19】



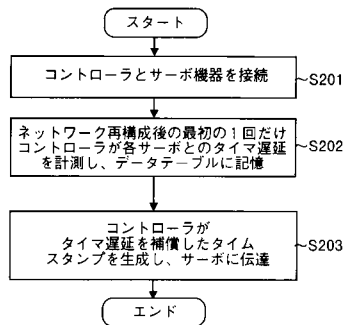
【図 20】



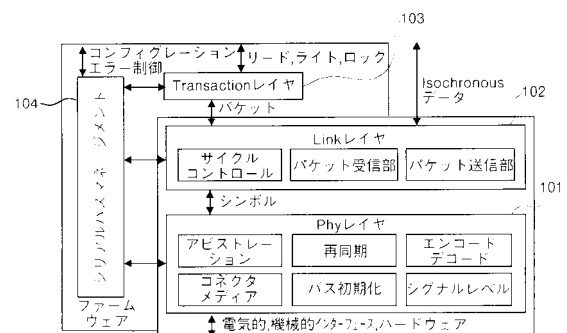
【図 21】



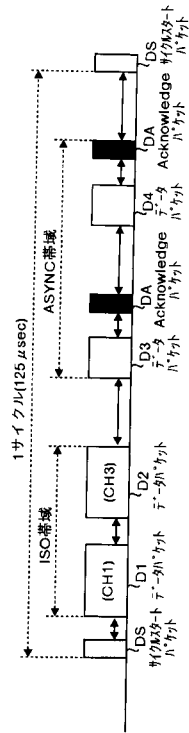
【図 22】



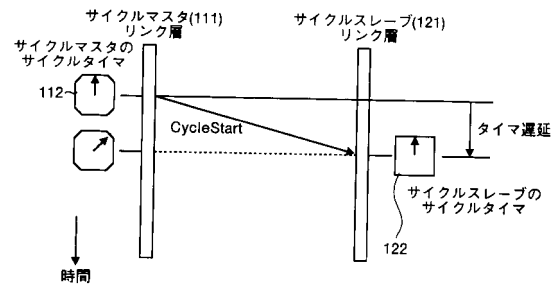
【図 23】



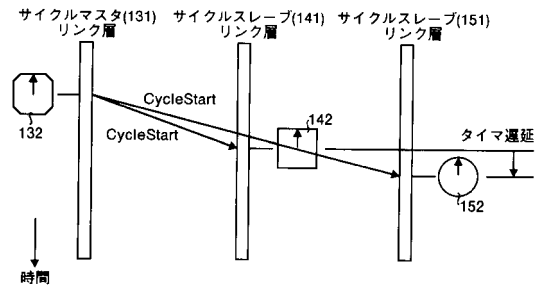
【図 24】



【図 25】



【図 26】



フロントページの続き

(72)発明者 牛尾 裕介
東京都千代田区丸の内二丁目2番3号 三菱電機株式会社内

審査官 羽岡 さやか

(56)参考文献 特開2000-078170(JP,A)
特表平09-502851(JP,A)
特開平09-046390(JP,A)
特開平09-233098(JP,A)

(58)調査した分野(Int.Cl., DB名)
H04L 12/28-12/46
H04L 7/00