1J-04

論理時間を用いた
組み込みシステム向け時間駆動分散処理環境

島袋 健司† 横山 孝典† 兪 明連†
東京都市大学‡

1. はじめに

近年，実世界のセンシングや制御を行なうサイバーフィジカルシステムが注目されている．例えば，自動車における自動運転システムの実用化が近いと言われ，さらに，自動車単体だけでなく，車々間通信や路車間通信を行い，分散した組み込み機器同士が協調動作するシステムも増えつつある．ゆえに，無線ネットワークを含むサイバーフィジカルシステム向けの分散処理環境が求められている．

サイバーフィジカルシステムは実時間に基づく処理を行うため，ジッタの少ない処理が要求される．ジッタの少ない分散処理環境に時間駆動(Time-Triggered)アーキテクチャがある[1]．時間駆動アーキテクチャは TDMA(Time Division Multiple Access)に基づく通信とともに，全ノードに時刻同期機能を持つ時間駆動ネットワークを必要とし，無線ネットワークのような通信時間が変動するネットワークには対応していない．

そこで我々は，論理時間を用いることで通信時間が変動するネットワークでも時間駆動アーキテクチャと同等の動作が可能な，論理時間を用いた時間駆動アーキテクチャを提案し，そのための分散処理環境を開発した[2]．しかしながらこれまでは，複数のメッセージを受信する算出タスクにおいてデータの時間的不整合が発生する可能性があった．

本論文では，複数メッセージ受信における時間的不整合を防ぐことのできる分散処理環境を提案する．

2. 分散処理環境の動作

2.1 時間駆動アーキテクチャ

時間駆動ネットワークを用いた時間駆動アーキテクチャの動作を図 1 を用いて説明する．全タスクは周期 10 で起動される．入力タスクはセンサ情報を送信する．算出タスクは受信したメッセージを基に計算し結果を送信する．出力タスクは受信データをアクチュエータに出力する．全ノードが同期して通信を行うため，メッセージの衝突が発生せず，ジッタの少ない処理が可能である．

A Time-Triggered Distributed Computing Environment Based on Logical Time for Embedded Control Systems
†Kenji Shimabukuro, Takanori Yokoyama and Myungryun Yoo, ‡Tokyo City University

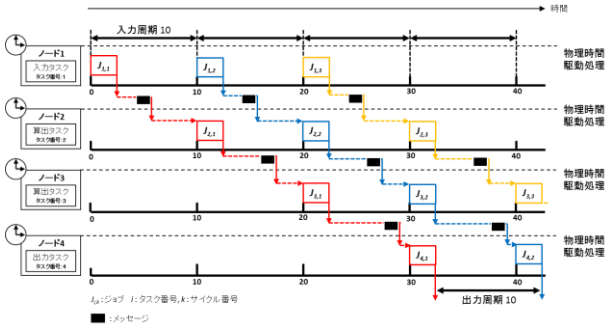

図1: 時間駆動アーキテクチャの動作例

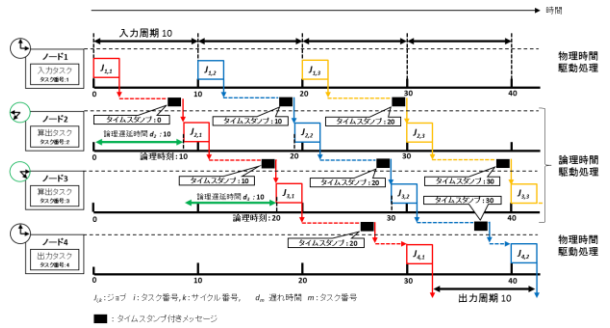

図2: 本分散処理環境の動作例

2.2 論理時間を用いた時間駆動アーキテクチャ

論理時間を用いた時間駆動分散処理環境の動作を図 2 を用いて説明する．入力タスクと出力タスクは実時間に基づいて時間駆動で起動される．一方算出タスクはメッセージ受信でイベント駆動され，論理時間で管理される．メッセージにはタスクの起動時刻を表すタイムスタンプが付加される．この例では，入力タスクは周期 10 で起動され，センサ情報を送信する．算出タスクはメッセージを受信すると処理結果にタイムスタンプとして論理起動時刻を付加して送信する．論理起動時刻は物理時間によらず，図 1 の時間駆動アーキテクチャの起動時刻と同じ値である．出力タスクは物理時間に従い周期 10 で起動され，データをアクチュエータに送信する．

以上のように，入出力タスクは物理時間駆動処理をし，算出タスクは論理時間を用いた時間駆動処理をすることで，メッセージ通信時間の変動を許容しながら，時間駆動アーキテクチャと同等の動作を実現できる．しかしこれまでは，複数のメッセージを受信する算出タスクにおいても，

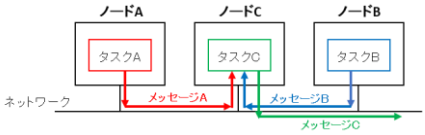

図3：分散処理環境の構成例

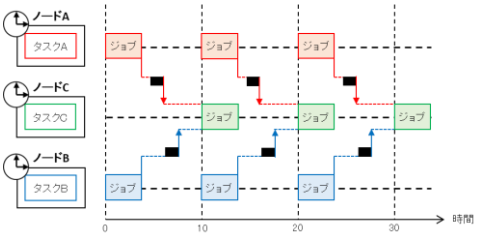

図4：TDMA に基づく通信の例

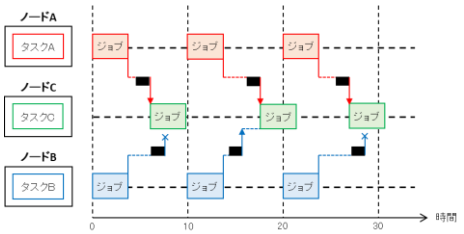

図5：時間的不整合のある処理の例

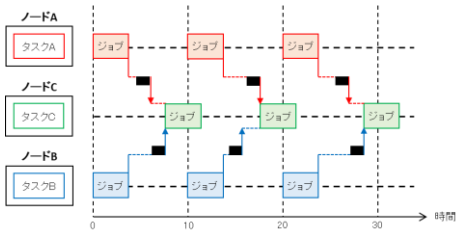

図6：時間的不整合のない処理の例

特定のひとつのメッセージの受信イベントで起動していたため，時間的不整合が発生するという問題があった．次節では，その問題を解決する手法について述べる．

3. 複数メッセージ受信への対応

3.1 時間駆動アーキテクチャにおける動作

TDMA に基づく時間駆動ネットワークを使用した場合は，メッセージ転送タイミングが変動しないため，時間的不整合は発生しない．

図 3 に分散処理環境の構成例を示し，この構成例における時間駆動アーキテクチャの動作を図 4 を用いて説明する．タスク A とタスク B は，同じネットワークを介してメッセージをタスク C に送信する．予め全てのメッセージの送信時刻が定義されているため，複数メッセージを受信する場合でも時間的不整合は発生しない．

3.2 従来の分散処理環境の動作

従来の論理時間を導入した分散環境においては，データの時間的不整合が発生する可能性がある．データの時間的不整合の例を図 5 を用いて説明する．タスク A とタスク B は周期 10 で起動される．特定のメッセージの受信イベントでタスクが起動される．この例では，メッセージ A の受信イベントでタスク C が起動される．タスク C の 1 番目と 3 番目のジョブにおいて，メッセージ B の到着を待たずにタスク C が実行されるため，2 つの受信データにおいて時間的不整合が生じている．

3.3 拡張後の分散処理環境の動作

本論文では，前節のような不整合が発生しない分散処理環境を提案する．すなわち，複数のメッセージの受信イベントにおいて，直近に受信したメッセージのタイムスタンプ値を記録し，全ての受信メッセージのタイムスタンプ値が整合した場合にタスクを起動するよう拡張した．拡張後の不整合のない処理を図 6 を用いて説明する．メッセージの受信タイミングは図 5 と同じである．タスク C はメッセージを受信する毎にタスクの起動判定を行い，同じタイムスタンプを持つメッセージを全て受信した後にタスクを起動する．これにより，不整合のない処理が可能となった．

4. 実装と評価

提案した分散処理環境をプロセッサ H8S/2638F を搭載した評価ボード上に実装した．CPU のクロック周波数は 20MHz である．本分散処理環境は CAN と ZigBee の通信規格に対応している．

拡張した箇所の処理の実行時間を表 1 に示す．一般的な車載システムのタスク起動周期は 10～100 ms であるため，実用上問題ない程度の処理時間と考える．また，より高性能のプロセッサを使用することで，実行時間を短縮できる．

表1：2 入力処理におけるタスク起動判定の実行時間

到着回数	平均[μs]	最大[μs]	最小[μs]
1	48.8	49.0	48.8
2	99.2	99.4	99.0

5. おわりに

本論文では，通信時間が変動するネットワークでも時間駆動アーキテクチャと同等の処理が可能な論理時間を用いた時間駆動分散処理環境において，複数入力処理においても時間的不整合のない分散処理を実現する手法を提案した．今後は，車々間通信や路車間通信で用いられる通信規格 IEEE802.11p に対応できるよう拡張する予定である．

謝辞

本研究は JSPS 科研費 18K11225 の助成を受けたものである．

参考文献

[1]. H.Kopetz, Should Responsive Systems be Event-Triggered or Time-Triggered? , IEICE Transactions on Information and Systems, Vol.E76-D, No.11, pp.1325-1332, 1993.

[2]. A.Ichimura, T.Yokoyama, M.Yoo, A Time-Triggered Distributed Computing Environment for Cyber-Physical Systems Based on Physical Time and Logical Time, Proceeding of TENCON 2018-2018 IEEE Region 10 Conference, pp28-31, 2018.