鮮度と同期度に基づく周期データの選択方式

宗像浩一[†,††] 吉川正俊[†] 植村俊亮[†]

本論文では，周期的に発生するデータ列が複数個与えられた際に，各データ列から1個ずつのデータを選択してデータの組合せを得るための基準として，データの鮮度と同期度を提案する．そしてこれらに基づいてデータを選択するための枠組みを述べる．データの鮮度は，取得したデータの中で一番古いデータの発生時刻から現時刻までの経過時間として定義する．またデータの同期度は，取得したデータの中で最新と最古のデータの発生時刻の差として定義する．そしてデータの同期度に着目して，いくつかの性質を明らかにする．また，データの鮮度と同期度は一般にトレードオフの関係にあることを示す．そこで両者を考慮した評価関数を定義し，これに基づいて各時刻でのデータの最適組合せを求めることを提案する．そしてこれを利用した，連続的に問合せ結果を取得する場合と，逐次的に問合せ結果を取得する場合の処理の枠組みについて述べる．最後に，道路上の障害物と事故とを2種類のセンサで検出するシステムに連続問合せを適用する例を検討する．そして単純に最新データの組合せを用いた場合と比較して，同期度がほぼ半減することを示す．

Acquiring Data Combinations from Periodically Generated Data Sequences Based on Freshness and Synchronousness

KOICHI MUNAKATA,[†,††] MASATOSHI YOSHIKAWA[†] and SHUNSUKE UEMURA[†]

We present query processing frameworks for periodically updated databases which take into account both freshness and synchronousness of the data. Our focus is on *periodically generated data sequences* (pgds's), each having a constant period. Different sequences, in general, have different periods of data generation. In order to process queries to the databases that include such data sequences, we propose new criteria for choosing the optimum combinations of the data to use: freshness and synchronousness of the data. Freshness and synchronousness are, in general, in the relation of trade-off. First, we reveal some properties on the synchronousness of pgds's. Based on the properties, we propose an evaluation function that takes into account both freshness and synchronousness of the data, in order to choose the optimum data combination. Then, we present query processing frameworks for *consecutive queries* and *interactive queries*. Once a *mediator* receives a consecutive query, it computes and returns the query results repeatedly whenever it acquires the optimum combinations of the data. For efficiency, the mediator pre-computes the points of time that give the optimum data combinations. We show the pre-computation results can also be used for the interactive query which enables application programs to set limits on the data synchronousness and response time.

1. はじめに

様々なリアルタイムシステムにおいて，周期的に発生するデータが利用されている．このようなシステムは，発電所，鉄鋼・化学分野などの産業プラント，上下水道施設，飛行機，ロケット，衛星などの制御システム，知的道路交通システム（ITS; Intelligent Transport Systems）など多岐に及ぶ．周期的に発生する複数のデータ列の中に，互いに異なる周期で発生するデータ列が含まれる場合には，ある時点における各データの瞬時値（スナップショット）の近似値を得ることが必要となる．一例として，知的道路交通システムについて考える．ここでは，交通状況，道路路面状況，他自車走行状況，障害物状況，天候状況などをセンサで検出してこれらを複合して処理し，種々の道路交通システムを実現することを目指している[9]．この中で道路

† 奈良先端科学技術大学院大学情報科学研究科
Graduate School of Information Science, Nara Institute of Science and Technology
†† 三菱電機 産業システム研究所
Industrial Electronics & Systems Laboratory, Mitsubishi Electric Corporation

上の障害物と事故とを検出し，後続車を減速／停止させて衝突を回避するシステムが提案されている．これを実現するために道路状況を検出するセンサとして，データ発生周期が異なる次のものが考えられる．

- 道路上の落下物や事故車両の有無を画像処理によって検出するセンサ (周期 33ms)
- 道路上の障害物を検出するスキャンレーザレーダ (周期 100ms)
- 路面状況（乾燥，湿潤，凍結，積雪の 4 状態の区別）を検出するレーザレーダ (周期 150ms)

このように複数のセンサで互いにデータ取得周期が異なる場合，定期的に全データのスナップショットの近似値を取得し，これを処理する必要がある．

周期的に発生する複数のデータを利用するリアルタイムシステムの開発に際して，システム全体を最初から設計･構築する場合には，発生するデータの周期を同期させて同時点で発生するデータを得ることが通常検討される．しかし，システム構成が複雑化するにつれて，システムの全てのセンサ出力を単独の固定周期に同期させるのは困難になる．まず異なる固定周期を持つ複数のセンサを用いる場合や，異なる固定周期を基準に設計された複数のシステムを統合して新たなシステムを構築する場合には，周期の異なるデータを利用することが必要となる．また大規模なシステムの中で一部のセンサを更新し，データの発生周期がより短い高機能なセンサに置き換えるような場合にも，周期の異なるデータを利用することが必要となる．周期の異なるデータを利用する際に，データ取得周期に比較して値が緩やかに変化するデータを用いる場合には，既に得られたデータを外挿または内挿によって補間することで，同期したデータの近似値を得ることができる．しかしセンサの出力はこのような条件を満たすデータのみではなく，値がデータ取得周期に比較して急激に変化するデータや，離散的な値に状態（例えば発生アラームの種別や機器の状態）を割り付けたデータを含む場合がある．このような場合には，データの補間は意味をなさない．

そこで本論文では，異なった周期で発生する種々の周期データ列を統合するために，複数の周期データ列からデータの組合せを選択するための基準を提案する．そしてその基準に基づいてリアルタイムシステムを構築するための，基本的な枠組みについて述べる．異なった周期で発生する複数のデータ列が与えられた際に，データの組合せを選択するための基準として，まずデータが新しい，即ち鮮度が高いことが挙げられる．さらに各データの発生時刻ができる限り近い，即ち同期度が高いことが挙げられる．選択するデータの組合せに関して，データの鮮度と同期度は，トレードオフの関係になる場合がある．ある時点でデータの複数の組合せが得られる場合，ある組合せにおいてはデータの鮮度は高いが同期度が低く，これと比較して別の組合せにおいては鮮度が低いが同期度は高い，という場合がある．このような場合に，鮮度と同期度の妥協点を見いだしてデータの組合せを決定する必要がある．

またリアルタイムシステムにおいては，周期的に発生する複数のデータ列を連続的に取得するという要求がしばしば発生する．例えば前述した車両の衝突回避システムにおいて，道路上の落下物や事故車両の有無を検出するセンサ，道路上の障害物を検出するスキャンレーザレーダ，路面状況を検出するレーザレーダの各出力を，道路管制センタのグラフィックパネルに表示するような場合が考えられる．このように発生するデータの周期が異なる場合，どのタイミングで取得したデータを更新して再処理するかが問題となる．ひとつには，全てのデータが同時に発生した場合にデータを更新して，新たなデータを用いることが考えられる．しかし一般にはデータが同時に発生することが決してない場合もあり得る．さらに，各データの発生周期の最小公倍数が長く，実際にはそれよりも短い周期でデータを更新する必要が生じる場合がある．別の方法として，任意のデータが発生する毎にそれを反映することが考えられる．この場合には，データ間で発生時刻が許容範囲を越えて大幅に異なるという場合があり得る．さらに次々と新しいデータが発生するために，それらを反映するための処理が膨大になり，CPU の処理速度が不足するような事態が起こり得る．そこでこのような問題に対処する方法として，連続問合せのための枠組みを提案する．この枠組みにおいては，データの鮮度と同期度を考慮した評価関数を定義し，これに基づいて最適となるデータの組合せを決定する．さらにこの枠組みを転用して，逐次問合せを実現するための方法を提案する．そこでは従来の問合せ文に加えて，データの同期度の上限と問合せ結果を得るまでに待つことができる最大許容時間とを指定する．これにより，問合せが発生してから指定された最大許容時間まで新たなデータの発生を待つことが可能になり，データの組合せで鮮度と同期度がともに優れているものを得られる可能性が出てくる．

本論文の構成は以下の通りである．2 節で異なった周期で発生するデータ列からデータの組合せを選択するための様々な問題点を，例題を用いて述べる．3 節で関連研究について述べる．4 節で周期データ列の同

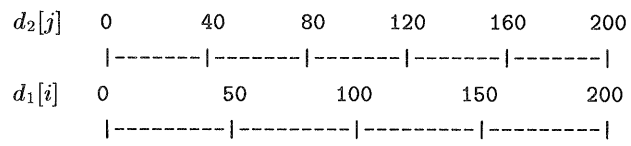


図1 $\tau_1 = 50$, $\tau_2 = 40$ の場合の，最小公倍数以下の **pgds** d_1 と d_2 の値.
Fig. 1 **pgds**'s d_1 and d_2 within the LCM, where $\tau_1 = 50$ and $\tau_2 = 40$.

期度に着目した性質を述べる．5節で異なった周期のデータ列からデータの組合せを選択するための基準として，データの鮮度と同期度を形式的に定義し，さらに両者を考慮した評価関数を提案する．6節でこの評価関数に基づき，連続問合せと逐次問合せを実現するための枠組みを述べる．7節で本手法を道路上の障害物検出システムに適用する例を検討し，本手法の有用性を示す．8節で種々の問題について論じる．9節はまとめである．

2. 例　　題

図1に，2個の周期データ列の発生点の例を示す．以後周期データ列を **pgds** (Periodically Generated Data Sequence の略) と呼ぶ．**pgds** d_1 ではデータは50単位時間毎に発生し，d_2 では40単位時間毎に発生している．この図では両者の最小公倍数である200までのデータ発生時点を示している．また d_1 と d_2 の初期発生時刻は0である．ここで，d_1 と d_2 のデータがそれぞれ1個ずつ必要となる問合せが発生したとする．本論文では，**pgds** から1個ずつ取得したデータの組合せにおいて，その鮮度を，取得したデータの中で一番古いデータの発生時刻から現時刻までの経過時間として定義し，同期度を，取得したデータの中で最新と最古のデータの発生時刻の差として定義する．また，問合せ処理の時間は，データの発生間隔に比較して充分に短いものと仮定して，0とする．従って問合せ結果は，データがそろい次第直ちに返されるものとする．

まず逐次問合せについて考える．利用者が d_1 と d_2 のデータを必要とする問合わせを時刻0で発行すると，前述の仮定に基づき，直ちに結果が得られる．この場合，両方のデータの鮮度と同期度はともに0である．次に利用者が時刻155で問合せを発行する場合を検討する．この場合，データの2種類の組合せが候補となる．即ち，$[t_1, t_2] = [100, 120]$ と $[t_1, t_2] = [150, 120]$ である．ここで t_1 と t_2 は，**pgds** d_1 と d_2 の発生時刻を表すものとする．もし $[t_1, t_2] = [100, 120]$ を選択した場合，これらの発生時刻は20単位時間離れており，問合せ結果が得られる時点で古い方のデータ発生時刻100から，55単位時間が経過することになる．一方 $[t_1, t_2] = [150, 120]$ を選択した場合では，これらの発生時刻は30単位時間離れており，問合せの結果が得られる時点で古い方のデータ発生時刻120から，35単位時間が経過することになる．もしデータの鮮度よりも同期度を優先する場合には，$[100, 120]$ のデータの組合せが望ましく，逆に鮮度を優先する場合には $[150, 120]$ の方が望ましい．しかし7節で述べる例のようにアプリケーションによっては，データ発生時刻が一定時間以上離れることが許されない場合もあり得る．例えばその許容値が25単位時間の場合には $[100, 120]$ を用いることになる．以上とは異なるデータ組合せの選択方法として，5単位時間待つという方法も考えられる．すると時刻160で $[150, 160]$ というデータの組合せが新たに得られる．この場合にはこれらの発生時刻は10単位時間離れており，問合せ結果が得られる時点で古い方のデータ発生時刻150から，10単位時間が経過することになる．上述のデータ組合せと比較して，データの鮮度，同期度の双方でこの場合がより望ましい値となる．

次に連続問合せについて検討する．例えば，道路管制センタのグラフィックパネルに各センサの値を表示する場合を想定する．もし毎回のデータ表示更新周期が，上述の d_1 と d_2 の最小公倍数である200単位時間より短いことが要求される場合，データ発生時刻が完全に一致しない場合でも両方のデータの適当な組合せを選択して随時提示する必要がある．もし10単位時間以下の発生時刻の差が許容される場合，$[0, 0]$ に加えて，$[50, 40]$ と $[150, 160]$ の組合せが利用できる．この場合 $[50, 40]$ を用いた問合せの結果は，時刻160まで表示されることになり，その時点で古い方のデータの発生から120単位時間が経過している．7節で述べる例のようにアプリケーションによっては，データの鮮度が重要となる場合もある．120単位時間が許容範囲を越えている場合にはこれらの組合せに代えて，発生時刻の差が20となる $[100, 80]$ および $[100, 120]$ を用いるのが適切となる．

この論文では以上述べた問題点を解決するための枠組みを示す．データの鮮度と同期度は上述したように

一般にはトレードオフの関係にある．そこで両者の妥協点を見いだすために評価関数を定義し，これを最小とするデータ組合せを選択することを提案する．この評価関数は連続問合せと逐次問合せの両方で利用することができる．

3. 関連研究

周期データ列を取得して処理する研究に関しては，これまで主にリアルタイムデータベースとテンポラルデータベースの分野で研究されてきた．しかし周期的に発生するデータ列の，データの鮮度を扱った研究はわずかしか発表されておらず，さらにデータの同期度を扱った研究は筆者らが知る限りは存在しない．

リアルタイムデータベースの分野では，主な力点は論文1) にあるように，トランザクション処理に置かれていた．即ち各種のタスクに優先度を割り付け，いかにタスクの実行をスケジュールするかが主要な課題であった．論文2) においては，種々のタスクのスケジュール方法が述べられている．そこではどの方法においても，直ちに利用できる最新のデータが用いられており，本論文におけるようなデータ組合せの各種の可能性については考慮されていない．論文3) ではいくつかのスケジュール方法にデータのデッドラインという概念を付加した上で，性能の比較を行っている．データデッドラインとは各データの有効期限であり，データを更新するトランザクションは，全てのデータの有効期限より以前にコミットする必要がある．さらに強制待機（forced wait）という概念が提案されている．これによると，トランザクションがデータを参照した際に，データの有効期限までにトランザクションが終了できるかどうかを調べる．もし終了しない可能性があれば，トランザクションは新たなデータが発生するまで強制的に停止される．この際に，タスクをスケジュールするための主な基準は，トランザクションが強制終了される割合と，CPU の利用率である．データの有効期限の概念は，本論文で述べたデータの鮮度と類似しているが，次の点が異なる．まずデータの有効期限はファームデッドラインであり，この期限を少しでも過ぎると，トランザクションは強制終了される．それに対してデータの鮮度はソフトデッドラインを持つ．データは時間の経過とともに徐々に劣化していくと捉え，それらよりも評価関数の値を小さくするデータの組合せが発生した時点でデータを交代させる．さらにデータの有効期限の概念ではデータの発生時刻を考慮せず，従ってデータ間での同期ということも考慮されない．本論文ではデータの組合せに関して，たとえ発生時刻が異なっていても，データ間の同期度を考慮した評価関数の値が他のデータの組合せよりも小さい場合にはそれらを使うことになる．また8節で述べるように，データの発生時刻が一定時刻異なるべき周期データ列間の組合せでも，同期がとれているとして扱う場合がある．論文4) では，周期的に更新されるデータを扱うトランザクションに対して実行期間を割り付ける方法について提案している．重要なデータに関して高い優先度を割り付けることにより，トランザクションの実行頻度がそのデータの更新頻度に近づくようにしている．このように目標とするところはタスクへの最適な実行時間の割り付けであり，筆者らの目標と異なる．

テンポラルデータベースの分野では，これまで主に時間データのモデル化と問合せについて研究されてきた．周期的に発生するデータ列を扱った文献は少ない．論文5) では，関係代数を拡張し，周期データ列（linear repeating point）を扱えるようにしている．そのために従来の関係（relation）を一般関係（generalized relations）に拡張して，各属性に制約を持たせている．さらに時間的な問合せ言語を提案している．ここでは周期的なデータのモデル化とそれに対する演算の定義を行っており，最適なデータ組合せを求める筆者らの課題とは異なる．論文6) でも周期的な時間データを扱っている．ここではデータを厳密な周期的事象（strictly-periodic events）と部分的な周期的事象（partially-periodic events）に関して，統一的な枠組みの中でモデル化している．そして集合演算に関してそのモデルが閉じていることを示している．さらにオブジェクト指向 SQL を拡張することにより，時間的な問合せを表現できることを示している．この研究でも，主要な課題は周期的なデータのモデル化であり，本論文の課題と異なる．

4. 周期データの性質

4.1 基本的な定義

以下では簡単のため，時刻を順序を持ち等間隔な離散的なデータ列と仮定し，0 で始まる正の整数を各時刻に順に割り付けることにする．

定義1 (周期データ列) 周期データ列 (**pgds**; Periodically Generated Data Sequence) d_i を次のように定義する:

$$d_i = \langle d_i[0], d_i[1], ... \rangle$$

但しこれに対応した周期データ発生時刻 $t_i[k]$ $(k = 0, 1, ...)$ は次のように記述できるものとする．

$$t_i[k] = t_i[0] + k \times \tau_i$$

ここで τ_i は d_i のデータ発生周期であり，正の整数である．また k を $t_i[k]$ のインデックスと呼ぶ．さらに周期データ発生時刻列 t_i を次のように定義する:

$$t_i = \langle t_i[0], t_i[1], ... \rangle$$

□

定義 2 (インデックス生成関数) 時刻 t における **pgds** d_i のインデックス生成関数 $I_i(t)$ を次のとおり定義する．

$$I_i(t) = \lfloor (t - t_i[0])/\tau_i \rfloor$$

但し $\lfloor real \rfloor$ は実数 $real$ 以下の最大の整数を表す．□

ここで $I_i(t_i[k]) = k$ となる点に注意されたい．

以下では簡単のため，τ_i と τ_j の最小公倍数を $\tau_{i,j}^{LCM}$，最大公約数を $\tau_{i,j}^{GCD}$ と表す．

4.2 同期度の性質

本節では，2 個の **pgds** d_1 と d_2 の性質を述べる．

d_1 と d_2 の周期 τ_1 と τ_2 は次のように記述できる．

$$\tau_1 = \tau_{1,2}^{GCD} \times \tau_1'$$
$$\tau_2 = \tau_{1,2}^{GCD} \times \tau_2'$$

ここで τ_1' と τ_2' は互いに素な正の整数である．また次の式が成立する．

$$\tau_{1,2}^{LCM} \times \tau_{1,2}^{GCD} = \tau_1 \times \tau_2 \quad (1)$$
$$\begin{aligned} \tau_{1,2}^{LCM} &= \tau_{1,2}^{GCD} \times \tau_1' \times \tau_2' \\ &= \tau_1 \times \tau_2' \\ &= \tau_2 \times \tau_1' \end{aligned}$$

τ_2 に関する $t_1[k]$ の剰余 $r_1[k]$ は次式で計算できる．

$$r_1[k] = t_1[k] - \lfloor t_1[k]/\tau_2 \rfloor \times \tau_2$$

以下では簡単のため，$r_1[k]$（但し k は 0 以上の整数）は常に τ_2 に関する $t_1[k]$ の剰余を表すものとする．

ここで $t_1[0] = t_2[0] = 0$ となる場合，

$$\min_i |t_2[i] - t_1[j]| = \min(r_1[j], \tau_2 - r_1[j])$$

となることに注意されたい．この式は次のことを表現している．即ち，まず **pgds** d_1 から任意のデータ $d_1[j]$ を選ぶ．このデータの発生時刻は $t = t_1[j]$ である．次に **pgds** d_2 から，$t_1[j]$ に最も近い時刻 $t = t_2[i]$ に発生するデータ $d_2[i]$ を選ぶ．両者の差 $t_2[i] - t_1[j]$ の絶対値は，$r_1[j]$ と $(\tau_2 - r_1[j])$ の小さい方の値となる．

図 2 (a) は周期 $\tau_1 = 6$ の **pgds** d_1 と周期 $\tau_2 = 10$ の **pgds** d_2 に関して，両者の最小公倍数 30 までに発生する $t_1[k]$ の τ_2 に関する剰余 $r_1[k]$ を示している．例えば $t_1[0] = 0$ に対応する $r_1[0] = 0$（図中で 0 {0} と表記）と $t_1[2] = 12$ に対応する $r_1[2] = 2$（図中で 2 {12} と表記）とは時間軸上で隣接しており，その差は 2 である．また $t_1[2] = 12$ に対応する $r_1[2] = 2$ と $t_1[4] = 24$ に対応する $r_1[4] = 4$ も同様で，その差が 2 である．このように，間に他の剰余を含まない，隣接した 2 個の剰余 $r_1[i]$ と $r_1[j]$ の間隔☆は，一定値となる．これに関して，次の定理が成立する．

定理 1 $D_{i,j}$ を，間に他の剰余を含まない隣接した 2 個の剰余 $r_1[i]$ と $r_1[j]$ の距離とする．但し $i \neq j$ かつ $|t_1[i] - t_1[j]| < \tau_{1,2}^{LCM}$ とする．すると次式が成立する．

$$D_{i,j} = \tau_{1,2}^{GCD}$$

証明: 長さが最小公倍数 $\tau_{1,2}^{LCM}$ の任意の区間で発生する d_1 のデータの個数を p とする．但し d_1 が区間の開始時点と終了時点とで発生する場合には，終了時点のデータを除いた個数を p の値とする．すると，次式が成立する．

$$p = \frac{\tau_{1,2}^{LCM}}{\tau_1} \quad (2)$$

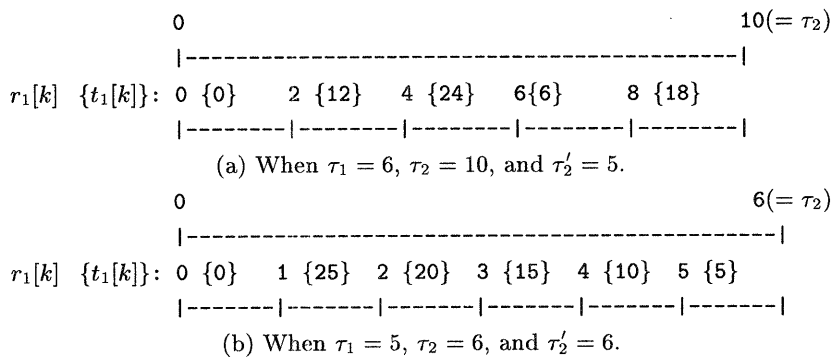
任意の剰余 $r_1[i]$ と $r_1[j]$ の距離を D とすると，次のようになる．

$$\begin{aligned} D = &|r_1[i] - r_1[j]| \\ = &|(t_1[i] - \lfloor t_1[i]/\tau_2 \rfloor \times \tau_2) \\ &-(t_1[j] - \lfloor t_1[j]/\tau_2 \rfloor \times \tau_2)| \\ = &|t_1[0] + i \times \tau_1 - \lfloor t_1[i]/\tau_2 \rfloor \times \tau_2 \\ &-t_1[0] - j \times \tau_1 + \lfloor t_1[j]/\tau_2 \rfloor \times \tau_2| \\ = &|i \times \tau_1 - \lfloor t_1[i]/\tau_2 \rfloor \times \tau_2 \\ &-j \times \tau_1 + \lfloor t_1[j]/\tau_2 \rfloor \times \tau_2| \\ = &\tau_{1,2}^{GCD} \times |i \times \tau_1' - \lfloor t_1[i]/\tau_2 \rfloor \times \tau_2' \\ &-j \times \tau_1' + \lfloor t_1[j]/\tau_2 \rfloor \times \tau_2'| \quad (3) \end{aligned}$$

この式から D は常に $\tau_{1,2}^{GCD}$ の倍数であることが分かる．ここで $i \neq j$ かつ $|t_1[i] - t_1[j]| < \tau_{1,2}^{LCM}$ という前提条件により，D の値が 0 になり得ないことに注意されたい．いま式 (2) と (3) から，隣接した 2 個の剰余 $r_1[i]$ と $r_1[j]$ の距離が常に $\tau_{1,2}^{GCD}$ となることを背理法により証明する．まず，この仮説が誤りであると仮定する．即ち隣接した剰余の組のうち，1 組の距離が $\tau_{1,2}^{GCD} \times 2$ 以上であるとし，それ以外の (p - 1) 組の距離が $\tau_{1,2}^{GCD}$ であると仮定する．すると，次のようになる．

$$\begin{aligned} \tau_2 &\geq (p-1) \times \tau_{1,2}^{GCD} + 2 \times \tau_{1,2}^{GCD} \\ &= p \times \tau_{1,2}^{GCD} + \tau_{1,2}^{GCD} \\ &= \frac{\tau_{1,2}^{LCM}}{\tau_1} \times \tau_{1,2}^{GCD} + \tau_{1,2}^{GCD} \quad (\text{式 (2) より}) \\ &= \frac{\tau_{1,2}^{LCM} \times \tau_{1,2}^{GCD}}{\tau_1} + \tau_{1,2}^{GCD} \\ &= \frac{\tau_1 \times \tau_2}{\tau_1} + \tau_{1,2}^{GCD} \quad (\text{式 (1) より}) \\ &= \tau_2 + \tau_{1,2}^{GCD} \end{aligned}$$

☆ 一般には $j = i \pm 1$ は成立しない．



(a) When $\tau_1 = 6$, $\tau_2 = 10$, and $\tau_2' = 5$.

(b) When $\tau_1 = 5$, $\tau_2 = 6$, and $\tau_2' = 6$.

図 2 0 から d_1 と d_2 の周期の最大公約数までの $t_1[k]$ に対応する，$r_1[k]$ の分布.
Fig. 2 The distribution of $r_1[k]$'s, each corresponding to $t_1[k]$ between zero and the LCM of τ_1 for d_1 and τ_2 for d_2.

$\tau_2 + \tau_{1,2}^{GCD}$ の値は実際には τ_2 の値より大きいため，明らかにこの式は誤りである．また隣接した剰余の組のうち，n $(2 \leq n \leq p)$ 組の距離が $\tau_{1,2}^{GCD} \times 2$ 以上であるとし，それ以外の (p - n) 組の距離が $\tau_{1,2}^{GCD}$ である場合にも，同様に矛盾が生じる．即ち仮説は誤りであり，隣接した任意の剰余 $r_1[i]$ と $r_1[j]$ の距離が $\tau_{1,2}^{GCD}$ より大きくなることはない．一方，隣接した任意の剰余の距離は $\tau_{1,2}^{GCD}$ の倍数となることを既に証明した．以上から，隣接した任意の剰余 $r_1[i]$ と $r_1[j]$ の距離はちょうど $\tau_{1,2}^{GCD}$ であることが証明される． □

定義 3 **(同期度の総和)** $E_{sum}^{sync}(d_1, d_2)$ を d_1 と d_2 の同期度の総和と呼び，次のとおり定義する．

$$E_{sum}^{sync}(d_1, d_2) = \sum_{j=m,m+1,\ldots,m+\tau_2'-1} \min_i |t_2[i] - t_1[j]|$$

但し m は整数で，$t_2[0] - t_1[m] \leq \tau_2/2$ とする． □

これは $E_{sum}^{sync}(d_1, d_2)$ の第 1 引数 d_1 に対応する，時間区間 $\tau_{1,2}^{LCM}$ 未満の $t_1[j]$ の各データを基準として，そのデータの発生時刻と一番近い $t_2[i]$ の発生時刻との差の絶対値を足し合わせたものである．E_{sum}^{sync} の値が小さいほど d_1 と d_2 の同期度が高いといえる．なお，$t_1[0] = t_2[0] = 0$ の場合，同期度の総和は次式で計算できる．

$$E_{sum}^{sync}(d_1, d_2) = \sum_{j=0,\ldots,\tau_2'-1} \min_i |t_2[i] - t_1[j]|$$

定義 4 **(同期度平均)** $E_{ave}^{sync}(d_1, d_2)$ を d_1 と d_2 の同期度平均と呼び，次のとおり定義する．

$$E_{ave}^{sync}(d_1, d_2) = E_{sum}^{sync}(d_1, d_2)/\tau_2' \quad (4)$$

□

これは $\tau_{1,2}^{LCM}$ の時間区間で発生する $t_1[j]$ の τ_2' 個のデータについて，その各データの発生時刻と一番近い $t_2[i]$ の発生時刻との差の絶対値を平均したものである．

$t_1[0] = t_2[0] = 0$ の場合，剰余 $r_1[k]$ の分布は，0 と τ_2 との間で，中点について対称となる．またもし τ_2' が偶数であれば，図 2 (b) に示すように中点となる $\tau_2/2$ にデータが存在する．もし τ_2' が奇数であれば，図 2 (a) に示すように，中点となる $\tau_2/2$ にはデータが存在しない．このことから，次の命題が得られる．

命題 1 $t_1[0] = t_2[0] = 0$ の場合，τ_2' が偶数であれば，次式が成立する．

$$\begin{aligned} & E_{sum}^{sync}(d_1, d_2) \\ &= 2 \times \sum_{k=1,\ldots,\tau_2'/2-1} (k \times \tau_{1,2}^{GCD}) + \frac{\tau_2'}{2} \times \tau_{1,2}^{GCD} \\ &= 2 \times \frac{\{1 + \tau_2'/2 - 1\} \times (\tau_2'/2 - 1)}{2} \times \tau_{1,2}^{GCD} \\ &\quad + \frac{\tau_2'}{2} \times \tau_{1,2}^{GCD} \\ &= \frac{\tau_2'(\tau_2' - 2)}{4} \times \tau_{1,2}^{GCD} + \frac{\tau_2'}{2} \times \tau_{1,2}^{GCD} \\ &= \frac{(\tau_2')^2}{4} \times \tau_{1,2}^{GCD} \end{aligned} \quad (5)$$

また τ_2' が奇数であれば次式が成立する．

$$\begin{aligned} & E_{sum}^{sync}(d_1, d_2) \\ &= 2 \times \sum_{k=1,\ldots,(\tau_2'-1)/2} (k \times \tau_{1,2}^{GCD}) \\ &= 2 \times \frac{\{1 + (\tau_2' - 1)/2\} \times (\tau_2' - 1)/2}{2} \times \tau_{1,2}^{GCD} \\ &= \frac{(\tau_2' + 1)(\tau_2' - 1)}{4} \times \tau_{1,2}^{GCD} \\ &= \frac{(\tau_2')^2 - 1}{4} \times \tau_{1,2}^{GCD} \end{aligned} \quad (6)$$

□

定理 2 **pgds** である d_1 と d_1' とが同じ周期を持ち，また d_2 と d_2' とが同じ周期を持つとする．これら 4 個の **pgds** d_1, d_2, d_1' およびは d_2' は任意の初期データ発生時刻を持つものとする．すると同期度の総

和 $E^{sync}_{sum}(d_1,d_2)$ と $E^{sync}_{sum}(d'_1,d'_2)$ に関して次式が成立する．

$$|E^{sync}_{sum}(d_1,d_2) - E^{sync}_{sum}(d'_1,d'_2)| \leq \tau^{GCD}_{1,2}/2$$

証明: 初期データ発生時刻 $t_1[0]$ と $t_2[0]$ とをまず 0 に固定し，次に $t_1[0]$ を移動させた場合の同期度の総和の差が $\tau^{GCD}_{1,2}/2$ 以下であることを示すことにより，一般性を失わずに定理を証明できる．剰余 $r_1[k]$ に関して定理 1 から，隣接した剰余間の距離は常に $\tau^{GCD}_{1,2}$ であり，またその分布は 0 と τ_2 の中点に関して対称であるため，$t_1[0]$ を移動させる距離は 0 から $\tau^{GCD}_{1,2}/2$ までを考慮すればよい．まず τ'_2 が偶数である場合を考える．この例を図 2 (b) に示す．$t_1[0]$ を 0 から n まで移動させることを考える．但し $1 \leq n \leq \tau^{GCD}_{1,2}/2$ とする．剰余 $r_1[k]$ （但し $0 \leq r_1[k] < \tau_2/2$）の各点は τ_2 からの距離に比べて 0 からの距離の方が短い．そして 0 からの距離は $t_1[0]$ の移動に伴い，n だけ増加する．一方各 $r_1[k]$ について $r_1[k]+\tau_2/2$ に剰余が存在し，その点は 0 からの距離に比べて τ_2 からの距離の方が短いか，または両者が等しい．この各点からの τ_2 までの距離は，$t_1[0]$ の移動に伴って n だけ減少する．従って同期度の総和 $E^{sync}_{sum}(d_1,d_2)$ について，$r_1[k]$ と $r_1[k]+\tau_2/2$ に位置する剰余との増減が相殺され，総和は不変となる．次に τ'_2 が奇数の場合を考える．この例を図 2 (a) に示す．この場合には $t_1[0]$ の移動に伴って，$\tau^{GCD}_{1,2} \times (\tau'_2-1)/2$ に位置する剰余のみに関して，0 からの距離の増加 n を相殺する点が存在しない☆．従って同期度の総和 $E^{sync}_{sum}(d_1,d_2)$ は n だけ増加する．先に述べたように n の値としては $\tau^{GCD}_{1,2}/2$ 以下を考慮すればよい．以上より同期度の総和 $E^{sync}(d_1,d_2)$ と $E^{sync}(d'_1,d'_2)$ との差は最大 $\tau^{GCD}_{1,2}/2$ となる． □

以上述べた定理と命題とから，**pgds** は次の性質を持つことが分かる．

- τ'_2 が奇数の場合を考える．同期度の最小値，最大値ともに，初期データ発生時刻 $t_1[0]$ と $t_2[0]$ が一致する場合に，最小の値をとる．この際に同期度の最小値は 0 であり，最大値は $\tau^{GCD}_{1,2} \times (\tau'_2-1)/2$ となる．これらの値は，$t_1[0]$ または $t_2[0]$ を任意の整数 i に関して $\tau^{GCD}_{1,2}/2 + i \times \tau^{GCD}_{1,2}$ だけ移動することにより，それぞれ $\tau^{GCD}_{1,2}/2$ だけ増加して最大となる．その結果，同期度の最小値は $\tau^{GCD}_{1,2}/2$，最大値は $\tau_2/2$ となる．
- τ'_2 が奇数の場合を考える．この場合同期度の総和 E^{sync}_{sum} は，$t_1[0]$ と $t_2[0]$ が同時に発生する場合に最小になり，その値は式 (6) で与えられる．また $t_1[0]$ と $t_2[0]$ の発生時刻の差が任意の整数 i に関して $\tau^{GCD}_{1,2}/2 + i \times \tau^{GCD}_{1,2}$ の場合に最大になり，その値は式 (6) から次のように計算できる．

$$\begin{aligned} &E^{sync}_{sum}(d_1,d_2) \\ &= \frac{(\tau'_2)^2-1}{4} \times \tau^{GCD}_{1,2} + \tau^{GCD}_{1,2}/2 \\ &= \frac{(\tau'_2)^2+1}{4} \times \tau^{GCD}_{1,2} \end{aligned}$$

- τ'_2 が偶数の場合を考える．初期データ発生時刻 $t_1[0]$ と $t_2[0]$ が一致する場合に，同期度の最小値は 0 となる．しかしその場合に同期度の最大値は $\tau_2/2$ となり，最大の値をとる．この値は，$t_1[0]$ または $t_2[0]$ を任意の整数 i に関して $\tau^{GCD}_{1,2}/2 + i \times \tau^{GCD}_{1,2}$ だけ移動することにより，$\tau^{GCD}_{1,2}/2$ だけ減少して最小となる．しかしそれにより，$t_1[i]$ と $t_2[j]$ のどの点も同時に発生することがなくなり，同期度の最小値が 0 から $\tau^{GCD}_{1,2}/2$ に増加して，最大の値をとることになる．
- τ'_2 が偶数の場合を考える．この場合，同期度の総和 E^{sync}_{sum} は式 (5) で与えられ，$t_1[0]$ あるいは $t_2[0]$ が変化しても不変である．

以上述べた性質は，周期の異なる複数の **pgds** からデータを取得するシステムを構築する際に利用して，データ組合せの同期度を向上させることができる．

5. 鮮度と同期度

周期の異なる複数個の **pgds** から，データ組合せを選択する基準としては，計測誤差，鮮度，同期度などいくつかの基準が考えられる．1 節で述べた道路上の障害物と事故を検出して衝突を回避するシステムにおいては，7 節で詳しく述べるように，複数のセンサで障害物を同時に捕捉するために，同期度が重要となる．また後続車にできる限り早く減速を促して衝突を回避するために，データの鮮度も重要となる．

そこで本節では，鮮度と同期度に着目してこれらを形式的に定義するとともに，それらを組合わせた評価関数を導入する．

5.1 鮮　　度

定義 5 **(データ組合せ)** n 個の **pgds** として d_1 から d_n が与えられたとする．これらの **pgds** に対するデータ組合せは，n 次元の配列 C の要素で識別される n 個のデータと定義する．但し配列 C の i 番目の要素 $C[i]$（但し $i=1,2,...,n$）は $d_i[j]$ へのインデックス j である．即ち，次のとおりである．

$$C = [C[1], C[2], ..., C[n]]$$

□

☆ 剰余 $r_1[\tau'_2]$ は次の周期に属する．

以後，配列 C で識別されるデータ組合せを，データ組合せ C と呼ぶ．

定義 6 **(鮮度)** 時刻 t におけるデータ組合せ C の鮮度 $E^{fresh}(C,t)$ を次のとおり定義する．

$$E^{fresh}(C,t) = t - \min_{i=1,2,\ldots,n}(t_i[C[i]]) \qquad (7)$$

□

アプリケーションによっては鮮度に上限を設けるのが適当な場合がある．これは $E^{fresh}(C,t)$ を修正し，上限を越えたデータを含むデータ組合せに対して十分に大きな値を返すようにすることにより実現できる．

5.2 同期度

定義 7 **(同期度)** データ組合せ C の同期度 $E^{sync}(C)$ を次のとおり定義する．

$$E^{sync}(C) = \max_{i=1,2,\ldots,n}(t_i[C[i]]) - \min_{j=1,2,\ldots,n}(t_j[C[j]]) \qquad (8)$$

□

鮮度の場合と同様に，アプリケーションによっては同期度に上限を設けるのが適当な場合がある．この場合にも $E^{sync}(C)$ を変更し，上限を越えたデータを含むデータ組合せに対して十分に大きな値を返すようにすることにより実現できる．

5.3 評価関数

定義 8 **(評価関数)** 時刻 t におけるデータ組合せ C の評価関数 $E(C,t)$ を次のとおり定義する．

$$E(C,t) = E^{fresh}(C,t) + w \times E^{sync}(C)$$

ここで w は E^{sync} の重みで正の実数である． □

定義 9 **(最適データ組合せ)** 時刻 t における最適データ組合せ $C^{opt}(t)$ とは，その時刻において最小の評価関数 $E(C,t)$ の値を与えるデータ組合せであると定義する． □

定義 10 **(最適評価関数値)** 時刻 t における最適評価関数値 $E^{opt}(t)$ を次のとおり定義する．

$$E^{opt}(t) = E(C^{opt}(t),t)$$

□

6. 問合せ処理

6.1 システム構成

ここでは，**pgds** を含むデータベースに対しての問合せ処理の枠組みを述べる．システム構成としては，メディエータ（mediator）を用いたデータ統合システム [8] を想定する．ここではメディエータによって，データ列の周期が異なるという情報源の異種性を調停することになる．各情報源が一定の周期 τ_i で周期データ列 d_i を発生するものとする．各情報源毎にラッパ（wrapper）を作成する．利用者からのデータ獲得要求を受けて，メディエータはこれをもとに各ラッパ毎のデータ獲得要求を作成する．そして各ラッパでは情報源で周期データが発生する毎にこれをメディエータに送出する．メディエータではラッパから受け取ったデータを一時的に保存するが，不必要になった時点で消去して構わないものとする．

6.2 連続問合せ

本節では，連続問合せについて述べる．メディエータがアプリケーションプログラムから連続問合せを受けると，新たな最適データ組合せをラッパから獲得するごとに処理を実行し，結果をアプリケーションプログラムに返す．メディエータが連続問合せを受けると，まずどの時点で最適データ組合せが発生するかを予め計算し，表 1 に示すような中間表を作成する．この表には，時刻 0 で始まり各周期データ列の最小公倍数となる時刻までの各時刻での，インデックス生成関数の値，各データ組合せの同期度，鮮度，評価関数値および最適評価関数値が記述される．表 1 において想定した各パラメータは次のとおりである．

- $t_1[0] = t_2[0] = 0$.
- $\tau_1 = 4$, $\tau_2 = 3$.
- 同期度の重み $w = 4$.

上記の例では，全ての周期データ列の初期データ発生時刻を 0 としている．しかし，もし全ての **pgds** が同時に生成される時刻が存在しない場合には，メディエータは全てのデータ組合せから最小の同期度を実現するものを選び，その最後のデータの発生時刻を $t = 0$ とみなして中間表を作成する．

中間表を作成するにあたり，複数のデータ組合せ間で E^{fresh} と E^{sync} の双方について値が一致する場合がある．この場合にはこれらのデータ組合せを構成するデータの中で，**pgds** 毎に最も新しいデータを選び，それらから構成されるデータ組合せを用いるものとする．また複数のデータ組合せ間で互いに E^{fresh} と E^{sync} の値が異なるにもかかわらず，同じ値の E^{opt} が発生する場合がある．これは，重み w として適当な実数値を割り付けることにより，発生を防ぐことが可能である．また事前に，E^{opt} が同じ値をとった場合に，E^{fresh} と E^{sync} とどちらを優先するかを決めておくことも有効である．

表 1 では **pgds** d_1 と d_2 の全てのデータ組合せを記載しているが，実際にはメディエータはこれら全ての情報を保持する必要はない．中間表を作成する際にメディエータは時刻 $t = 0$ から開始し，時刻 $t = \tau_{1,2}^{LCM}$

表 1 周期データ列 d_1 と d_2 の中間表．但し $\tau_1 = 4$, $\tau_2 = 3$ である．
Table 1 An example of the intermediate table for d_1 and d_2, where $\tau_1 = 4$ and $\tau_2 = 3$.

t	0	1	2	3	4	5	6	7	8	9	10	11
$I_1[t]$	0	0	0	0	1	1	1	1	2	2	2	2
$I_2[t]$	0	0	0	1	1	1	2	2	2	3	3	3
$E^{sync}([0,0])$	0											
$E^{fresh}([0,0])$	0	1	2	3	4	5	6	7	8	9	10	11
$E^{fresh}([0,0]) + 4 \times E^{sync}([0,0])$	0	1	2	3	4	5	6	7	8	9	10	11
$E^{sync}([0,1])$	3											
$E^{fresh}([0,1])$	-	-	-	3	4	5	6	7	8	9	10	11
$E^{fresh}([0,1]) + 4 \times E^{sync}([0,1])$	-	-	-	15	16	17	18	19	20	21	22	23
$E^{sync}([0,2])$	6											
$E^{fresh}([0,2])$	-	-	-	-	-	-	6	7	8	9	10	11
$E^{fresh}([0,2]) + 4 \times E^{sync}([0,2])$	-	-	-	-	-	-	30	31	32	33	34	35
$E^{sync}([0,3])$	9											
$E^{fresh}([0,3])$	-	-	-	-	-	-	-	-	-	9	10	11
$E^{fresh}([0,3]) + 4 \times E^{sync}([0,3])$	-	-	-	-	-	-	-	-	-	45	46	47
$E^{sync}([1,0])$	4											
$E^{fresh}([1,0])$	-	-	-	-	4	5	6	7	8	9	10	11
$E^{fresh}([1,0]) + 4 \times E^{sync}([1,0])$	-	-	-	-	20	21	22	23	24	25	26	27
$E^{sync}([1,1])$	1											
$E^{fresh}([1,1])$	-	-	-	-	1	2	3	4	5	6	7	8
$E^{fresh}([1,1]) + 4 \times E^{sync}([1,1])$	-	-	-	-	5	6	7	8	9	10	11	12
$E^{sync}([1,2])$	2											
$E^{fresh}([1,2])$	-	-	-	-	-	-	2	3	4	5	6	7
$E^{fresh}([1,2]) + 4 \times E^{sync}([1,2])$	-	-	-	-	-	-	10	11	12	13	14	15
$E^{sync}([1,3])$	5											
$E^{fresh}([1,3])$	-	-	-	-	-	-	-	-	-	5	6	7
$E^{fresh}([1,3]) + 4 \times E^{sync}([1,3])$	-	-	-	-	-	-	-	-	-	25	26	27
$E^{sync}([2,0])$	8											
$E^{fresh}([2,0])$	-	-	-	-	-	-	-	-	8	9	10	11
$E^{fresh}([2,0]) + 4 \times E^{sync}([2,0])$	-	-	-	-	-	-	-	-	40	41	42	43
$E^{sync}([2,1])$	5											
$E^{fresh}([2,1])$	-	-	-	-	-	-	-	-	5	6	7	8
$E^{fresh}([2,1]) + 4 \times E^{sync}([2,1])$	-	-	-	-	-	-	-	-	25	26	27	28
$E^{sync}([2,2])$	2											
$E^{fresh}([2,2])$	-	-	-	-	-	-	-	-	2	3	4	5
$E^{fresh}([2,2]) + 4 \times E^{sync}([2,2])$	-	-	-	-	-	-	-	-	10	11	12	13
$E^{sync}([2,3])$	1											
$E^{fresh}([2,3])$	-	-	-	-	-	-	-	-	-	1	2	3
$E^{fresh}([2,3]) + 4 \times E^{sync}([2,3])$	-	-	-	-	-	-	-	-	-	5	6	7
E^{opt}	0	1	2	3	4	5	6	7	8	5	6	7

表 2 インデックス表の例
Table 2 An example of the index table.

t	0	1	2	3	4	5	6	7	8	9	10	11
t_1	0	0	0	0	0	0	0	0	0	8	8	8
t_2	0	0	0	0	0	0	0	0	0	9	9	9
E^{opt}	0	1	2	3	4	5	6	7	8	5	6	7

表 3 圧縮されたインデックス表の例
Table 3 An example of the condensed index table.

t	0	9
t_1	0	8
t_2	0	9

までに新たに発生するデータ組合せの鮮度と同期度を計算する．しかし，もしある時刻において次の両方の条件を満たすデータ組合せ C が既に存在する場合には，新たに発生したデータ組合せを無視して構わない．

- データ組合せの鮮度 $E^{fresh}(C,t)$ の値が新たに発生したデータを含むデータ組合せの鮮度以下である．
- データ組合せの同期度 $E^{sync}(C)$ の値が新たに発生したデータを含むデータ組合せの同期度以下である．

これら双方の条件を満たす場合には，新たに発生したデータ組合せの評価関数値がデータ組合せ C の評価関数値を下回ることはあり得ない．これにより，表1で次のデータ組合せに関しては除外することができる．

$[0,1]$, $[0,2]$, $[0,3]$, $[1,0]$, $[1,3]$, $[2,0]$, $[2,1]$.

このように作成した中間表に基づき，メディエータは表2に示すようなインデックス表を作成する．この表は各時刻においてどのデータ組合せを用いるかを示している．表2では時刻 $t=0$ において，時刻 $t_1[0]=t_2[0]=0$ で発生する $d_1[0]$ と $d_2[0]$ とから構成されるデータ組合せ $[0,0]$ を用いることを示している．このデータ組合せは，時刻3, 4, 6, 8で新たなデータが発生するにもかかわらず，時刻 $t=9$ まで使用し続けることになる．これはデータ組合せ $[0,0]$ の同期度が0であり，その後で発生するデータ組合せよりも評価関数値が小さくなるためである．しかし時刻9においてデータ組合せ $[0,0]$ よりも小さな評価関数値を与えるデータ組合せ $[2,3]$（$t_1[2]=8$ で発生する $d_1[2]$ と $t_2[3]=9$ で発生する $d_2[3]$）が得られることが表2から分かる．このようにインデックス表によってメディエータはどの時刻で発生したデータ組合せを用いるべきかが分かる．最小公倍数の時刻を経過した後では，メディエータは現在時刻 t の代わりに $(t-\lfloor t/\tau_{1,2}^{LCM}\rfloor\times\tau_{1,2}^{LCM})$ 番目の列を参照する．

メディエータがアプリケーションプログラムから連続問合せを受けると，現在時刻に基づいてインデックス表の列を参照し，データ組合せを選択する．そして問合せ処理を実行して結果をアプリケーションプログラムに返す．時刻 t において，時刻 $t-1$ で採用したデータ組合せよりも最適評価関数値を小さくするデータ組合せが新たに得られた場合には，メディエータは新たなデータ組合わせを用いて再度問合せ処理を実行し，結果をアプリケーションプログラムに返す．アプリケーションプログラムから連続問合せの停止要求を受けるまで，メディエータはこの処理を続ける．

連続問合せに関して，インデックス表をさらに圧縮することができる．例えば表2に示すインデックス表は表3に示すように変換できる．圧縮されたインデックス表は，新たな最適評価関数値を与えるデータ組合せが発生する時刻のみを保持している．また E^{opt} の値を削除している．

連続問合せを実システムに適用するには，システム設計者が E^{sync} の重み w の値を決定するのを支援するユーザインタフェースが必要になると考えられる．

6.3 逐次問合せ

本節では従来の問合せにデータの鮮度と同期度の概念を加えて拡張した，逐次問合せについて検討する．

拡張した逐次問合せにおいては，従来の問合せ文（例えばSQL）に加えて，2個のパラメータ ε と δ を付加する．ε はデータ組合せの同期度 E^{sync} の最大許容値を示し，δ はアプリケーションプログラムが問合せを発行してから結果を得るまでの時間の最大許容値を示す．即ち，拡張問合せは次の形式で与えられる．

$Q(q,\varepsilon,\delta)$

ここで q は従来の問合せ文である．

拡張問合せ $Q(q,\varepsilon,\delta)$ が時刻 t で与えられると，メディエータは次のように問合せ処理を実行する．

アルゴリズム1

(1) 中間表において ε よりも大きな E^{sync} を与えるデータ組合せを無視したうえで，E^{opt} の値を再計算する．

(2) 連続問合せと同様にして，インデックス表を作成する．

(3) 時刻 $(t+\delta)$ から順次時刻を $(\tau_{1,2}^{LCM}-1)$ さかのぼり，メディエータは E^{opt} を与えるデータ組合せを選択する．

(4) もし E^{opt} を与えるデータ組合せが問合せ発行時刻 t で既に得られる場合には，そのデータ組合せを用いてメディエータは問合せ処理を実行し，結果をアプリケーションプログラムに返す．もし問合せ発行時刻以降に E^{opt} を与えるデータ組合せの一部のデータが得られる場合には，それらのデータが得られるまで待つ．そして全てのデータがそろった時点で，メディエータは問合せ処理を実行して結果をアプリケーションプログラムに返す． □

逐次問合せにおいて ε と δ の値をシステムの最終利用者が指定するのは困難と考えられる．従って現実には，ε と δ の値を決定することを支援するユーザインタフェースを用意し，それを用いてシステム管理者が予め指定するなどの手段が必要になろう．

表 4 同期度とデータ取得間隔の比較
Table 4 Comparison of synchronousness and data acquisition intervals.

方法	重み w	同期度の平均値	同期度の最大値	データ取得間隔の平均値	データ取得間隔の最大値	データ取得回数
(a)	-	16	32	100	100	33
(b)	1	9.56	32	60.0	100	55
	5	8.57	24	76.7	100	43
	10	8.36	20	86.8	100	38
	50	8.28	17	97.1	100	34
	100	4.08	16	194	1716	17

7. 適 用 例

ここでは，本論文で述べた連続問合せをアプリケーションに適用する具体例を検討し，本手法の有用性を示す．

データの鮮度と同期度とを考慮したデータ組合せの選択方式が有用になる例として，複数のセンサから得られたデータ組合せを総合的に勘案して意志決定を行う場合が挙げられる．ここでは1節で述べた知的道路交通システム (ITS) に適用する場合について，具体的に検討する．

道路状況を監視して，渋滞中の車両や落下物などの障害物や事故をセンサで検出し，後続車を直ちに減速／停止させて衝突を回避するシステムを構築するために，種々のセンサが開発されている．これらのセンサは文献 10) で指摘されているように，環境変化への頑健性や計測処理の早さや精度などの点で，性能に一長一短がある．そこで多数のセンサを準備して状況に応じて各センサの特長を生かしながら欠点を補完するようにうまく組合せ，センシング性能の改善を図るセンサフュージョン技術の開発が重要である．

道路上の障害物を検出するセンサのひとつとして，カメラと画像処理部が一体になった装置が開発されている．これにより，映像を装置内で処理して障害物や事故を検出し，その結果のみをセンタに伝送することにより，大容量の映像データを伝送する時間を削減することが可能となる．この際，日本の商用カメラでは通常 NTSC 方式が採用されており，29.97 フレーム/秒で画像が取得される．これを周期に換算すると約 33ms/フレームとなり，この結果，センサ出力も周期が約 33ms（あるいはその倍数）に固定される．

このような装置の欠点として，カメラ映像の処理において太陽やヘッドライトによる物体の影を障害物と誤認することが挙げられる．そこでこの欠点を補完する方法として，スキャンレーザレーダを用いて障害物の道路からの高さを検出することが考えられる．現在開発されているスキャンレーザレーダで，データの取得周期が 100ms に固定されているものがある [11]．

これら2種類のセンサを組み合わせることにより物体の影を障害物と誤認することが排除でき，障害物の検出精度を向上させることができる．この際に，2種類のセンサで同時に障害物を捕捉するために，データの同期度が重要となる．さらに，一刻も早く障害物や事故を検出し，後続車を減速／停止させて衝突を回避するために，取得データの鮮度も重要となる．

いま，上記の2種類のセンサから得られるデータ組合わせを選択する方法として，次の2通りを検討する．

(a) スキャンレーザレーダのデータの取得周期に合わせて 100ms 毎に，その時点で最新のデータ組合せを選択する場合．

(b) 本論文で述べた手法を使用する場合．

まず (a) の方法においては，データ取得周期は 100ms に固定される．そして同期度の最大値はカメラから得たデータの頻度で決定され，約 32ms となる．また同期度の平均値は，16ms と計算できる．これを表 4 に示す．

次に (b) の方法について検討する．まず全体的な見通しを得るため，4節で述べた性質を用いて，同期度だけに着目した定量化を行う．スキャンレーザレーダから得た周期データ列を d_1，カメラから得た周期データ列を d_2 とする．そして $\tau_1 = 100, \tau_2 = 33$ とおくと，$\tau_{1,2}^{GCD} = 1$，$\tau_{1,2}^{LCM} = 3300$，$\tau_1' = 100, \tau_2' = 33$ が得られる．$t_1[0] = t_2[0] = 0$ の場合，式 (6) から，同期度の総和は $E_{sum}^{sync}(d_1, d_2) = 272$，さらに式 (4) から同期度平均は $E_{ave}^{sync}(d_1, d_2) \approx 8.24$ と計算できる．また同期度の最大値は，$\tau_{1,2}^{GCD} \times (\tau_2' - 1)/2 = 16$ となる．次に，6.2 節で述べた連続問合せに基づき評価関数の重み w の値を変化させた場合の，同期度の平均値と最大値，データ取得間隔の平均値と最大値，および各周期の最小公倍数までのデータの取得回数を表 4 に示す．ここでは $t_1[0] = t_2[0] = 0$ としている．また同期度の平均値としては，その同期度が利用される時間区間を考慮した加重平均値を示している．

表 4 から分かるように，(b) の方法を用いると，(a) の方法に比較して，同期度の平均値および最大値を低下させることが可能である．これらの値は評価関数に

おける同期度の重み w を増加させることにより，より大幅に低下できる．しかしこれに反して，データ取得間隔の平均値は増加することが分かる．一方データ取得間隔の最大値は，同期度の重み w を 100 に設定した場合においてのみ，1716ms となり，それ以外では常に 100ms となる．実際に同期度の重み w を決定する際には，表 4 に示した各データを考慮して，アプリケーションに適した値を選択する必要がある．

仮に 80km/h で走行中の車両から障害物が落下した場合，その落下物の初期速度は秒速に換算して約 22.2m/s となる．評価関数における同期度の重み w を 50 に設定した場合，表 4 から同期度の最大値は 17ms となることが分かる．その間に，落下物が移動する距離は約 0.377m と計算できる．即ち，これ以上の長さの落下物であれば，カメラとスキャンレーザレーダの両者で同時にその物体を検出することが可能となる．

これに対して（a）の方法による同期度の最大値は 32ms であるため，この間に落下物は，約 2 倍の 0.710m 程度移動する．即ち，本論文で述べた手法を用いることによって，単純に最新のデータ組合わせを用いた場合と比較して同期度の最大値がほぼ半減し，突発的に生じる落下物の検出精度が向上することが期待できる．

8. 議　　論

これまではデータの鮮度がより高い方が望ましいものとして，議論を進めてきた．しかし状況によってはこれが適当でない場合もあり得る．例えばパイプ中を流れる水の流量を A，B の 2 地点で計測する場合を考える．通常は A 地点を通過した水は 60 単位時間後に B 地点に達するとする．水の流速が一定とすると，$d_A[I_A[t-60]] = d_B[I_B[t]]$ が成立する．ここで d_A と d_B は，それぞれ A 地点と B 地点で計測される流量である．A 地点から B 地点に到るまでに水漏れが生じていないかを調べるために，B 地点のデータは，60 単位時間だけ以前に A 地点で計測されたデータと比較する必要がある場合が考えられる．このように現在より一般に l_i だけ以前のデータを使用する必要がある場合には，式 (7) と (8) における $t_i[C[i]]$ に代えて $t_i[C[i]] + l_i$ を用いればよい．

本論文の 6.2 節で述べた連続問合せにおいては，インデックス表を問合せが発行された時点で作成する．これとは別のデータ組合わせの選択方式として，予め同期度と鮮度の最大許容値を指定しておき，これを満たす任意のデータ組合せが発生した時点でそれらを有効なデータ組合わせとして採用する動的な選択方式が考えられる．この方式ではインデックス表を保持するためのメモリが不必要となり，さらに発生するデータの周期に揺らぎが生じても問題がないなどの長所がある．それに対して 6.2 節で述べた選択方式では，一旦インデックス表を作成した後は，表を参照するだけで最適なデータ組合せを高速に決定できるという長所がある．なおこの方式において，長時間が経過してデータの発生時刻がずれてきた場合には，中間表とインデックス表を作成しなおすことで対処可能である．以上の長所と短所を比較して，アプリケーションに適した手法を選択する必要がある．

筆者らはこれまでに周期データ列の同期度に関する性質に関して研究している [7]．この研究では 2 個以上の周期データ列が与えられた際に，それらの中で最大の周期で連続的に問合せ処理を行うことを想定している．そして各周期データ列が任意の初期データ発生時刻を持つ場合の同期度の最悪値，即ち上限値を示している．本論文の 6.3 節で述べた逐次問合せにおいて，ε の値をこの上限値に設定すれば，必ずこれを満たすデータ組合せを得られることが保証される．そしてこの値をそこからさらに減少させ，所望の頻度でデータ組合せを取得できるかを試行錯誤により求めることができる．また同期度の上限値は，6.2 節で述べた連続問合せで評価関数の同期度の重みを決定するための参考となる．

9. おわりに

本論文では，次の事項について述べた．

- 周期データ列からデータ組合せを選択するための基準として，データの鮮度と同期度を提案した．そして，鮮度を優先した場合，同期度を優先した場合，さらに一定時間内に新たにデータが発生するのを待つ場合を検討し，様々なデータ組合せの合理的な選択肢が存在することを示した．
- 2 個の周期データ列に関する基本的な性質を定理 1 で述べた．この定理に基づき同期度の総和と同期度平均という概念を提案した．これらは周期データ列からデータ組合せを選択する際の基準を与えるものである．また定理 2 において，条件によっては 2 個の周期データ列のうち 1 個の初期データ発生時刻をずらすことにより，同期度の総和を周期の最大公約数の半分まで削減できることを示した．
- データの鮮度と同期度に基づいて周期データ列からデータ組合せを選択するための評価関数を提案した．

- 提案した評価関数を用いた連続問合せの枠組みを提案した．この枠組みを用いることにより，最適なデータ組合せが発生するたびに連続的に問合せ結果を得ることができる．またこの方式では最適なデータ組合せが発生する時刻を予め計算しておくため，実行時に最適なデータ組合せを選択するための処理時間を最小限にすることができる．
- さらに提案した評価関数を用いた，逐次問合せの枠組みを提案した．この枠組みにおいては，SQLなどの従来の問合せに加えて，2個のパラメータεとδを指定する．パラメータεは同期度の最大許容値を表し，δは問合せ結果を得るまでの最大許容時間を表す．
- 一部の周期データ列のデータに関して，他の周期データ列より指定された時間だけ以前に発生したデータを用いる必要がある場合でも，提案した問合せの枠組みを容易に拡張して対処できることを示した．
- 本手法を，知的道路交通システムの道路上の障害物の検出に具体的に適用する例を検討した．この場合に，周期的に最新のデータ組合せを用いるデータ選択方式と比較して同期度をほぼ半減でき，本手法が有用であることを示した．

本論文においては，データの鮮度と同期度を考慮した，周期データ列からのデータ選択方式について提案した．ここではラッパにおける問合せ処理時間を考慮していない．データの鮮度と同期度はラッパでのデータ処理時間に依存するため，これらを考慮した問合せ処理の最適化が今後の課題となる．また3個以上の周期データ列からデータを選択・結合する処理方式とその最適化も今後の課題である．さらに，鮮度と同期度に基づいた評価関数を拡張し，データ自身の重要性を考慮することも将来の課題である．

謝辞 数多くの有益なコメントをいただいた査読者の方々に深く感謝の意を表する．

参考文献

1) Liu, C. L. and Layland, J. W. : Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment, *Journal of ACM*, Vol. 20, No. 1, pp. 46–61 (1973).
2) Abbott, R. and Garcia-Molina, H. : Scheduling Real-Time Transactions, *SIGMOD Record*, Vol. 17, No. 1, pp. 71–81 (1988).
3) M. Xiong, R. Sivasankaran, J. A. Stankovic, K. Ramamritham and D. Towsley. : Scheduling Access to Temporal Data in Real-Time Databases, A. Bestavros, K. J. Lin, and S. H. Son, editors, *Real-Time Database Systems Issues and Applications*, Kluwer Academic Publishers, pp. 167 –191 (1997).
4) H. Nakazato and K. J. Lin. : Interval Assignment for Periodic Transactions in Real-Time Database Systems, *Proceedings of the Seventh International Conference on Data Engineering, Kobe, Japan*, pp. 378–385 (1991).
5) F. Kabanza, J. M. Stevennne, and P. Wolper. : Handling Infinite Temporal Data. *Proceedings of the Ninth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Nashville, Tennessee*, pp. 392–403. ACM press, (1990).
6) A. Kurt and M. Ozsoyaglu. : Modeling and Querying Periodic Temporal Databases, *DEXA Workshop*, pp. 124–133 (1995).
7) K. Munakata, M. Yoshikawa and S. Uemura. : On Synchronous Properties of Periodically Generated Data Sequences. *International Symposium on Database Applications in Non-Traditional Environments (DANTE) '99, Kyoto, Japan*, pp. 298–305 (1999).
8) G.Wiederhold. : Mediators in the Architecture of Future Information Systems, *IEEE Computer*, Vol. 25, No. 3, pp. 38–49 (1992).
9) 小泉 寿男. : ITS と情報処理技術, 情報処理, Vol. 40, No. 10, pp. 978–981 (1999).
10) 新川 清, 梶原 康也, 本田 幸弘, 鈴木 尋善. : 自動運転道路システム, 三菱電機技報, Vol. 70, No. 16, pp. 16–21 (1996).
11) 赤須 雅平. : 自動車用小型スキャンレーザレーダ, 三菱電機技報, Vol. 73, No. 10, pp. 60–63 (1999).

(平成11年 9 月20日受付)
(平成11年12月17日採録)

(担当編集委員 角谷和俊)

宗像 浩一（正会員）

1988年大阪大学大学院工学研究科電気工学専攻修士課程修了．同年三菱電機(株)産業システム研究所（当時名称応用機器研究所）入社、現在に至る．この間、1994年から1995年までスタンフォード大学留学、1999年奈良先端科学技術大学院大学情報科学研究科博士後期課程入学．異種情報源の統合に関する研究開発に従事．電子情報通信学会会員．

吉川 正俊（正会員）

1980年京都大学工学部情報工学科卒業．1985年同大学院工学研究科博士後期課程修了．工学博士．同年京都産業大学計算機科学研究所講師．同大学工学部助教授を経て1993年より奈良先端科学技術大学院大学情報科学研究科助教授．XMLデータベース，多次元空間索引，異種情報源の統合などの研究に従事．電子情報通信学会，ACM，IEEE Computer Society 各会員．

植村 俊亮（正会員）

1966年京都大学大学院工学研究科修士課程修了．同年電子技術総合研究所（当時名称電気試験所）入所．1988年東京農工大学教授（工学部数理情報工学科）．1993年奈良先端科学技術大学院大学情報科学研究科教授．工学博士．データベースシステム，自然言語処理，プログラム言語の研究に従事．電子情報通信学会，ACM，IEEE などの会員．