分散リアルタイムネットワーク用プロセッサとその応用

慶應義塾大学
山﨑 信行 yamasaki@ics.keio.ac.jp
産業技術総合研究所
堀　俊夫 t.hori@aist.go.jp

本稿では，まず，本プロジェクトにおける基盤概念となる「リアルタイム性」について述べ，分散リアルタイムシステムを構築するための構成要素となるリアルタイム通信・処理用のプロセッサについて述べる．そして，そのプロセッサを応用した分散リアルタイムネットワークシステムの一例として，分散センサネットワークシステムについて述べる．

◆リアルタイム性

まず，「リアルタイム性」という専門用語の定義について説明を行う．リアルタイムとは，「速い」とか「すぐに」という意味ではなく，また，そういう意味をまったく含まないので注意が必要である．

リアルタイム性[1)]とは，通信や処理等の真偽が時間にも依存するという性質である．狭義には，与えられた時間制約（デッドライン）を守るということを意味する．

通常のシステム（ノンリアルタイムシステム）では，送信元 A から受信先 B にデータが正しく届けば正しい通信システムであるし，演算した結果が正しければ正しい演算システムといえる．ところが，リアルタイムシステムの場合，そこに時間制約という新たな基準が加わる．リアルタイム通信システムの場合，たとえば「送信元 A から受信先 B に 1ms で通信する」という時間制約が付与される．この場合，通信されたデータの中身が正しかったとしても，1ms 以上（たとえば 1.5ms）かかって届いた場合には「偽」となる．演算の場合も同様である．

◆ハードリアルタイムとソフトリアルタイム

リアルタイム性は，以下の 2 つに大別できる．

・ハードリアルタイム
・ソフトリアルタイム

ハードリアルタイム性とは，必ず時間制約を守らなければならない性質であり，時間制約を少しでも破ると価値が 0 になる性質である．狭義には，時間制約を破った場合，システムに損害を与える可能性のあるリアルタイム性のことである．一方，ソフトリアルタイム性とは，時間制約を多少破ることを許容する性質であり，時間制約を破っても価値はただちに 0 にはならない．多くの場合，時間制約を破ると，時間経過とともに価値が減少していく性質を指す．また，時間制約を破っても，システム自身に損害を与えることはない．

ハードリアルタイム性の一例としては，分散制御がある．センサとアクチュエータがネットワークを介して接続されて制御が行われているシステムの場合，センサノードからアクチュエータノードに対して，周期的な時間制約（たとえば 1ms）を満たして，通信および制御のための演算の両方を行うことができないと（つまり，リアルタイム性が損なわれてしまうと），アクチュエータの制御ができなくなってしまう．アクチュエータの挙動としては，リプルが発生したり，最悪の場合，制御不能になって事故を引き起こす可能性もある．

ソフトリアルタイム性の一例としては，VOD（Video on Demand）等のネットワークを介した MPEG のデコードがある．この場合，周期的な時間制約（たとえば 33ms）を満たして，通信およびデコードのための演算の両方を行うことができないと（つまり，リアルタイム性が損なわれてしまうと），正常に動画を視聴すること

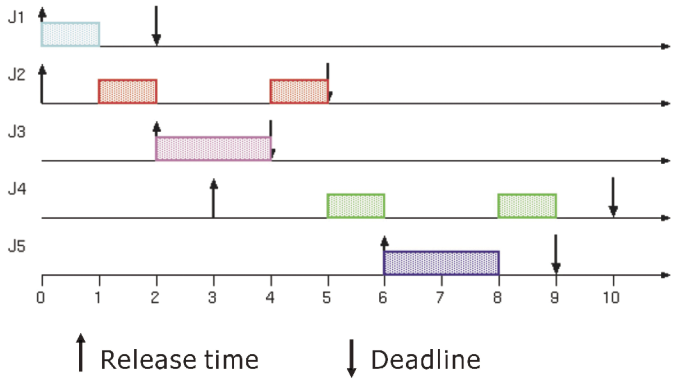

図 -1　EDF スケジューリング

ができなくなってしまう．ただし，リアルタイム性を破る度合いに応じて品質が低下するが，システムにダメージを与えることはない．挙動としては，たとえば，ブロックノイズが発生する，動画と音声の同期が取れない，動画の動きが不自然になる等の影響が発生する．

また，リアルタイム性を要求するアプリケーション（タスク）の種類によって，以下のように特徴が分かれる．

ハードリアルタイム性：主に制御系の通信や演算を行うタスクが要求するリアルタイム性であり，以下のような特徴がある．

・通信：データ量は小さいが，レイテンシ（遅延）に対する要求が厳しい．スループットよりレイテンシを重視する．
・演算：演算量は小さい場合が多いが，時間制約を厳守する必要がある．
・時間粒度とデッドライン：時間粒度が小さく，デッドラインが短い場合が多い（100μs ～ 10ms 程度）．

ソフトリアルタイム性：主にマルチメディア系の通信や演算を行うタスクが要求するリアルタイム性であり，以下のような特徴がある．

・通信：データ量が非常に大きく（ストリーミング等），レイテンシよりもスループットを重視する．
・演算：演算量はそれなりに大きい（MPEG のデコード等）が，時間制約は制御系に比較すれば厳しくない．
・時間粒度とデッドライン：時間粒度が比較的大きく，デッドラインが長い場合が多い（10ms ～ 1s 程度）．

このように，同じリアルタイム性といっても，ソフトリアルタイムとハードリアルタイムでは，リアルタイム性を要求するアプリケーションも異なるし，特徴も異なることが分かる．我々は，これら性質の異なるリアルタイム性を同時に扱うことのできるリアルタイム通信・処理プロセッサの研究開発を行っている．

◆リアルタイムスケジューリング

複雑な環境でリアルタイム処理を行うためには，リアルタイムスケジューラを用いたスケジューリングが必須である．たとえば，いわゆる従来の組込みプログラミングにおいては，すべての起こり得る状況が分かっている場合がほとんどであり，リアルタイムスケジューラを用いなくても，プログラマがすべての場合を想定してプログラミングを行えば，組込みプログラム（リアルタイムプログラム）を作成することが可能であった．

それに対して，ネットワークで相互に接続されたリアルタイムシステムである分散リアルタイムネットワークシステムにおいては，すべての起こり得る場合を想定することは不可能であり，リアルタイムスケジューリング（リアルタイムスケジューラ，アドミッションコントロール等）が必須となる．

リアルタイムスケジューリングアルゴリズムには EDF（Earliest Deadline First），RM（Rate Monotonic）等があり，デッドラインや周期等の条件からスケジューリングされ優先度を付与されたリアルタイムタスクは，ある一定時間間隔（ティック）ごとにプリエンプティブに実行される [2]．

図 -1 に EDF（Earliest Deadline First）によるスケジューリングの例を示す．EDF では，図 -1 に示すように，デッドラインが近いタスクほど高い優先度を与えてスケジューリングを行う．ここで，プリエンプション（横取り）とは，現在行っている処理を中断し，別の処理を行うことである．中断された処理は後で再開可能なように，必要十分なデータ等をバックアップしておく必要がある．

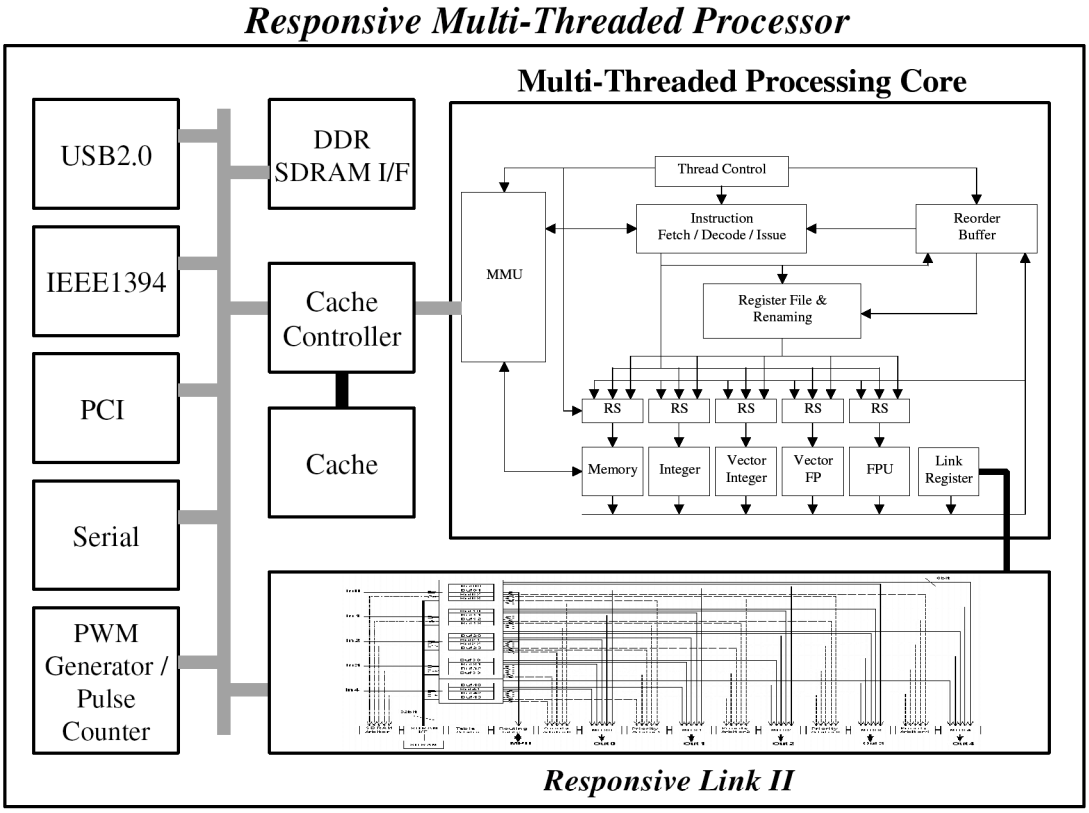

図 -2　RMT プロセッサのブロック図

プリエンプションは, 演算の場合, コンテキストスイッチに相当し，通信の場合，パケットの追い越しに相当する．したがって，演算に関してはコンテキストスイッチを，通信に関してはパケットの追越しを最適に行うリアルタイム通信・処理アーキテクチャを実現することが目標である．

我々が開発しているプロセッサ（ハードウェア）は，基本的にこれらのリアルタイムスケジューラ（ソフトウェア）でスケジューリングされ，優先度が付与された演算や通信を最適に（つまりリアルタイムに）実行することを目的としている．

◆リアルタイム通信・処理プロセッサ

並列分散リアルタイム処理用途に研究開発を行った Responsive Multi-Threaded Processor（以下，RMT プロセッサ）および RMT プロセッサに搭載されているリアルタイムネットワーク規格 Responsive Link（以下，レスポンシブリンク）について述べる．

RMT プロセッサは，分散リアルタイムネットワークシステムを実現するために，リアルタイム通信およびリアルタイム処理を同時にハードウェアレベルで行うことを目的にして設計された世界初のシステムオンチップである．RMT プロセッサは，ハードリアルタイム通信・処理およびソフトリアルタイム通信・処理の両方に対応する．

分散リアルタイムシステムを容易かつ効率的に実現するには，リアルタイム通信およびリアルタイム処理を行うための基本機能を有した共通プラットホームを用意し，それらをブロックを組み立てるように組み合わせてシステムを構築できるようにすればよい．プラットホームに必要な機能としては，リアルタイム演算機能，リアルタイム通信機能，コンピュータ用周辺機能，各種周辺制御機能が考えられる．プラットホームとしてさまざまなシステムの中に容易に組み込んで使用できるようにするために, RMT プロセッサは以下の機能をすべて 1 チップに集積（System-on-a-chip）している（**図 -2** 参照）．

・リアルタイム演算機能（RMT プロセッシングコア）
・リアルタイム通信機能（レスポンシブリンク）
・コンピュータ用周辺機能（PCI64, USB2.0, IEEE1394,

DDR SDRAM I/F, DMAC 等）
・各種周辺制御機能（PWM 発生器，パルスカウンタ等）

システム設計者は本チップに必要な I/O（センサ，アクチュエータ，ディジタルビデオカメラ等）を接続するだけで，必要な機能を実現できる．それら I/O を接続し，固有の機能を有した RMT プロセッサをそのシステムにふさわしいトポロジでレスポンシブリンクを用いて複数個接続することによって，分散リアルタイムシステムを構築することが可能となる．

また，RMT プロセッサは，TSMC の協力により，以下のような最先端プロセスで設計・実装されている．

・製造メーカ：TSMC
・プロセスルール：0.13μm CMOS 8 層 Cu 配線
・ゲート数：約 10M ゲート
・動作電圧：
　・Core：1.0V
　・I/O：2.5V
・ダイサイズ：10.0mm × 10.0mm
・チップサイズ：4.5cm × 4.5cm BGA

◆レスポンシブリンク

レスポンシブリンク[3]のリアルタイム通信アーキテクチャは，分散リアルタイムシステム用であることを十分に考慮して設計されており，以下のようなユニークな特徴を有している．特に最初の 5 項目は世界初の実装である．

・ハードリアルタイム通信（イベント）とソフトリアルタイム通信（データ）の分離
・パケットに優先度を付加し，ノードごとに高優先度パケットが低優先度パケットの追越しを行う
・パケットの優先度が異なると，優先度ごとに別経路を設定することが可能で，専用回線や迂回路を容易に実現可能
・ノードごとに優先度を付け替えることが可能であり，分散管理型でパケットの加減速を制御可能
・256 レベルの優先度
・ハードウェアによる前方エラー訂正（FEC）
・通信速度を動的に変更可能
・Plug & Play 機構

これらの特徴的な機能によって，柔軟なリアルタイム通信を実現している．

現在のレスポンシブリンクの一方向の最大通信速度は 1Gbps である．通信速度は消費電力を抑えるために動作中に動的に変更可能である．

RMT プロセッサには，5 組（5 × 5 のネットワークスイッチ）のレスポンシブリンクが設計･実装されている．そのうち 1 組はプロセッサに接続し，4 組がチップ外部に出ている．全二重でデータとイベントの 2 系統を 1 組（1 本）として，1 チップ上にはそれが 5 組あるので，バックボーンとしては約 20Gbps/chip のスイッチング速度を有している．

レスポンシブリンクは，現在，ISO/IEC JTC1 SC 25 において，国際標準化作業を行っている．また，今年度から情報処理学会試行標準としても標準化作業を行う．これらの標準化が実現されると，異なるシステム間でのリアルタイム通信も可能になり，より大規模な分散制御システムが実現可能になると考えられる．

◆ RMT プロセッシングコア

RMT プロセッサの演算部（いわゆる MPU）である RMT プロセッシングコアのリアルタイム処理アーキテクチャは，ハードウェアでさまざまなレベルのリアルタイム処理をサポートしている．以下のような特徴を有している．

・優先度付細粒度マルチスレッディング
　・8 スレッド同時実行
　・32 スレッド格納可能なコンテキストキャッシュ
　・ハードウェアによるコンテキストスイッチ
　・256 レベルの優先度
　・優先度を用いたすべてのファンクショナルユニットの制御
　・割込み発生時のハードウェアによるスレッドの制御
・高性能ベクトルプロセッサ
　・柔軟な複合演算のサポート
　・複数スレッドによるベクトルユニットの共有

これらの特徴的な機能によって，柔軟なリアルタイム処理を実現している．リアルタイム処理（QoS）を実現するためにシリコンエリアの多くを使用しているが，普通の MPU としても演算能力はかなり高く，ピークパフォーマンスは以下の通りである．マルチスレッディング機構により，下記の性能を同時に引き出すことが可能である．

・Scalar 32bit Integer　1.2GIPS
・Scalar 64bit Floating Point　600MFLOPS

図 -3　SELF の外観（右が寝室，左がリビング・ルーム）

・Vector 32bit Integer	9.6GIPS
・Vector 64bit Floating Point	4.8GFLOPS
・消費電力	Max. 8W

◆開発環境

現在，RMT プロセッサは開発環境（基板，クロス環境，RT-OS 等）の研究開発を行っているので，すぐに応用することはできない．そこで，RMT プロセッサの開発環境が整うまでは，1 つ前のバージョンである Responsive Processor[4] を用いて，分散リアルタイムネットワークシステムのプロトタイプを構築している．レスポンシブプロセッサは RMT プロセッサと同様な機能を有したシステムオンチップである．レスポンシブプロセッサのプロセッシングコアは通常のシングルパイプラインの RISC プロセッサ（SPARC）であり，リアルタイム処理をハードウェアレベルでサポートすることはできないが，従来どおりの RT-OS を用いて処理することは可能である．レスポンシブプロセッサは，分散リアルタイムネットワークを構築する際に最も重要なリアルタイム通信機能（レスポンシブリンク）を有しているので，プロトタイプ構築には十分使用できると考えられる．

レスポンシブプロセッサの開発環境は，GNU ベース（gcc, gas, gdb 等）であり，ほぼすべてのプラットホーム（Linux, FreeBSD, Windows, Solaris 等）に移植されている．開発者は，ボードと PC を PCI，RS-232C，USB 等で接続し，gdb ベースのデバッグ環境で開発を行うことができる．

以下では，プロセッサを応用した分散リアルタイムネットワークシステムの一例として，分散センサネットワークシステムを紹介する．

◆分散センサネットワーク

「センサネットワーク」は，近年よく聞かれるようになった言葉である．そのまま解釈すれば，これは「多数のセンサで構成されるネットワーク」という漠然としたものであり，実際，それが対象とする領域は非常に広い．たとえば，Intel 社は，MEMS（Microelectromechanical Systems）技術を利用してセンサ，演算機能，通信機能を 1 チップに納めた「Smart Dust」を開発している．その狙いは，これを空間にばらまいてアドホックなセンサネットワークを構成することにある．一方，一般的なサイズのセンサを利用した例としては，ITS（Intelligent Transport Systems）の実現に向けて整備が進みつつある交通管制用の情報収集システムがある．いずれも，観測可能な範囲に限界があるセンサを多数配置して互いを補い，全体として広域の情報を収集することがその主たる目的であり，ユビキタス知的情報インフラの実現形態の 1 つと考えることができる．そして，環境モニタリングや交通管制以外にも，防犯，マーケティング・リサーチ，医療・介護・福祉など，応用範囲も非常に幅広く，今後の発展が期待される研究分野でもある．

このような技術的・社会的背景の下，RMT プロセッサが提供する演算性能やさまざまな入出力機能を活用するシステムの例として，我々は現在，人間の行動を観察するための分散センサネットワークを開発している．以下ではこのシステムを紹介する．

◆センサ化環境 SELF

現在，我々は SELF という名のシステムを開発している[5]．SELF は，Sensorized Environment for LiFe の略

であり，その名の通り，人間の生活を支援するためのセンサ化環境を意味する．SELF では，生活環境に多数のセンサを埋め込んで分散センサネットワークを構成し，そこで活動する人間の生理状態や行動を観察すると同時に，人間を支援することに重点を置いている．

これらの技術は，前述の防犯，マーケティング・リサーチ，医療などへの応用を想定する上で非常に有意義である．また，人間の行動をモデル化することそれ自体が，我々が関係しているデジタルヒューマン研究[6]の行動モデル構築に重要な役割を果たす．なお，本稿は分散センサネットワークについて述べるため，デジタルヒューマン研究の詳細については文献 6）等を参照されたい．

現在の SELF は，寝室とリビングルームの 2 部屋で構成されている（**図 -3**）．寝室は，主に睡眠時の生理情報の取得・蓄積を目的としている．ただし，この部屋のデータ処理や通信は，RMT プロセッサが提供するような非常に時間粒度の小さいリアルタイム性が必要ないため，これ以上の解説は省略する．

一方，リビングルームは，人間の行動を観察し，行動情報を蓄積すると同時に人間を支援することを目的としている．通常，観察を目的とするシステムでは，ビジョンを使ったモニタリング手法が用いられることが多く，この場合，非常に精度よく人間の位置と動きを検出できる．しかし，ここで得られる情報は，人間の位置や動き（移動）といった抽象レベルの情報であり，システムの目的である行動，たとえば「イスに座る」「食事をとる」「寝る」といった情報を直接得ることはできない．そこで，我々は，室内の各物体にタグを付け，人間の行動に伴う物体の位置変化を計測することで，人間の行動情報を取得する手法を選択した．この利点は，抽象的な人間の行動を物体の移動という具体的な事象にマッピングでき，その後の処理（解釈）が容易になるというところにある．

人間が日常扱う物体は，小さいものでおよそ 2 ～ 3cm であるから，我々の目的を達成するためには，数 cm の計測精度と 1 ～ 2cm 程度の分解能が要求される．物体へのタグ付けと位置計測が可能なデバイスには，コンタクトレス IC や IR タグ，ポヒマスセンサ等があるが，これらはいずれも，計測精度が悪い，あるいは測定範囲が狭いという点で，我々の用途には不向きである．そこで，SELF では超音波発信器と超音波センサを採用した．具体的には，個々の超音波発信器に固有の ID を与えて物体に取り付け，また，多数の超音波センサを環境（天井や壁面）に設置する．発信器の ID とそれが取りつけられている物体は既知であるから，その位置を逐次計測することにより「いつ，どの物体が，どこからどこまで，どのような経路で移動したか」を知ることができるシステムである．

現在，リビングルームには，天井と壁面（三方）におよそ 30cm の間隔で 308 個の超音波センサが埋め込まれている．また，固有の ID を付与された超音波発信器が 20 ～ 30 個，室内のさまざまな物体（イスやコップ，ステープラ，ゴミ箱等）に取りつけられている．つまり，部屋全体が，超音波の各センサへの到達時間差から発信器の 3 次元位置を求める分散センサネットワークシステムとなっている．

以下に述べるように，この部屋のセンサ系には μs 程度での同期およびセンサデータのリアルタイム並列処理が要求されるため，ここに RMT プロセッサを活用する．なお，現在はまだ RMT プロセッサとその利用環境が整っていないため，以下で述べる SELF のシステム構成は，前バージョンのレスポンシブプロセッサを利用したものである．

◆分散センサネットワークを支えるリアルタイム通信機能

さて，超音波を利用した距離測定では，発信器とセンサとの時刻同期の精度が計測精度に影響を及ぼす．たとえば，音速を毎秒 340m とすると，超音波の発射時刻とセンサの計時開始時刻が 3μs ずれるごとに約 1mm の計測誤差が生じる．一方，前述のように，SELF では 1 ～ 2cm 程度の精度と分解能が必要である．したがって，目標精度を仮に 1cm とすれば，これは，発信器が超音波を発射してから 30μs 以内にすべてのセンサが計時を開始しなければならないことを意味する．現在利用可能な OS やネットワーク機器では，予測不可能なパケットのバッファリング等が発生するため，このデッドラインを常に保証することは難しい．

次に，個々の発信器を駆動する時間間隔は以下のように決定している．部屋の大きさを考慮すれば，発信器とセンサとの距離は部屋の対角線（約 6.3m）以下であることが保証される．また，センサが発信器からの直接波と反射波の両方を観測する場合，直接波が必ず先にセンサに到達することを考慮すれば，すべての直接波は必ず 18.43ms（= 6.3m ／ 340m/s）以内に観測されるはずである．そこで，SELF では 50Hz で発信器を駆動するよ

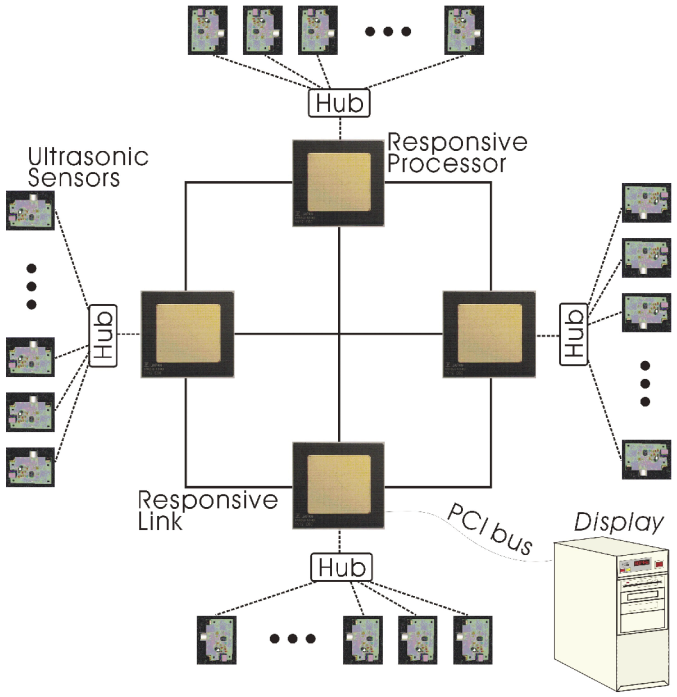

図 -4　システム構成概念図

うに決定した．以上から，このセンサ系は，50Hz で動作し，30μs のデッドラインを持つ周期的なハードリアルタイムシステムであるといえる．

SELF ではこれを**図 -4** に示すように，天井および壁面（3 面）に 1 つずつ，計 4 個のレスポンシブプロセッサを配置して，その面の超音波センサで計時を開始させ，同時にそのデータを処理させている．

具体的には，1 つのレスポンシブプロセッサをマスタノードとして 50Hz で超音波発信器を駆動させており，同期信号をレスポンシブリンク（イベントリンク）を利用して他の 3 プロセッサに通知することで，30μs 以内にすべてのセンサの計時を開始させている．また，センサデータは各センサにつき 16bit のカウンタ値であり，この値が発信器とセンサとの距離（cm）を示す．各プロセッサは自らが担当する面の 70 ～ 80 個のセンサデータを入力し，レスポンシブリンク（データリンク）を利用してマスタノードに送信する．

このセンサデータは，マスタノードのレスポンシブプロセッサから PCI バスを介して外部の計算機（Pentium III）に転送されて記録されている．このデータは，**図 -5** のような 3 次元位置ブラウザで動作中や記録後に確認できる．

現在，個々のレスポンシブプロセッサでは，距離データを生データのまま送信しているが，今後は，後述のロバスト位置推定アルゴリズムを利用して，各プロセッサで位置推定まで完了させることを予定している．

なお，現在の構成（プロセッサ 4 個）では，すべてのレスポンシブプロセッサ間をフルメッシュ接続できるため，パケットはすべて直接マスタノードから各プロセッサに送信しており，パケットの優先度とルーティングについては特に工夫していない．今後，利用するプロセッサ数が増加した際には，これらを検討する必要がある．

ここまでは，システムのハードウェア要件（リアルタイム通信）から決定される仕様に基づいて，SELF の分散センサネットワークに RMT プロセッサ（レスポンシブプロセッサ）を利用する必要性を述べ，その設計・実装を示した．以下ではデータ処理（ソフトウェア）の観点から，リアルタイム並列処理の有用性を示す．ただし，こちらはまだ実装は完了していないため，その構想を述べるにとどめる．

◆マルチスレッド処理機能を利用した精度向上の試み

SELF のリビングルームに埋め込まれた 300 個以上の超音波センサは，すべての位置が既知であり，理論的には，これらの中から直線上にない 3 個のセンサを選び，それぞれから発信器までの距離が計測できれば，三角測

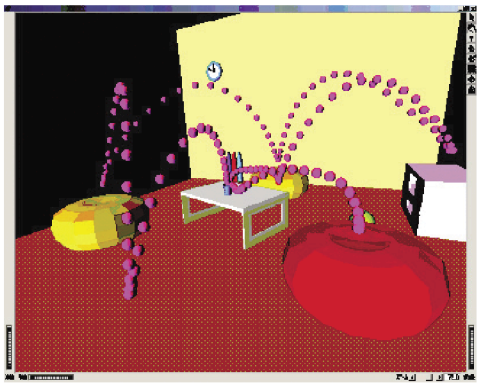

図 -5　3 次元位置ブラウザ（物体の移動履歴を表示）

量の原理により発信器の 3 次元位置が決定できる．

しかし，実際には，1）発信器の指向性により音波を観測できないセンサがある，2）人間の身体や他の物体で隠れ（occlusion）が生じ，音波を観測できないセンサがある，3）壁面や床面で発生した反射波を観測するセンサがある等の理由で利用できないデータも多数存在する．そして，最小二乗法のように，すべてのデータを公平に利用する手法では，解がこのような外れ値（outlier）に大きく影響を受ける．

SELF では外れ値の影響を最小化するよう，パラメータ推定に RANSAC（Random Sample Consensus）法を利用し，LMedS（Least Median of Squares）法によるロバスト推定で発信器の位置を求めている．具体的には以下の方法により発信器の位置を推定する [7]．

1．すべてのセンサのうち，データが観測できたセンサを n 個とする
2．n 個から直線上にない 3 個をランダムに選び，3 個の球面の方程式から発信器の位置を推定する
3．残りの（n − 3）個のセンサと推定位置（x, y, z）との距離を計算し，計測データとの差を求めて，その中央値を求める
4．2. ～ 3. を規定回数繰り返した後，最小の中央値を与えた推定位置を最適解として採用する

前述の通り，SELF の超音波発信器は 50Hz で駆動されるため，これは 20ms のデッドラインを持つソフトリアルタイムタスクである．このデッドライン自体はそれほど高速なプロセッサでなくとも十分に満足できるが，最終的な推定精度は繰り返し回数とのトレードオフの関係にある．一方，2. ～ 3. の処理は明らかに並列実行が可能であり，RMT プロセッサを利用してタスクの並列度が上げられれば，繰り返し回数を増加でき，すなわち，位置推定精度の向上が期待できる．

◆今後の展望

本稿では，分散リアルタイムネットワーク用プロセッサと，それを応用した分散センサネットワークについて述べてきた．今後，RMT プロセッサが知的社会インフラを支える基盤技術の 1 つとなることを目指す上で，そこで使用している要素技術の標準化が重要な意味を持つ．これについては，ISO 等で進められている標準化を一層押し進めていく必要がある．一方，プロセッサを利用する立場からすると，現在の開発環境は十分に使いやすいとは言い難い．RMT プロセッサ導入の敷居を低くするためにも，その性能を有効に活用できる開発環境，RT-OS，ミドルウェア等の研究開発を行っていく予定である．

一方，現在開発中の SELF（分散センサネットワーク）では，まだレスポンシブプロセッサが提供する性能のほんの一部しか使えていない．実際，このプロセッサを使う「旨み」は，リアルタイム通信をサポートするネットワークでセンサ系とアクチュエータ系を結合できるところにある．今後は，人間への物理的な支援の統合をも念頭に，開発を続けていく予定である．

謝辞　本稿で紹介した研究の一部は，文部科学省の科学技術振興調整費の支援による．また，RMT プロセッサの研究の一部は，科学技術振興事業団 SORST の支援による．

参考文献

1）Stankovic, J. A.: Misconceptions about Real-Time Computing, IEEE Computer, pp.2-10（1988）.
2）Liu, C. L. and Layland, J. W.: Scheduling Algorithms for Multiprograming in a Hard-Real Environment, Journal of the ACM, Vol.20, No.1, pp.40-61（1973）.
3）Yamasaki, N.: Responsive Processor for Parallel/Distributed Real-Time Control, 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.1238-1244（2001）.
4）山崎信行，松井俊浩：並列分散リアルタイム制御用レスポンシブプロセッサ，日本ロボット学会誌，Vol.19, No.3, pp.68-77（2001）.
5）Hori, T., Nishida, Y., Suehiro, T. and Hirai, S.: Development of a Networked and Sensorized Environment, 2000 IEEE International Conference on Systems, Man, and Cybernetics, pp.983-988（2000）.
6）独立行政法人産業技術総合研究所サイバーアシスト研究センター，デジタルヒューマン研究ラボ（編）：デジタル・サイバー・リアル－人間中心の情報技術－，丸善，東京（2002）.
7）西田佳史，相澤洋志，堀　俊夫，柿倉正義：超音波式 3 次元タグを用いた人の日常活動の頑健な計測～冗長なセンサ情報に基づくロバスト位置推定～，第 20 回日本ロボット学会学術講演会予稿集（CD-ROM），3C18（2002）.

（平成 14 年 12 月 11 日受付）