

# 艦載用マルチプロセッシングシステム

藤浪清久 川添博道 中島務

三菱電機株式会社 鎌倉製作所

艦載情報処理装置の構成と、課題について述べる。この装置はセンサーデータから目標情報を自動抽出し、不要信号除去、目標追尾といった一連の処理を実施するものである。装置はこれらの機能をリアルタイムに処理する必要性からマルチプロセッサ構成とした。また、装置上のどのプロセッサに処理を割り当てるかを定めるスケジューリング方法も開発した。

## Multiprocessing System on Shipboard

Kiyohisa Fujinami Hiromichi Kawazoe Tsutomu Nakajima  
MITSUBISHI ELECTRIC CORPORATION KAMAKURA WORKS

A structure and problems of a data processing system on shipboard are discussed. The data processing system extracts target information automatically from sensor data, removes noise optimally from signal and tracks targets suitably for a purpose. We adopted a multi-processor structure to implement these functions in a real-time processing mode. A processor assignment algorithm on the multi-processor system is also developed.

## 1.はじめに

近年における電子部品の高性能化、信号処理技術の高度化は艦船搭載用電子装備品にも押し寄せており、現在開発又は計画されている電子装備品の大半はマイクロプロセッサを多数使用し、数メガバイトものプログラムを搭載している。

艦船用情報処理装置はその中でもこの影響が特に大きいシステムの1つである。このシステムで使用されているこれらのマイクロプロセッサはセンサーが受信した信号をミリ秒単位で処理し、その結果をデータベースに登録し、他システムと情報交換するための管理を行っている。更に多機能化、高機能化した複数センサーを制御するための処理も並行して行う必要がある。

その結果、現在の艦載情報処理装置は過去の装置と比べて以下のような特徴を有している。

### (1)多目標同時処理

同時に複数の目標処理を行う。

### (2)自動処理

一人のオペレータが一度に複数の目標について対応するのは困難なので、処理の大半を自動化する。

### (3)多機能化

目標の種類に応じて信号処理の内容を変更する。

## 2.艦載情報処理装置

ここでは、艦載情報処理装置に要求される機能について述べる。この情報処理装置が果たすべき機能範囲は、センサーによって目標が検出された後、その目標に対する情報を他のシステムにネットワーク経由で出力するまでに行われるあらゆる目標情報に関する一連の処理として定義される。その主機能は目標関連機能、目標検出機能と目標追尾機能である。その他付随する機能としてユーザインタフェース処理等も重要となる。図1に情報処理装置の機能ブロック図を示す。

### (1)目標関連機能

センサーから検出結果が得られた時刻を $t_{k+1}$ とする。時刻 $t_k$ 時の既追尾目標の位置及び速度を元に時刻 $t_{k+1}$ 時の予測位置を算出し、検出結果がどの追尾目標に相当するのかが相関判定を行う。相関有り判定された検出結果は(3)の目標追尾機能に渡され、一方どの目標とも相関の得られなかった検出結果は次の目標検出機能に渡される。

### (2)目標検出機能

目標検出機能はセンサーからの検出結果に対し、目標外信号を排除する役割を持っている。この機能は従来、オペレータが誤追尾目標を手動で排除していたものと等価なものであり、自動化には不可欠な機能といえる。

この誤追尾目標排除機能は、過去の履歴情報を参照しているため、追尾時間の経過と共に必要とされる計算機資源が指数関数的に増大する。そのため、実際には計算機資源があまり増大しないような工夫をしている。本報告では、この目標検出機能と次の目標追尾機能を合わせて多目標同時処理と呼ぶ。

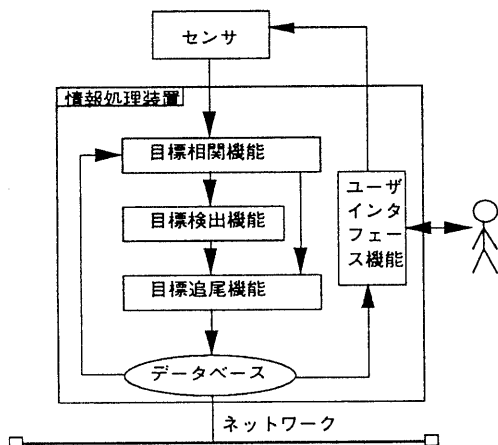


図1 情報処理装置機能ブロック図

### (3)目標追尾機能

センサーが受信した信号からノイズを除去して目標信号だけを取りだし、目標の位置及び速度を算出することである。本報告では以下フィルタリングと呼び、一般にはここに多くの数値演算処理が必要とされる。

### (4)ユーザインタフェース機能

オペレータが入力する各種操作を解読し、オペレータが要求している処理を実行する。また、オペレータに必要な情報を抽出してオペレータに伝達する機能である。

## 3.マルチプロセッシング方式の検討

2で述べた諸機能を実現するために、次のような項目を検討した。

### (1)高速数値演算プロセッサの検討

フィルタリングの処理では、算出する目標の位置及び速度の精度を上げるために64ビット浮動小数点を使用して各演算を行っており、行列どししの乗算、逆行列の算出等浮動小数点に対する四則演算処理が多くある。

この演算処理を通常のCISCプロセッサを用いて実行すると、要求されている処理時間を満たすことができない。そのため、浮動小数点演算を高速に実行できる高速数値演算プロセッサの導入を検討した。

### (2)数十台のプロセッサ搭載の検討

本情報処理装置では同時に数十の目標に対してフィルタリングを行なう必要がある。この処理を一台のプロセッサで要求されている時間内に実施することは不可能である。そこで、フィルタリング機能を数十台のプロセッサに搭載し、各々の目標に対するフィルタリング処理を各プロセッサに分散させることにより要求されている処理時間を満足させることを検討した。

### (3)目標の動的フィルタ割当の検討

フィルタリングを行なうプロセッサを十分多く搭載できれば1つの目標に対してどんなときでも決まったプロセッサを割り当てることができる。

(これを以下静的フィルタ割当と呼ぶ。)しかし、大量のプロセッサを搭載することはコストの面から、また装置の物理的な大きさの面から限界が存在する。さらにフィルタリングの処理間隔が状況に応じて変更されるため、処理負荷も動的に変わる。したがって、上記(2)で述べたような単純なプロセッサ数の増加だけで対処するわけにはいかない。

限られたプロセッサ数で、要求されている処理時間内に要求されている目標数の目標に対してフィルタリング処理を実行するために、各プロセッサの負荷状況を参照し、各々の目標をどのプロセッサでフィルタリングを行うかをリアルタイムに決める動的フィルタ割当方式を検討した。その詳細を以下に示す。

#### a. 動的フィルタ割当アルゴリズムの開発

本情報処理装置に要求されている目標数のフィルタリング処理に必要なとされる計算を、限られたハードウェア資源で行なうには、目標を高速かつ最適にプロセッサに割り当てるアルゴリズムが必要である。そのために、各プロセッサから現在フィルタリング処理を行なっている目標数はいくつか、またその要求処理時間はどのくらいか等の情報を収集し、高速、かつ最適に各々の目標をフィルタリング処理を行なうプロセッサに割り当てるアルゴリズムの開発を行なった。

図3-1に示すように、目標1から目標4をプロセッサ1及びプロセッサ2にそれぞれ静的に割り当てたとする。このときの、センサーからの目標の入力順番は目標1、2、3、4の順である。図3-2のように時刻 $t+k$ において、センサーからの目標の入力順番が目標1、3、2、4と変化したとする。このとき、静的フィルタ割当を行なうと図3-2のように目標4のフィルタリング処理が要求時刻をオーバーしてしまう。これを、

図3-3に示すように動的フィルタ割当方式を用いると、目標3が入力された時に、プロセッサの負荷を調べ、負荷が低いプロセッサ2に対して目標3を割り当てることができる。次の入力目標2は負荷の低いプロセッサ1へ、その次の目標4はプロセッサ2へ割り当てることができる。これによりすべての目標に対して要求されている時刻内にフィルタリング処理が実行される。

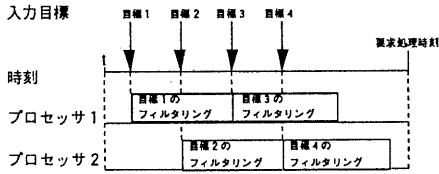


図3-1 静的フィルタ割当 (時刻)

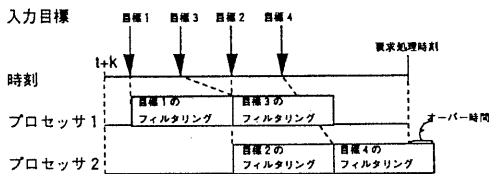


図3-2 静的フィルタ割当 (時刻+k)

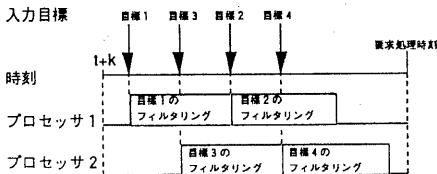


図3-3 動的フィルタ割当 (時刻+k)

しかし、動的フィルタ割当方式は目標に対してフィルタリングを行なうプロセッサが刻々と変わるためシステム開発上、ソフトウェアのデバッグ、試験の面からは困難さが増大する。そのため、情報処理装置で扱っている目標数が少なくフィルタリング処理を行なうプロセッサの負荷が低いときはこの方式を使用せず、負荷が高くなったときのみ使用したほうがよい。その処理の流れを図3-4に示す。

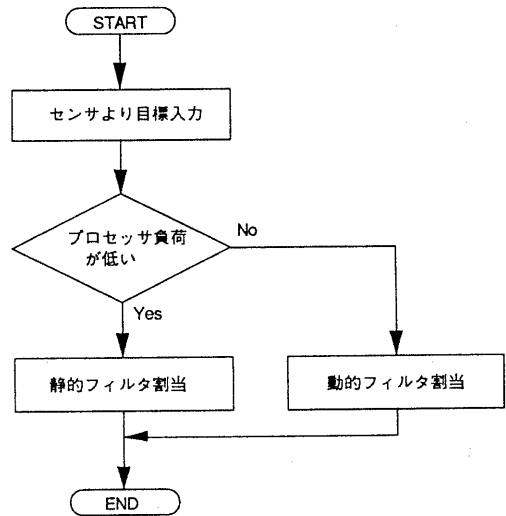


図3-4 処理フロー

b. フィルタリングに必要な内部データを共有するための共有メモリの搭載

フィルタリング処理において、実際に計算を行なうためには前回計算した結果（これを以下内部データと呼ぶ。）を利用する必要があり、次の計算を開始するまでその内部データを保存しておく必要がある。

図3-5に示すように時刻 $t$ で目標1に対するフィルタリング処理がプロセッサ1上で行なわれていたとする。次の時刻 $t+k$ ではプロセッサ1の処理負荷が高く、目標1のフィルタリング処理がプロセッサ1で不可能になり、負荷の軽いプロセッサ $n$ にてフィルタリング処理を行うとする。(図3-6参照)

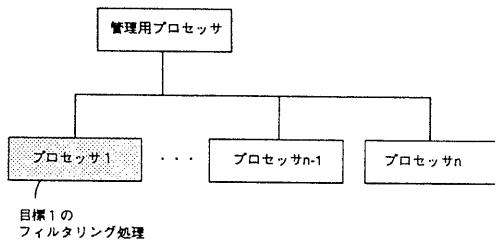


図3-5 フィルタリング処理 (時刻 $t$ )

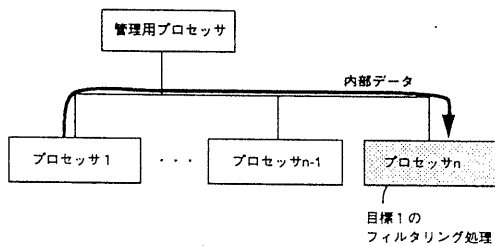


図3-6 フィルタリング処理 (時刻 $t+k$ )

この場合、もし目標1の内部データをプロセッサ1上に保存しておいたなら、時刻 $t+k$ でプロセッサ1からプロセッサ $n$ へ目標1の内部データを移動させる必要がある。

しかし、プロセッサ1においては目標1のフィルタリング処理が終了したあと他の目標のフィルタリング処理が動作する場合が有り、目標1の内部データをプロセッサ1に保存しておくことは難しい。そのため、図3-7に示すように各プロセ

ッサがアクセス可能な共有メモリを設けてフィルタリング処理が終了するごとにその内部データを共有メモリに保存し、次の割当が行なわれたときはその共有メモリから内部データを指定のプロセッサに転送すればよい。この場合、共有メモリではその内部データを目標ごとに管理する必要がある。

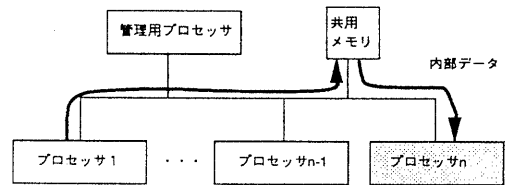


図3-7 共有メモリの搭載

### c. 目標のプロセッサ割当を行なう管理プロセッサの搭載

以下に示す理由から管理用のプロセッサを搭載する必要があると考えた。

#### (ア) 目標のプロセッサ割当

現在どのプロセッサ上でどの目標に対するフィルタリング処理が行なわれているのかというプロセッサの割当や、またそのフィルタリング処理に対する処理時間要求等を1箇所のプロセッサで管理したほうが目標をプロセッサに割当やすい。

#### (イ) 目標どうしの相関

2.(1)で述べた要求機能を実現させるためには現在追尾している目標のフィルタリング結果がすべて必要である。そのためには、1箇所でそのデータを集中して管理し、相関処理を行なう必要がある。

#### 4. H/W構成と制御方式

3で述べた検討結果から図4に示すシステムを社内で作成し評価を実施した。本情報処理装置では、1本の内部バスに目標のプロセッサ割当を行なう管理用プロセッサ（CISCプロセッサ）、内部データを保存するための共有メモリ及びフィルタリング処理機能を搭載する高速数値演算プロセッサが接続されている。<sup>1)</sup>

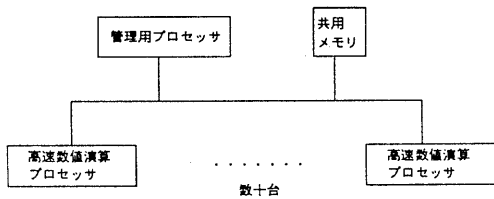


図4 システム構成

以下、本情報処理装置の詳細について述べる。

##### (1) 32ビットCISCプロセッサの採用

本情報処理装置には目標に対するフィルタリングを高速に行なうことが要求されており、更に保守及びシステムの拡張も考慮する必要がある。そこで高速で動作し、今後のシステムの拡張に対しても柔軟に対処でき、また、現在そのハードウェア及びソフトウェアの開発環境が十分整備されているプロセッサが必要である。

従来本情報処理装置の果たす役割には専用のプロセッサや、汎用のスーパーミニコンが使用されていた。しかしながら開発環境の問題やリアルタイム性の面から使用には耐えない。

そのため、本情報処理装置では現在ワークステーションなどで広く一般に使用されているモトローラ社の32ビットプロセッサMC68030を使用した。

OSは、自社で開発したものを使用している。これは、一般に販売されているリアルタイムOSではタスク切替時のオーバーヘッドに問題があり、また、本情報処理装置ではOSに対してそれほど複雑な処理を要求していないためでもある。

ソフトウェア開発には、Ada言語を使用した。これは、その保守時に読みやすく、また、プログラムの改修に対しても検証の容易なプログラミング言語が必要とされたからである。

Ada言語は米国国防総省（DOD）の主導で、組込システムの応用分野で用いるために開発された高水準プログラミング言語であり、本システムの開発に対しては最適なプログラミング言語であった。<sup>2)</sup>

##### (2) 高速数値演算プロセッサの採用

3.(1)で述べたように目標に対するフィルタリング処理を高速に行なうために浮動小数点演算を高速に行なう高速数値演算プロセッサを採用した。本プロセッサは本システム上に数十台搭載し要求されている目標数の目標を要求されている処理時間内に処理できるようにした。

高速数値演算プロセッサの性能はフィルタリング処理において、モトローラ社の32ビットプロセッサMC68030の約10倍の性能を達成することができた。

##### (3) 目標の動的フィルタ割当方式実現

3.(3)のaで述べたフィルタ割当方式アルゴリズムを図4の管理プロセッサに搭載して目標のフィルタリングを行なう高速数値演算プロセッサに対して目標の割り当てを行なっている。また、3.(3)のbで述べたように内部データを管理するために管理用プロセッサ及びフィルタリングを行なう高速数値演算プロセッサと同一バス上に共有メモリを配置した。

## 5.今後の展望

これまで述べてきたように、本情報処理装置は要求された諸機能を満足しているが、さらに改良を行っていく必要がある。以下、その改善点を示す。

### (1)S/Wのチップ化

本情報処理装置では4.(2)で述べたように高速数値演算プロセッサを採用した。しかしそれでも要求性能を満たすのがやっとであり、しかも高速数値演算プロセッサ一個につきフルサイズのボード一枚が完全に占領されている。

そこでこの高速数値演算プロセッサを含んだボード全体をチップ化し、情報処理装置本体を小型化していくと共に、チップ化することによって演算スピードの高速化が望める。

### (2)高度並列処理

アルゴリズム並列性に対応した高度並列アーキテクチャを開発する。

### (3)バスの高速化・標準化

大規模化するにつれバスネックが予想される。より高速のバスを採用した高度並列アーキテクチャを実現する必要があるが、その際CPUや言語への依存性を極力少なくした標準化バスを採用する必要がある。

## 6.おわりに

社内試作した艦載情報処理装置をもとに必要な機能、性能を得るために(1)32ビットCISCプロセッサの採用(2)高速数値演算プロセッサの採用(3)目標の動的フィルタ割当アルゴリズムの開発の検討を行った。

今後更に解決していかなければならない問題も多いが、艦載情報処理装置に関する問題点の一端が御理解いただければ幸いである。

## 参考文献

- 1) Y. ベイカー 他 : マルチマイクロプロセッサシステム、啓学出版株式会社
- 2) J. G. P. バーンズ : プログラミング言語Ada [第2版]、(株)サイエンス社