

(19) 日本国特許庁(JP)

再公表特許(A1)

(11) 国際公開番号

W02013/088494

発行日 平成27年4月27日 (2015. 4. 27)

(43) 国際公開日 平成25年6月20日 (2013. 6. 20)

(51) Int. Cl.	F I	テーマコード (参考)
<b>G06F 15/173 (2006.01)</b>	G06F 15/173	640C 3G384
<b>F02D 45/00 (2006.01)</b>	F02D 45/00	372Z 5B045
	F02D 45/00	376A

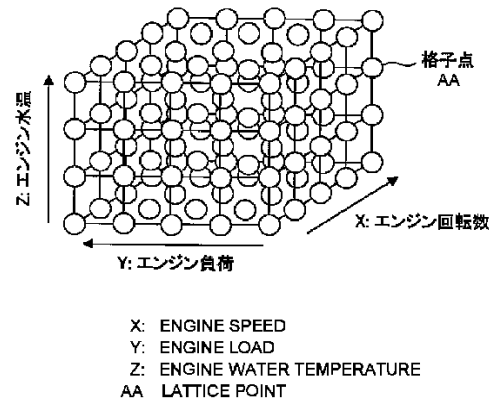
審査請求 有 予備審査請求 未請求 (全 20 頁)

出願番号 特願2013-548971 (P2013-548971)	(71) 出願人 000003207 トヨタ自動車株式会社 愛知県豊田市トヨタ町1番地
(21) 国際出願番号 PCT/JP2011/078691	
(22) 国際出願日 平成23年12月12日 (2011. 12. 12)	
(81) 指定国 AP (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), EA (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), EP (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OA (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN	(74) 代理人 100106150 弁理士 高橋 英樹 (74) 代理人 100082175 弁理士 高田 守 (74) 代理人 100113011 弁理士 大西 秀和 (72) 発明者 渡邊 智 愛知県豊田市トヨタ町1番地 トヨタ自動車株式会社内
	Fターム(参考) 3G384 AA01 AA03 BA05 BA08 BA13 BA24 BA26 BA27 DA19 EE16 EE25 EE35 EE36 5B045 GG09 GG12 KK02 KK08

(54) 【発明の名称】 エンジン制御装置

(57) 【要約】

マルチコアプロセッサに格子状に配置された複数のコアの少なくとも一部には、第1、第2及び第3の運転条件を軸とする3次元直交座標系上に配列された複数の格子点が1対1に関連付けられ、そして、関連付けられた格子点における制御値を演算するための演算プログラムが割り当てられる。格子点を関連付けられた各コアは、現在の動作点が属する3次元直交座標系上の運転空間が自身に関連付けられた格子点を含む隣接8格子点により定義される空間である場合、自身に関連付けられた格子点における制御値を前記演算プログラムにより演算して補間計算コアに送信し、そうでない場合は制御値の演算を休止する。補間計算コアは、現在の動作点を囲む隣接8格子点における制御値を用いてその動作点における制御値を補間計算する。マルチコアプロセッサは、補間計算コアで得られた現在の動作点における制御値をアクチュエータの制御目標値として出力する。



**【特許請求の範囲】****【請求項 1】**

複数のコアが格子状に配置されたマルチコアプロセッサを有し、前記マルチコアプロセッサを用いて 1 又は複数のアクチュエータの制御目標値を演算するエンジン制御装置において、

前記複数のコアの少なくとも一部には、第 1 の運転条件、第 2 の運転条件及び第 3 の運転条件を軸とする 3 次元直交座標系上に配列された複数の格子点が 1 対 1 で関連付けられ、そして、関連付けられた格子点における前記 1 又は複数のアクチュエータの制御値を演算するための演算プログラムが割り当てられ、

格子点を関連付けられた各コアは、現在の動作点が属する前記 3 次元直交座標系上の運転空間が自身に関連付けられた格子点を含む隣接 8 格子点により定義される空間である場合、自身に関連付けられた格子点における制御値を前記演算プログラムにより演算して補間計算コアに送信し、前記運転空間が自身に関連付けられた格子点を含む隣接 8 格子点により定義される空間ではない場合には、自身に関連付けられた格子点における制御値の演算を休止するようにプログラムされ、

前記補間計算コアは、前記動作点が属する運転空間を定義する隣接 8 格子点における制御値を用いて前記動作点における制御値を補間計算するようにプログラムされ、

前記マルチコアプロセッサは、前記補間計算で得られた前記動作点における制御値を前記制御目標値として出力することを特徴とするエンジン制御装置。

**【請求項 2】**

隣接する 4 つのコアからなる複数のコアグループが設定され、

前記各コアグループには、前記第 1 及び第 2 の運転条件の各座標が同一で前記第 3 の運転条件の座標が異なる 4 つの連続する格子点が関連付けられ、

前記各コアグループは、前記第 1 の運転条件と第 2 の運転条件とを軸とする 2 次元直交座標系上での格子点と同じ並びで配列されていることを特徴とする請求項 1 に記載のエンジン制御装置。

**【請求項 3】**

前記マルチコアプロセッサは、各コアが担当する格子点を前記コアグループ内で順番に入れ替えることを特徴とする請求項 2 に記載のエンジン制御装置。

**【請求項 4】**

前記マルチコアプロセッサは、所定の運転トリップ毎に前記入れ替えを実施することを特徴とする請求項 3 に記載のエンジン制御装置。

**【請求項 5】**

格子点を関連付けられた各コアは、現在の動作点が属する前記 3 次元直交座標系上の運転空間が自身に関連付けられた格子点により定義される空間であり、自身が前記運転空間を定義する複数の格子点のうちの代表格子点に関連付けられている場合、前記補間計算コアとして機能するようにプログラムされていることを特徴とする請求項 2 乃至 4 の何れか 1 項に記載のエンジン制御装置。

**【請求項 6】**

前記第 1 の運転条件と第 2 の運転条件はエンジン回転数とエンジン負荷であることを特徴とする請求項 2 乃至 5 の何れか 1 項に記載のエンジン制御装置。

**【請求項 7】**

前記第 3 の運転条件はエンジン水温であることを特徴とする請求項 6 に記載のエンジン制御装置。

**【請求項 8】**

前記複数のコアのそれぞれは、CPU と当該 CPU で実行されるプログラムを記憶したローカルメモリとを備え、

前記ローカルメモリには、前記演算プログラムと当該コアが前記補間計算コアとなった場合に実行される補間計算プログラムとが記憶されていることを特徴とする請求項 5 に記載のエンジン制御装置。

10

20

30

40

50

## 【請求項 9】

格子点を関連付けられた各コアは、関連付けられた格子点における前記 1 又は複数のアクチュエータの最適制御値を演算することを特徴とする請求項 1 乃至 8 の何れか 1 項に記載のエンジン制御装置。

## 【発明の詳細な説明】

## 【技術分野】

## 【0001】

本発明は、複数の運転条件に基づいてアクチュエータの制御目標値を決定する自動車用エンジンの制御装置に関し、詳しくは、複数のコアを有するマルチコアプロセッサを用いてアクチュエータの制御目標値を演算するエンジン制御装置に関する。

10

## 【背景技術】

## 【0002】

近年、1つの半導体チップ上に複数のCPUコアを搭載した演算装置、すなわち、マルチコアプロセッサの使用が様々な分野で提案されている。マルチコアプロセッサは、例えば特開2008-269487号公報に開示されているように、自動車用エンジンの制御の分野においてもその使用が検討されている。特開2008-269487号公報に開示されている技術は、エンジン制御装置にマルチコアプロセッサを使用する場合の消費電力の低減を目的とした技術である。この技術によれば、エンジン制御の停止中に特定の制御処理が実行される際は、マイクロコンピュータの動作モードはエンジン動作時における動作モードとは異なる低消費電力モードに設定される。低消費電力モードでは、マイクロコンピュータが使用するコアの数はエンジン動作時よりも少なくされる。エンジン制御の停止中はエンジン動作中のようにオルタネータ等からの発電電力はなく、車載バッテリーの充電も行われない。そのため、特定の制御処理の実行時に動作するコアの数が多いため、バッテリーの充電電力はどんどん消費されていくためである。

20

## 【0003】

マルチコアプロセッサを使用する利点の一つが、その処理能力の高さである。CPUコアが1つのシングルコアプロセッサと比較した場合、マルチコアプロセッサのほうがより高い処理能力を得ることができる。処理能力の高さはマルチコアプロセッサをエンジン制御装置に使用することの1つの動機となる。近年、エンジンに搭載されるアクチュエータの数と種類は益々増えているからである。エンジンの運転を適切に制御するためには、それらアクチュエータの制御目標値をエンジンの運転条件に応じた適切な値に設定する必要がある。そのために従来のエンジン制御装置では最適化演算が行われているが、その際の演算負荷はアクチュエータの数や種類が多くなるほど増大する。このため、従来のエンジン制御装置で使用されているシングルコアプロセッサの処理能力では、益々増大する演算負荷に対してやがて不足することが懸念されていた。マルチコアプロセッサを使用することは、今後益々演算負荷が増大することが予想されるエンジン制御の分野にとって大きなメリットがある。

30

## 【0004】

ただし、マルチコアプロセッサの処理能力は、単純にコアの数のみで決まるものではない。処理能力を高めるためには複数のコアを効率よく動作させる必要があり、そのためにはコアを動作させるソフトウェアに工夫が求められる。例えば、マルチコアプロセッサでは複数のコアにタスクを分配して並列に処理することが可能であるが、その並列化の手法の如何によっては全体としての処理能力に大きな違いが生じることになるし、消費電力にも大きな違いが生じることになる。

40

## 【先行技術文献】

## 【特許文献】

## 【0005】

【特許文献 1】特開2008-269487号公報

【特許文献 2】特開2011-053876号公報

【特許文献 3】特開2008-197776号公報

50

【非特許文献】

【0006】

【非特許文献1】堀江健志、林健一、“トラスネットワークにおける最適全対全通信方式”、情報処理学会論文誌、vol.34、no.4、PP628-637、1993.

【発明の概要】

【0007】

本発明は、エンジン制御にかかる1又は複数のアクチュエータの制御目標値をマルチコアプロセッサにより高速度で且つ少ない消費電力で演算できるようにすることを課題とする。そして、そのような課題を達成するために、本発明は次のようなエンジン制御装置を提供する。

10

【0008】

本発明が提供するエンジン制御装置は、複数のコアが格子状に配置されたマルチコアプロセッサを有する。ここでいう格子とは平面格子を意味するが、そのパターンには限定はない。コアの配置パターンとして一般的な正方格子や矩形格子の他、斜方格子等の他のパターンでもよい。

【0009】

マルチコアプロセッサに搭載される複数のコアには、第1の運転条件、第2の運転条件及び第3の運転条件を軸とする3次元直交座標系上に配列された複数の格子点が1対1に関連付けられている。第1の運転条件と第2の運転条件は、アクチュエータの操作量とエンジンの制御量とを関連付ける上で特に重要な運転条件であり、その1つの代表例がエンジン回転数とエンジン負荷の組である。第3の運転条件は任意であるが、現在のエンジンの運転状態のもとで優先されるべき運転条件としてもよい。例えば、エンジンの暖機中はエンジン水温を第3の運転条件とし、暖機の完了後は第3の運転条件を車速或いは吸入空気温度に変更することも可能である。

20

【0010】

3次元直交座標系上の格子点を関連付けられるコアはマルチコアプロセッサに搭載されるコアの全てでなくてもよい。例えば、コアが $N \times M$ の矩形格子状に配置されている場合、その一部である $n \times m$ の区画のコアにのみ格子点が関連付けられているのもよい。格子点を関連付けられた各コアには、関連付けられた格子点における各アクチュエータの制御値を演算するための演算プログラムが予め割り当てられている。割り当てられた制御値演算プログラムは、各コアがローカルメモリを有する場合、そのローカルメモリに記憶することができる。その場合、各コアのローカルメモリには、後述する補間計算のための補間計算プログラムも記憶することができる。以下、マルチコアプロセッサに搭載されるコアのうち、制御値演算プログラムを割り当てられたコアを制御値演算コアと呼ぶ。なお、制御値演算プログラムは、最適制御によって最適制御値を演算するプログラムであることが好ましい。

30

【0011】

前述の3次元直交座標系上には、そこに配列された複数の格子点によって複数の運転空間が定義されている。3次元直交座標系上で格子点がつる3次元の格子パターンは単純立方格子であるので、隣接する8つの格子点によって1つの運転空間が定義される。各運転条件の各現在値により定まる現在の動作点は、3次元直交座標系上の何れかの運転空間に属している。

40

【0012】

それぞれの制御値演算コアは、現在の動作点が属する運転空間が自身に関連付けられた格子点を含む隣接8格子点により定義される空間である場合、自身に関連付けられた格子点における制御値を前記演算プログラムにより演算して補間計算コアに送信するようにプログラムされている。また、それぞれの制御値演算コアは、現在の動作点が属する運転空間が自身に関連付けられた格子点を含む隣接8格子点により定義される空間ではない場合には、自身に関連付けられた格子点における制御値の演算を休止するようにプログラムされてもいる。なお、現在の動作点が属する運転空間が自身に関連付けられた格子点を含む

50

隣接 8 格子点により定義される空間であるかどうかの確認は、制御値演算コアごとに一定の時間ステップで（例えばエンジンの制御周期ごとに）行われる。

【 0 0 1 3 】

補間計算コアは、現在の動作点が属する運転空間を定義する隣接 8 格子点における制御値を用いて、現在の動作点における制御値を補間計算するようにプログラムされている。マルチコアプロセッサは、補間計算コアによる補間計算で得られた現在の動作点における制御値をアクチュエータの制御目標値として出力する。

【 0 0 1 4 】

つまり、本発明によれば、エンジン制御装置は、現在の動作点を 3 次元的に囲む隣接 8 格子点に関連付けられた 8 つのコアによって各格子点における制御値を演算し、その演算結果に基づいて現在の動作点における制御値を補間計算する。これによれば、各格子点における制御値の演算は別々のコアによって並列に行われることから、アクチュエータの制御目標値の演算に要する時間は従来のシングルコアプロセッサを用いる場合に比較して大幅に短縮される。また、エンジン制御装置は、前記隣接 8 格子点に関連付けられた 8 つのコア以外のコアについては制御値の演算を休止させるので、アクチュエータの制御目標値を高速で演算可能にしながら消費電力を抑えることができる。

10

【 0 0 1 5 】

ところで、3次元直交座標系上の格子点と平面上のコアとはランダムに関連付けることもできるし、一定の規則に従って関連付けることもできる。ただし、各コアで演算した制御値のデータを補間計算コアに渡す際には、その物理的な距離に応じた通信ディレイが生じる。この点を考慮に入れるならば、次のような規則に従って格子点とコアとの関連付けを行うことが好ましい。

20

【 0 0 1 6 】

その好ましい規則によれば、隣接する 4 つのコアからなる複数のコアグループが設定される。そして、各コアグループには、第 1 及び第 2 の運転条件の各座標が同一で第 3 の運転条件の座標が異なる 4 つの連続する格子点に関連付けられる。そして、各コアグループは、第 1 の運転条件と第 2 の運転条件とを軸とする 2 次元直交座標系上での格子点と同じ並びで配列される。このような規則で格子点とコアとの関連付けを行えば、各格子点における制御値は物理的にも互いに近接した 8 つのコアで演算されるので、コア間における通信ディレイ時間の差を小さくすることができる。通信ディレイ時間の差が小さいことでプロセッサ全体として高い効率で演算することが可能となる。

30

【 0 0 1 7 】

また、上述のような規則で格子点とコアとの関連付けが行われる場合、各コアが担当する格子点をコアグループ内で順番に入れ替えることがより好ましい。これによれば、一部のコアに演算負荷が集中することによるコアの劣化を防ぐことができる。なお、各コアが担当する格子点の入れ替えは、例えば、所定の運転トリップ毎に実施することによい。

【 0 0 1 8 】

補間計算コアとして機能するコアは何れかのコアに固定することができる。その場合、制御値演算コアのうちの何れか 1 つを補間計算コアとして定めてもよいし、制御値演算コア以外のコアの中から補間計算コアを選定してもよい。しかし、上述のような規則で格子点とコアとの関連付けが行われるのであれば、補間計算コアとして機能するコアを制御値演算コアの間で動的に変化させることがより好ましい。その場合、それぞれの制御値演算コアは、現在の動作点が属する運転空間が自身に関連付けられた格子点を含む隣接 8 格子点により定義される空間である場合、以下の処理を実行するようにプログラムされる。

40

【 0 0 1 9 】

まず、それぞれの制御値演算コアは、自身に関連付けられた格子点が現在の動作点が属する運転空間を定義する隣接 8 格子点のうちの代表格子点にあたるかどうか判定する。以下、代表格子点に対応するコアを代表コアと呼ぶ。それぞれの制御値演算コアは、自身が代表コアでない場合、自身に関連付けられた格子点における制御値を代表コアである他のコアに送信する。一方、自身が代表コアである場合には、現在の動作点が属する運転空間

50

を定義するその他の格子点を関連付けられた各コアから、それらのコアで演算された制御値を受信する。そして、現在の動作点が属する運転空間を定義する隣接 8 格子点における制御値に基づいて、現在の動作点における制御値を補間計算プログラムにより補間計算する。このような処理を行うように各コアがプログラムされることで、コア間の通信ディレイ時間の差を小さくできるだけでなく、通信ディレイ時間そのものを小さくすることができる。

【図面の簡単な説明】

【0020】

【図1】本発明の実施の形態にかかるマルチコアプロセッサのコアの配列を示す図である。

10

【図2】本発明の実施の形態にかかるマルチコアプロセッサのハードウェアアーキテクチャの概略を示す図である。

【図3】本発明の実施の形態にかかるマルチコアプロセッサのソフトウェアアーキテクチャの概略を示す図である。

【図4】エンジン回転数とエンジン負荷とエンジン水温を軸とする 3 次元直交座標系上の格子点の配列を示す図である。

【図5】3次元直交座標系上に配列された複数の格子点とマルチコアプロセッサ上の複数のコアとの関連付けについて示す図である。

【図6】3次元直交座標系上に配列された複数の格子点のマルチコアプロセッサ上の複数のコアへの関連付けの仕方について詳細に説明するための図である。

20

【図7】本発明の実施の形態によるアクチュエータの制御目標値の演算方法について説明するための図である。

【図8】本発明の実施の形態によるアクチュエータの制御目標値の演算方法について説明するための図である。

【図9】本発明の実施の形態において各コアにより実行される最適制御値算出ルーチンを示すフローチャートである。

【図10】本発明の実施の形態によるコアグループ内でのコア割付のローテーションの方法について説明するための図である。

【図11】本発明の実施の形態において各コアにより実行されるコア割付ローテーションルーチンを示すフローチャートである。

30

【図12】本発明の実施の形態の変形例 1 によるアクチュエータの制御目標値の演算方法について説明するための図である。

【図13】本発明の実施の形態の変形例 2 によるアクチュエータの制御目標値の演算方法について説明するための図である。

【発明を実施するための形態】

【0021】

本発明の実施の形態のエンジン制御装置について図を参照して説明する。

【0022】

本発明が適用されるエンジン制御装置は、マルチコアプロセッサを用いてエンジン制御にかかるアクチュエータの制御目標値を演算する装置である。本発明においては、エンジン制御装置により制御される自動車用エンジンの種類や構造に限定は無く、また、アクチュエータの種類や数にも限定は無い。例えば、本実施の形態のエンジン制御装置はガソリンエンジンの制御装置であってもよい。その場合、エンジン制御装置は、スロットル、点火装置、インジェクタ、EGR弁、可変動弁機構、及びウエストゲートバルブ等のアクチュエータの制御目標値を演算する。また、本実施の形態のエンジン制御装置はディーゼルエンジンの制御装置であってもよい。その場合、エンジン制御装置は、インジェクタ、EGR弁、ディーゼルスロットル、及び可変ノズルターボ等のアクチュエータの制御目標値を演算する。

40

【0023】

図1は、本実施の形態のエンジン制御装置が有するマルチコアプロセッサのコアの配列

50

を示す図である。本実施の形態にかかるマルチコアプロセッサ100は、多数のコア102が格子状に配置されているという構造上の特徴を有している。コア102の数は100個以上であり、マルチコアプロセッサ100はメニーコアプロセッサとも呼ばれる。図2は、マルチコアプロセッサ100のハードウェアアーキテクチャの概略を示す図である。各コア102はキャッシュ付きのCPU104とローカルメモリ106を備え、コア102同士はバス108で接続されている。ローカルメモリ106には、CPU104で実行される各種のプログラムとそのプログラムの実行時に使用される各種のデータが記憶されている。また、図示は省略するが、バス108にはコア間で共有される共有メモリも接続されている。

#### 【0024】

図3は、マルチコアプロセッサ100のソフトウェアアーキテクチャの概略を示す図である。マルチコアプロセッサ100では、それぞれのコア102上で異なるOS110が動作する。また、それぞれOS110上で異なるアプリケーション112が動作する。後述する最適制御値演算プログラムや補間計算プログラムは、そのようなアプリケーション112の1つである。各コア102に関連するOS110とアプリケーション112は、コア102ごとにローカルメモリ106にインストールされている。このようなソフトウェアアーキテクチャを有するマルチコアプロセッサ100では、各コア102が独自にアプリケーション112を実行することができる。アプリケーション112の実行のために他のコア102による演算結果や他のコア102が有するデータが必要な場合には、バス108を通じたコア間通信(OS間通信)によってデータのやり取りが行われる。

#### 【0025】

本実施の形態のエンジン制御装置は、アクチュエータの操作量とエンジンの制御量とを関連付ける上で重要な3つの運転条件について各現在値を取得し、それらに基づいて各アクチュエータの制御目標値を決定する。その3つの運転条件のうち特に重要な第1の運転条件と第2の運転条件はエンジン回転数とエンジン負荷であり、第3の運転条件はエンジン水温である。エンジン回転数、エンジン負荷及びエンジン水温が決まれば、エンジンの運転状態を最適にすることができる各アクチュエータの制御目標値を特定することができる。その特定には、エンジンを実際に試験して得られた適合データを用いることができる。ただし、エンジン回転数とエンジン負荷とエンジン水温の全ての組み合わせについて適合データを得ようとすると適合作業に膨大な時間と工数がかかってしまう。そこで提案されるのが、エンジン回転数、エンジン負荷及びエンジン水温のある特定の組み合わせについてのみ適合データを取得し、その他の組み合わせについては取得した適合データに基づいて補間計算を行うことである。本実施の形態のエンジン制御装置は、補間計算を利用したアクチュエータ制御目標値の演算をマルチコアプロセッサ100によって並列的に実施する。

#### 【0026】

マルチコアプロセッサ100で使用される適合データは、図4に格子点で示されるエンジン回転数とエンジン負荷とエンジン水温の組み合わせについて準備されている。図4は、エンジン回転数をX軸としエンジン負荷をY軸としエンジン水温をZ軸とする3次元直交座標系上の格子点の配列を示す図である。この3次元直交座標系上には100個の格子点が5行5列4段に配列されている。3次元直交座標系上で格子点がとる3次元の格子パターンは単純立方格子であり、単位格子である立方体の頂点に配置される8つの格子点によって1つの運転空間が定義される。格子点のそれぞれには、その位置におけるエンジン回転数、エンジン負荷及びエンジン水温に対して適合された適合データが関連付けられている。各格子点における適合データは最適制御値演算プログラムに組み込まれて当該格子点が関連付けられたコア102に割り当てられている。

#### 【0027】

図5は、図4に示す3次元直交座標系上に配列された格子点とマルチコアプロセッサ100上のコア102との関連付けについて示す図である。マルチコアプロセッサ100に搭載されたコア102のうち一部のコアに付された数字は、そのコアに関連付けられた格

10

20

30

40

50

子点の座標番号である。よって、(2, 1, 3)という数字が付されたコアには、X座標が2、Y座標が1、Z座標が3の格子点が関連付けられている。図5では、図4に示す100個の格子点の全てではなく、そのうちの原点に近い3行3列4段の計36個の格子点と36個のコアの関連付けのみを示している。図4において各コア102に付された格子点の座標番号から分かるように、格子点とコア102とは1対1に関連付けられている。なお、格子点の数とマルチコアプロセッサ100が有するコア102の数との関係によっては、格子点に関連付けられていないコア102が残っている場合がある。しかし、以下の説明では、コア102とは格子点に関連付けられているコアを意味するものとする。また、以下の説明では、特定のコアに言及する場合には、そのコアに関連付けられた格子点の座標番号を用いる場合がある。

10

**【0028】**

ところで、格子点は3次元直交座標系上に配置されているのに対し、コア102は2次元直交座標系上に配置されているので、格子点の配列を単純にコア102の配列に置き換えることはできない。本実施の形態では、一定の規則に従って、3次元直交座標系上の格子点の2次元直交座標系上のコア102への関連付けが行われている。図6は、その規則に関して詳細に説明するための図である。以下、本実施の形態で採っている3次元直交座標系上の格子点の平面上のコア102への関連付けの仕方について図6を用いて説明する。

**【0029】**

図6では、図5と同様に各コア102に格子点の座標番号が付されるとともに、隣接する4つのコアを囲む枠が描かれている。この枠は、それにより囲まれた4つのコア102によって1つのグループが作られていることを意味している。枠の中央に付されている符号は、そのコアグループの識別番号であり、かつ、そのコアグループに関連付けられた格子点の内容を示す番号でもある。各コアグループには、X軸及びY軸の各座標が同一でZ軸の座標が異なる4つの連続する格子点に関連付けられる。例えば識別番号Gr.11のコアグループには、X座標が1でY座標が1であるZ座標が1から4までの4つの格子点に関連付けられている。それら4つの格子点は、Z座標の数字の小さい格子点から順に、コアグループ内の左下のコアを基点にして時計回りで各コアに割り付けられている。そして、各コアグループは、エンジン回転数をX軸としエンジン負荷をY軸とする2次元直交座標系上での格子点と同じ並びで配列されている。つまり、コアグループの識別番号は、そのコアグループ自体の座標にもなっている。例えば識別番号Gr.32は、それがX軸方向に3番目でY軸方向に2番目のコアグループであることを意味している。

20

30

**【0030】**

格子点に関連付けられたコア102には、その格子点のエンジン回転数、エンジン負荷及びエンジン水温に対して適合された適合データを使用する最適制御値演算プログラムが割り当てられている。最適制御値演算プログラムは補間計算プログラムとともにローカルメモリ106にインストールされている。最適値演算プログラムは、エンジン特性モデルを用いてアクチュエータの最適制御値を演算するプログラムである。前述の適合データはエンジン特性モデルのパラメータの値に用いられる。本実施の形態のマルチコアプロセッサ100では、格子点に関連付けられたコア102のみがアクチュエータ制御目標値の演算のための最適値演算コアとして機能し、それ以外のコアはエンジン制御に必要なその他の処理を担う。

40

**【0031】**

次に、マルチコアプロセッサ100を用いたアクチュエータの制御目標値の演算方法について図7及び図8を用いて説明する。図7に示す3次元直交座標系において、点Aは、エンジン回転数、エンジン負荷及びエンジン水温の各現在値により定まる動作点、つまり、現在の動作点を示している。ここでいう現在とは、今回の時間ステップの開始時点現在を意味する。エンジンを適切に制御するためには、現在の動作点におけるアクチュエータの最適制御値を演算し、それをアクチュエータの制御目標値として設定する必要がある。ただし、3次元直交座標系において最適制御値を演算するための適合データが用意されて

50



いるのは5行5列4段に配列された100個の格子点のみであり、それ以外の領域では適合データは用意されていない。よって、現在の動作点における最適制御値を最適制御値演算プログラムによって直接に算出することはできない。

#### 【0032】

そこで、本実施の形態では、現在の動作点を囲む8つの格子点における最適制御値を演算し、それらを用いて現在の動作点における最適制御値を補間計算することが行われる。図7に示す例では、現在の動作点Aは、座標(2, 2, 2)、座標(2, 3, 2)、座標(3, 2, 2)、座標(2, 2, 3)、座標(2, 3, 3)、座標(3, 2, 3)、座標(3, 3, 2)及び座標(3, 3, 3)の格子点によって定義される空間に属している。よって、これら隣接8格子点における最適制御値が分かれば、それらを用いた補間計算によって現在の動作点Aにおける最適制御値を得ることができる。

10

#### 【0033】

現在の動作点Aにおける最適制御値は、具体的には次の手順で演算される。まず、図8に示すコアの配置において、マルチコアプロセッサ100は、格子点を関連付けられたコア102のうち(2, 2, 2)、(2, 3, 2)、(3, 2, 2)、(2, 2, 3)、(2, 3, 3)、(3, 2, 3)、(3, 3, 2)及び(3, 3, 3)のコアを演算に使用する。これら8つのコアのそれぞれは、自身のローカルメモリ106にインストールされている最適制御値演算プログラムによって、自身に関連付けられた格子点における最適制御値を演算する。このとき、これら8つのコアによる最適制御値の演算は並列に行われる。これら8つのコアにて最適制御値の演算が完了したら、その演算結果のデータは8つのコアのうちの代表コアに集められる。各コア102は、現在の動作点を囲む隣接8格子点のうち3次元直交座標系において原点に最も近い格子点を代表格子点として認識し、その代表格子点を関連付けられたコアを代表コアとして認識するようにプログラムされている。図8に示す例では、(2, 2, 2)のコアが代表コアとなる。代表コアである(2, 2, 2)のコアは、自身が演算した最適制御値のデータと、他の7つのコアから受信した最適制御値のデータとを用いて、自身のローカルメモリ106にインストールされている補間計算プログラムによって現在の動作点Aにおける最適制御値を補間計算する。

20

#### 【0034】

以上述べた方法によれば、補間計算のために必要な隣接8格子点における最適制御値はそれぞれ別々のコアにより並列に演算される。これにより、アクチュエータの制御目標値の演算に要する時間を従来のシングルコアプロセッサを用いる場合に比較して大幅に短縮することができる。

30

#### 【0035】

また、上述の規則で格子点とコアとの関連付けを行ったことにより、隣接8格子点における最適制御値の演算を物理的にも互いに近接した8つのコアにて行うことができる。さらに、その8つのコアのうちの1つが補間計算コアとして機能する。これらによれば、各コアで演算した最適制御値のデータを補間計算コアに渡す際に生じる通信ディレイ時間の差を小さくできるだけでなく、通信ディレイ時間そのものを小さくすることができる。

#### 【0036】

なお、以上述べた方法は、各コア102において図9のフローチャートに示す最適制御値算出ルーチンが実行されることにより実施される。このルーチンは、各コア102のローカルメモリ106にインストールされたアプリケーションによって、コア102ごとに一定の時間ステップ(例えばエンジンの制御周期)で実行される。

40

#### 【0037】

図9のフローチャートの最初のステップS102では、コア102はエンジン回転数、エンジン負荷及びエンジン水温の各現在値を取得する。それらは、最適制御値演算コア以外のコアによってセンサの出力値に基づいて計算されている。次に、コア102は、エンジン回転数、エンジン負荷及びエンジン水温の各現在値から現在の動作点を特定し、現在の動作点が自身に関連付けられた格子点を含む隣接8格子点で定義される空間内にあるかどうか判定する。

50

## 【 0 0 3 8 】

ステップ S 1 0 2 の判定の結果が否定である場合、コア 1 0 2 は最適制御値の演算を休止して今回時間ステップでの演算を終了する。無駄な電力消費を抑えるためである。

## 【 0 0 3 9 】

ステップ S 1 0 2 の判定の結果が肯定である場合、コア 1 0 2 はステップ S 1 0 4 の処理を実行する。ステップ S 1 0 4 では、コア 1 0 2 は最適制御値演算プログラムを実行し、自身に関連付けられた格子点における最適制御値を演算する。

## 【 0 0 4 0 】

次のステップ S 1 0 6 では、コア 1 0 2 は自身が代表コアに該当するかどうか判定する。代表コアであるかどうかは、自身に関連付けられた格子点と現在の動作点との位置関係から判断することができる。

10

## 【 0 0 4 1 】

ステップ S 1 0 6 の判定の結果が否定である場合、コア 1 0 2 はステップ S 1 1 2 の処理を実行する。ステップ S 1 1 2 では、コア 1 0 2 は自身が演算した最適制御値のデータを代表コアに送信する。どのコアが代表コアに該当するかは、3次元直交座標系上での現在の動作点の座標から判断することができる。最適制御値のデータの送信により、当該コア 1 0 2 による今回時間ステップでの演算は終了する。

## 【 0 0 4 2 】

ステップ S 1 0 6 の判定の結果が肯定である場合、コア 1 0 2 はステップ S 1 0 8 の処理を実行する。ステップ S 1 0 8 では、コア 1 0 2 は、現在の動作点が属する運転空間を定義するその他の格子点に関連付けられた他の7つのコアから、それらのコアで演算された最適制御値のデータを受信する。

20

## 【 0 0 4 3 】

次のステップ S 1 1 0 では、代表コアであるコア 1 0 2 は、自身が演算した最適制御値を含む隣接8格子点における最適制御値に基づいて、現在の動作点における最適制御値を補間計算プログラムにより補間計算する。補間計算で得られた現在の動作点における最適制御値は、アクチュエータの制御目標値として出力される。これにより、当該コア 1 0 2 による今回時間ステップでの演算は終了する。

## 【 0 0 4 4 】

以上のルーチンがそれぞれのコア 1 0 2 にて実行されることで、アクチュエータの制御目標値を高速で演算可能にしながら消費電力を抑えることができる。

30

## 【 0 0 4 5 】

ところで、本発明の制御装置が適用される自動車用のエンジンの特徴として、運転条件ごとに運転頻度が異なるということが挙げられる。例えば、一般に暖機前の運転時間は暖機後の運転時間より短く、アイドルの運転時間は全負荷の運転時間より短い。このことは、エンジン回転数、エンジン負荷及びエンジン水温を軸とする3次元直交座標系においても運転頻度が高い空間と運転頻度が低い空間とが存在することを意味する。この場合、運転頻度が高い空間に属する格子点に関連付けられたコアは、他のコアに比べて相対的に演算負荷が高くなる。特定のコアに演算負荷が集中するとそのコアの劣化を招くことになってマルチコアプロセッサ 1 0 0 全体としての寿命を縮めてしまう。

40

## 【 0 0 4 6 】

そこで、本実施の形態のエンジン制御装置では、図 1 0 に示すように、各コアが担当する格子点をコアグループ内で順番に入れ替えることが行われる。つまり、コアの割付のローテーションが実行される。図 1 0 に示す例では、ローテーションは矢印で示すように時計回りに行っているが、反時計回りに行うことも可能である。

## 【 0 0 4 7 】

コアの割付のローテーションは、各コア 1 0 2 において図 1 1 のフローチャートに示すコア割付ローテーションルーチンが実行されることにより実施される。このルーチンの最初のステップ S 2 0 2 では、前回のコアの割付ローテーションの実施からの累積のエンジンの運転時間が所定時間異常経過したかどうか判定される。その判定の結果が肯定の場合

50

、ステップS 2 0 4の処理が実行される。ステップS 2 0 4では、今回運転開始時のイグニッションのオンのタイミングにてコアの割付ローテーションが実施される。このように所定の運転トリップ毎にコアの割付ローテーションを実施することで、一部のコアに演算負荷が集中することによるコアの劣化を防ぐことができる。なお、このルーチンは、各コア1 0 2のローカルメモリ1 0 6にインストールされたアプリケーションによって実行される。

【0 0 4 8】

その他、

本発明は上述の実施の形態に限定されるものではなく、本発明の趣旨を逸脱しない範囲で種々変形して実施することができる。例えば次のように変形して実施してもよい。

10

【0 0 4 9】

上記実施の形態では、最適制御値を演算したコアの中から選定される代表コアにて補間計算を行っている。しかし、その他のコアにおいて補間計算を行うことも可能である。例えば、図1 2及び図1 3はそれぞれ上記実施の形態の変形例を示している。図1 2の変形例に示すように、格子点を関連付けられたコア1 0 2以外の何れか1つのコアを補間計算コアとすることができる。或いは、図1 3の変形例に示すように、格子点を関連付けられたコア1 0 2のうちの何れか1つを補間計算コアとして固定することもできる。これらの変形例では、各コアで演算した最適制御値のデータを補間計算コアに送信する際、その物理的な距離に応じて通信ディレイ時間は大きくなる。しかし、補間計算で用いる最適制御値の演算は、物理的に近接したコアで行われるので、コア間における通信ディレイ時間の差は小さい。通信ディレイ時間の差が小さいことでマルチコアプロセッサ全体としては高い効率で演算することができる。

20

【0 0 5 0】

格子点を関連付けるコアは、マルチコアプロセッサに搭載されるコアの一部でもよく全部でもよい。マルチコアプロセッサに搭載される全てのコアに格子点を関連付けるのであれば、エンジン制御に必要なその他の処理は、格子点を関連付けられているコアの中で平均的な負荷が比較的小さいコアに担わせればよい。なお、マルチコアプロセッサに搭載されるコアの数は好ましくは4個以上、より好ましくは9個以上あればよく、その数に具体的な限定は無い。また、3次元直交座標系上に配列する格子点の数は、マルチコアプロセッサに搭載されるコアの数以下であるならば、その数に具体的な限定は無い。

30

【0 0 5 1】

アクチュエータの制御目標値を決定する際に基づく3つの運転条件のうち、第3の運転条件はエンジンの状態によって変更してもよい。例えば、エンジンの暖機が完了するまでの間は、上記実施の形態と同じくエンジン水温を第3の運転条件として使用し、エンジンの暖機の完了後は、吸入空気温度を第3の運転条件として使用するようにしてもよい。エンジンの暖機が完了している場合でも吸入空気温度には変化がある。このため、暖機が完了している状況でエンジンの状態に与える影響は、エンジン水温よりも吸入空気温度のほうが大きい。よって、エンジンの暖機が完了したらエンジン水温から吸入空気温度へと第3の運転条件を変更することにより、格子点の3次元配列をより有効に活用してエンジンの状態をよりの確に制御することが可能となる。エンジンの暖機完了後の第3の運転条件は吸入空気温度に替えて車速としてもよい。車速により走行風の強さが異なり、排気系の温度に違いが生じるからである。その他、連続的に変化する運転条件であれば第3の運転条件として好ましい。例えば気圧やE G R率は第3の運転条件として好適である。

40

【符号の説明】

【0 0 5 2】

1 0 0 マルチコアプロセッサ

1 0 2 コア

1 0 4 C P U

1 0 6 ローカルメモリ

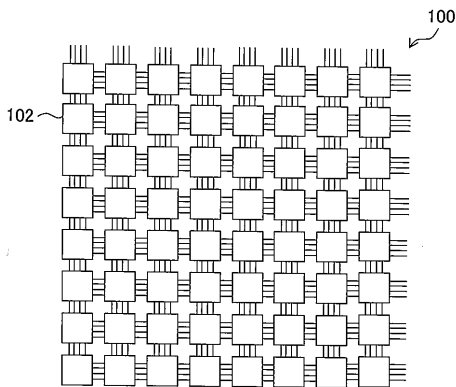
1 0 8 バス

50

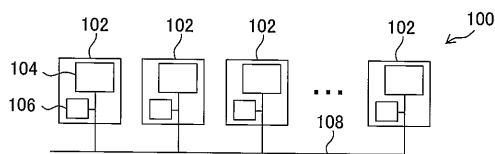
1 1 0 OS

1 1 2 アプリケーション

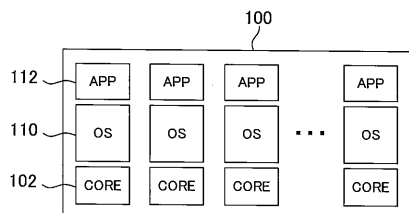
【 図 1 】



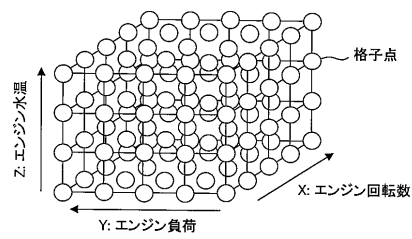
【 図 2 】



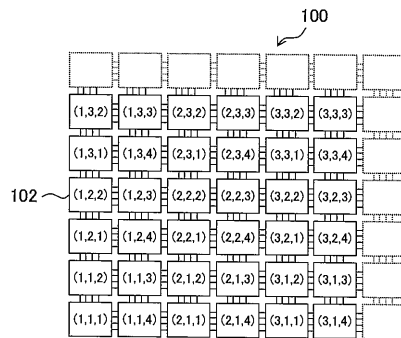
【 図 3 】



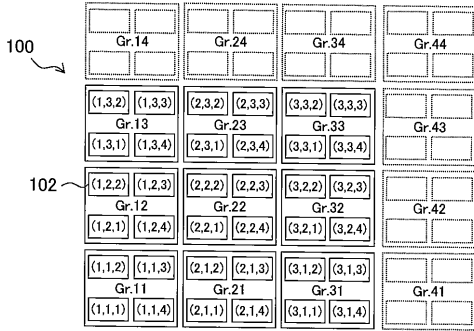
【 図 4 】



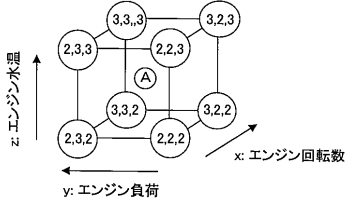
【 図 5 】



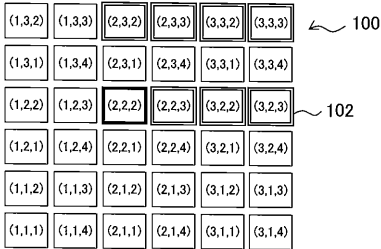
【 図 6 】



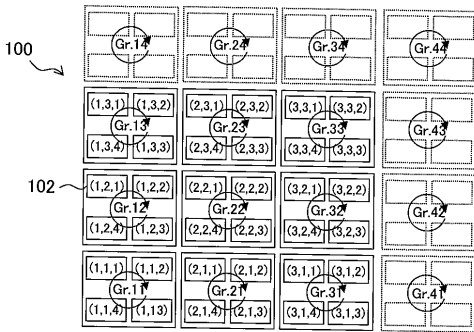
【 図 7 】



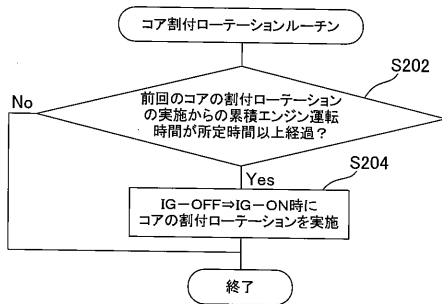
【 図 8 】



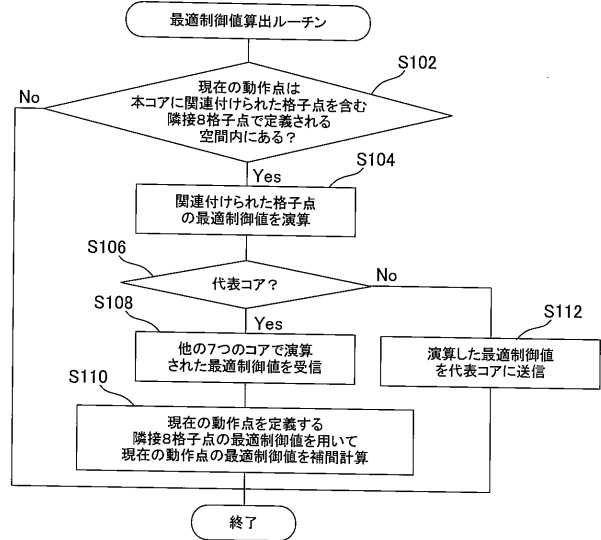
【 図 10 】



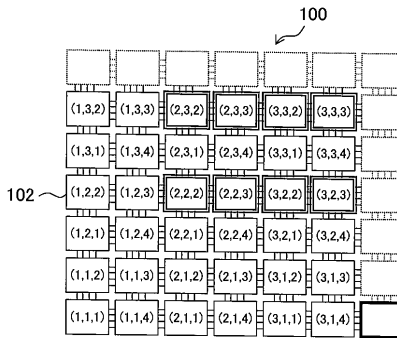
【 図 11 】



【 図 9 】



【 図 12 】



【 図 13 】

