

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3679451号

(P3679451)

(45) 発行日 平成17年8月3日(2005.8.3)

(24) 登録日 平成17年5月20日(2005.5.20)

(51) Int.Cl.⁷

F I

G O 1 C 21/00

G O 1 C 21/00

D

G O 8 G 1/0969

G O 8 G 1/0969

G O 9 B 29/10

G O 9 B 29/10

A

請求項の数 1 (全 18 頁)

(21) 出願番号 特願平7-97215
 (22) 出願日 平成7年4月21日(1995.4.21)
 (65) 公開番号 特開平8-292039
 (43) 公開日 平成8年11月5日(1996.11.5)
 審査請求日 平成14年4月5日(2002.4.5)

(73) 特許権者 591132335
 株式会社ザナヴィ・インフォマティクス
 神奈川県座間市広野台二丁目6番35号
 (74) 代理人 100084032
 弁理士 三品 岩男
 (72) 発明者 佐藤 裕幸
 神奈川県座間市広野台2丁目4991番地
 株式会社ザナヴィ・インフォマティクス
 内

審査官 片岡 弘之

(56) 参考文献 特開平03-027083(JP, A)

最終頁に続く

(54) 【発明の名称】 現在位置算出装置

(57) 【特許請求の範囲】

【請求項1】

車輪の回転に伴い移動する車両に搭載され、当該車両の現在位置を算出する現在位置算出装置であって、

道路地図を表す地図データを記憶する手段と、

車両の進行方位を検出する手段と、

車輪の回転速度を検出する手段と、

検出された車輪の回転速度と、設定された距離係数に応じて車両の走行距離を算出する走行距離算出手段と、

検出された車両の走行距離と検出された車両の進行方位に応じて、車両が存在する前記地図データの表す道路地図上の位置を算出する手段と、

補正量を求めて、当該補正量を用いて前記走行距離算出手段に設定された距離係数を補正する距離係数補正手段とを有し、

前記距離係数補正手段は、

前記距離係数に施された前記補正量に応じて定まる値の大きさの合計である累積補正量を求め、求めた当該累積補正量が小さいほど、前記距離係数に施す補正の補正量を小さく修正する

ことを特徴とする現在位置算出装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】

本発明は、車両等の移動体に搭載され、走行距離や進行方位等を測定して、これらより当該車両の現在位置を算出する現在位置算出装置に関するものであり、特に、走行距離の測定誤差を補正する技術に関するものである。

【0002】**【従来の技術】**

従来、車両の現在位置は、ジャイロ等の方位センサにより測定した車両の進行方向と、車速センサまたは距離センサにより測定した車両の走行距離に基づいて算出することが行われている。

【0003】

また、車両の走行距離は、一般的には、トランスミッションの出力軸、または、タイヤの回転数を計測して、その回転数に、タイヤ1回転あたりに車両が進む距離である距離係数を乗ずることにより求められている。

【0004】

また、このように車両の進行方向と走行距離から求めた現在位置の誤差を補正するために、特公平6-13972号記載の技術のように、道路データに整合するように、求められた車両の現在位置を修正する、いわゆる、マップマッチングの技術が知られている。このマップマッチングの技術によれば、位置算出の精度を高めることができる。

【0005】

ところで、走行時には、タイヤの磨耗や、温度変化による膨張等により、タイヤの直径、すなわち、距離係数が時事刻々と変化する。このため、走行距離の算出において誤差が発生し、現在位置の算出が高精度に行えなくなる。たとえば、タイヤ1回転あたりの走行距離係数に、1%の誤差が存在すると、100km走行した場合、1kmの誤差が発生してしまう。

【0006】

このような走行距離の測定誤差は、通常の道路を走行している場合は、前述したマップマッチングの技術により、ある程度修正できる。しかし、高速道路等の道路を走行する際には、マップマッチングで利用できるカーブや交差点等の特徴が道路にないため、十分に誤差を修正することができなくなる。

【0007】

さらに、一旦、測定した現在位置と真の現在位置との間に1km程度の誤差が発生してしまうと、マップマッチングの技術によっては、正しい位置に補正することが困難になる。

【0008】

そこで、走行距離の測定の誤差を無くすために、従来は、(1)交差点を曲がった時(始点)から、次の交差点を曲がる(終点)までの道路データと、計測した回転数より得られる走行距離とを比較することにより、タイヤ1回転あたりの距離係数を補正することが行われていた。また、(2)特公平6-27652号公報に記載のように、2つのビーコンの間の地図上の距離と、走行して計測した距離を比較することにより、前述した距離係数を補正する技術も知られている。

【0009】**【発明が解決しようとする課題】**

しかし、前述した各技術では、距離係数が真の値に近い値を有している場合でも、道路地図の一部に誤りがあった場合や、一時的に各センサが誤った値を出力してしまった場合等には、距離係数が真の値から、はずれる方向に大きく補正されてしまうことがある。

【0010】

そこで、本発明は、距離係数が真の値に近い値を有している場合に、距離係数が真の値から大きくはずれる方向に補正されてしまうことを防ぐことができる現在位置算出装置を提供することを目的とする。

【0011】**【課題を解決するための手段】**

10

20

30

40

50

前記目的達成のために、本発明は、車輪の回転に伴い移動する車両に搭載され、当該車両の現在位置を算出する現在位置算出装置であって、道路地図を表す地図データを記憶する手段と、車両の進行方位を検出する手段と、車輪の回転速度を検出する手段と、検出された車輪の回転速度と、設定された距離係数に応じて車両の走行距離を算出する走行距離算出手段と、検出された車両の走行距離と検出された車両の進行方位に応じて、車両が存在する前記地図データの表す道路地図上の位置を算出する手段と、補正量を求めて、当該補正量を用いて前記走行距離算出手段に設定された距離係数を補正する距離係数補正手段とを有し、前記距離係数補正手段は、現在地点までの一定距離を車両が走行した間、若しくは現在までの一定期間における、前記距離係数に施された前記補正量に応じて定まる値の大きさの累積である累積補正量を求め、求めた当該累積補正量の大きさに応じて、前記距離係数に施す補正の補正量を修正することを特徴とする現在位置算出装置を提供する。

10

【0012】

【作用】

本発明の現在位置算出装置によれば、現在地点までの一定距離を車両が走行した間、もしくは、現在までの一定期間の、前記距離係数に施した補正の補正量の推移状況、または、前記ずれ量に応じて求めた補正量の推移状況に応じて、前記距離係数に施す補正の補正量を修正する。

【0013】

したが、たとえば、前記補正量の推移状況が、補正量が値零周辺で安定して推移している推移状況である場合に、より小さい値になるように前記補正量を修正すれば、距離係数が真の値に近い値を有している場合に、ノイズ等によって距離係数が真の値から大きくはずれ方向に補正されてしまうことを防ぐことができる。なぜならば、補正量が値零周辺で安定して推移している場合には、距離係数が真の値に近い値を有していると推定することができるからである。

20

【0014】

【実施例】

以下、本発明に係る現在位置算出装置の一実施例について説明する。

【0015】

図1に本実施例に係る現在位置算出装置の構成について説明する。

【0016】

30

すなわち、本実施例に係る現在位置算出装置は、車両のヨーレイトを検出することで進行方位変化を検出する角速度センサ201と、地磁気を検出することで車両の進行方位を検出する地磁気センサ等の方位センサ202と、車両のトランスミッションの出力軸の回転速度に比例した時間間隔でパルスを出力する車速センサ203を備えている。

【0017】

また、現在位置周辺の地図や現在位置を示すマーク等を表示するディスプレイ207と、ディスプレイ207に表示する地図の縮尺切り替えの指令をユーザに（運転者）から受け付けるスイッチ204と、デジタル地図データを記憶しておくCD-ROM205と、そのCD-ROM205から地図データを読みだすためのドライバ206とを備えている。また、以上に示した各周辺装置の動作の制御を行うコントローラ208を備えている。

40

【0018】

次に、コントローラ208は、角速度センサ201の信号（アナログ）をデジタル信号に変換するA/D変換器209と、方位センサ202の信号（アナログ）をデジタル信号に変換するA/D変換器210と、車速センサ203から出力されるパルス数を0.1秒毎にカウントするカウンタ216と、スイッチ204の押圧の有無を入力するパラレルI/O211と、CD-ROM205から読みだされた地図データを転送するDMA（Direct Memory Access）コントローラ212と、ディスプレイ207に地図画像を表示する表示プロセッサ213とを有する。

【0019】

また、コントローラ208は、さらに、マイクロプロセッサ214と、メモリ215とを

50

有する。マイクロプロセッサ 214 は、A/D 変換器 209 を介して得た角速度センサ 201 の信号、A/D 変換器 210 を介して得た方位センサ 202 の信号、カウンタ 216 がカウントした車速センサ 203 の出力パルス数、パラレル I/O 211 を介して入力するスイッチ 204 の押圧の有無、DMA コントローラ 212 を介して得た CD-ROM 205 からの地図データを受け入れて、それら信号に基づいて処理を行い、車両の現在位置を算出して、それを表示プロセッサ 213 を介してディスプレイ 207 に表示させる。この車両位置の表示は、図 2 に示すように、すでにディスプレイ 207 に表示している地図上に矢印マーク等を重畳して表示することにより行う。これにより、ユーザは、地図上で車両の現在位置を知ることができる。メモリ 215 は、このような動作を実現するための処理（後述）の内容を規定するプログラムや、後述する各種テーブル等を格納した ROM と、マイクロプロセッサ 214 が処理を行う場合にワークエリアとして使用する RAM とを含んでいる。

10

【0020】

以下、本実施例に係る現在位置算出装置の動作について説明する。

【0021】

まず、車両の進行方位及び走行距離を算出する処理、算出された進行方位及び距離から車両の現在位置を決定する処理、得られた車両位置および方位を表示する処理の三つの処理について説明する。

【0022】

図 3 に、車両の進行方位及び走行距離を算出する処理の流れを説明する。

20

【0023】

この処理は、一定周期、たとえば 100 m S 毎に起動され実行されるマイクロプロセッサ 214 のルーチンである。

【0024】

このルーチンでは、最初、A/D 変換器 209 から角速度センサ 201 の出力値を読み込む（ステップ 401）。この角速度センサ 201 の出力値には、方位変化が出力されるので、車両の進行方向の相対的な値しか検出できない。このため、次に、A/D 変換器 210 から方位センサ 202 の出力値を読み込み（ステップ 402）、この方位センサ 202 の出力値により算出された絶対方位と角速度センサ 201 から出力される方位変化（角速度出力）とを用いて、車両の推定方位を決定する（ステップ 403）。

30

【0025】

この方位の決定は、たとえば、長い時間、車速が低い時には、角速度センサ 201 の誤差が大きいので、一定時間以上車速が低い場合には、方位センサ 202 の出力する方位のみを利用するという方法により行う。

【0026】

次に、車速センサ 203 の出力するパルス数を、0.1 秒毎に、カウンタ 216 で計数して、その計数値を読み込む（ステップ 404）。この読み込んだ値に、距離係数 R を乗算することで、0.1 秒間に進んだ距離を求める（ステップ 405）。この距離係数 R の求め方については後述する。

【0027】

次に、このようにして求められた 0.1 秒間あたりの走行距離値を、前回得られた値に積算して、車両の走行距離が 20 m となったかどうかを調べ（ステップ 406）、20 m に満たない場合（ステップ 406 で No）、今回の処理を終了して、新たな処理を開始する。

40

【0028】

走行距離算出処理の結果、積算された走行距離が一定距離、例えば 20 m となった場合（ステップ 406 で Yes）、その時点での進行方向と走行距離（20 m）とを出力する（ステップ 407）。ステップ 407 では、さらに、積算距離を初期化して、新たに走行距離の積算を開始する。

【0029】

50

次に、算出された進行方位および走行距離に基づいて、車両の現在位置を求める処理について説明する。

【0030】

図4に、この処理の流れを示す。

【0031】

本処理は、図3からの進行方位および走行距離が出力されるのを受けて起動され、実行されるマイクロプロセッサ214のルーチンである。すなわち、本処理は、車両が20m進む毎に起動される。

【0032】

さて、この処理では、まず、ステップ407で出力された進行方位と走行距離とを読み込む（ステップ501）。次に、それらの値に基づいて、車両の移動量を緯度経度方向、別々に、それぞれ求める。さらに、これらの各方向における移動量を、前回の処理で求められた車両の現在位置に加算して、現在位置（A）を求める（ステップ502）。 10

【0033】

もし、装置の始動直後など、前回求められた位置がない場合には、別途設定された位置を前回求められた位置として用いて現在位置（A）を求める。

【0034】

次に、求めた現在位置（A）の周辺の地図を、CD-ROM205から、ドライバ206およびDMAコントローラ213を介して、読み出し、現在位置（A）を中心とする予め設定された距離D内にある道路データ（線分）を抜き出す（ステップ503）。 20

【0035】

なお、道路データとしては、たとえば、図6に示すように、2点間を結ぶ複数の線分81～85で近似し、それら線分を、その始点と終点の座標によって表したものを等を用いることができる。たとえば、線分83は、その始点（x3、y3）と終点（x4、y4）によって表現するようにする。

【0036】

次に、ステップ503で抜き出された線分の中から、その線分の方位が、求められている進行方向と、所定値以内にある線分だけを抜き出す（ステップ504）。さらに、抜き出されたすべての線分に対し、現在位置（A）から垂線をおろし、その垂線の長さを求める（ステップ505）。 30

【0037】

次に、それら垂線の長さをを用いて、ステップ504で抜き出されたすべての線分に対して、以下に定義されるエラーコスト値を算出する。

【0038】

エラーコスト = $\alpha \times | \text{進行方向} - \text{線分方位} | + \beta \times | \text{垂線の長さ} |$

ここで、 α 、 β は、重み係数である。これら係数の値は、進行方向と道路の方位のずれと現在位置と道路のずれのどちらを、現在位置が、その上にある道路を選択する上で重視するかによって変化させてよい。たとえば、進行方向と方位が近い道路を重視する場合は、 α を大きくするようにする。

【0039】

そして、各線分のエラーコストが算出されたならば、エラーコストが算出された線分のうち、エラーコスト値が最も小さい線分を選び（ステップ506）、その選択された線分と垂線との交差する点（線分の垂線のあし）を、修正された現在位置（B）とする（ステップ507）。 40

【0040】

ところで、前述したステップ503では、現在位置（A）を中心とする予め設定された距離D内にある道路データ（線分）を抜き出したが、この距離Dは、前回行ったステップ506で選択した道路のエラーコストの値に基づいて決定する値でもよい。

【0041】

ここで、エラーコストに基づいて、検索範囲を求める理由は、エラーコストの大きい場 50

合は、前回求めた現在位置（B）の精度に対する信憑性が低いと考えられるので、より広い範囲を検索して道路を探す方が、正しい現在位置を求める上で適当であるからである。

【0042】

次に、得られた車両位置および方位を表示する処理について説明する。

【0043】

図5に、この処理の流れを示す。

【0044】

本処理は、1秒毎に起動され実行されるマイクロプロセッサ214のルーチンである。

【0045】

最初、スイッチ204が押圧により地図の縮尺の変更を指示されているかを、パラレルI/O211の内容を見て判断する（ステップ601）。もし、押されていれば（ステップ601でYes）、それに対応して、所定の縮尺フラグを設定する（ステップ602）。 10

【0046】

次に、図4の処理で求められた現在位置（B）を読み出し（ステップ603）、ステップ602で切り替えられた縮尺フラグの内容に応じた縮尺の地図をディスプレイ207に、例えば、図2に示すような状態で表示する（ステップ604）。

【0047】

そして、地図に重畳して、車両の現在位置（B）と車両の進行方位を、たとえば、先に示した図2のように、矢印記号“↑”を用いて表示する（ステップ605）。そして最後に、これらに重畳して、北を示す北マークと、縮尺に対応した距離マークとを、図2のように表示する（ステップ606）。 20

【0048】

なお、本実施例においては、上記のように矢印記号を用いて車両位置および方向を示したが、車両位置および方向の表示形態は、位置および進行方向が、表示状態が明確に示されるものであれば、その形態は任意でよい。また、北マーク等も同様である。

【0049】

次に、図3の処理ステップ405において距離の算出に用いた距離係数Rの求め方について説明する。

【0050】

さて、前述したように車両の走行距離は、車速センサ203の出力するパルス数に距離係数Rを乗じて求める。しかし、タイヤの摩耗等によりタイヤ1回転あたりの車両の進行距離は変化するので、距離係数Rを固定値とすると、走行に伴い距離が正確に求まらなくなってくる。そこで、本実施例では、逐次、図4の処理で得られた現在位置（B）（ステップ508）と、CD-ROM205からドライバ206を介して読みだした地図データから求まる道路方位と、図3の処理で得られた車両方位（ステップ403）とを比較することにより、現在位置（B）が、実際の位置に対して進んでいるか遅れているかを判断し、距離係数Rを動的に修正する。 30

【0051】

このような距離係数Rの修正は、たとえば、次のようにして行うことができる。

【0052】

すなわち、補正変数Rshを導入する。そして、距離係数 $R = R_0 \times (1 + Rsh)$ に従い、動的に距離係数Rを修正する。なお、ここでR0は、予め定めた距離係数Rの初期値を示している。 40

【0053】

さて、補正変数Rshは、マイクロプロセッサ214が、次に示す処理を実行することにより逐次変更する。

【0054】

まず、補正変数Rshの求め方の概要を説明する。

【0055】

補正変数Rshは、道路のカーブの個所の車両の実際の曲がり方に応じて決定する。 50

【0056】

たとえば、図4のステップ508で出力された現在位置（B）が、図7の道路Aを進行するものであったとする。一方、たとえば、逐次出力される現在位置（B）のうちの、走行上カーブ開始点oより所定距離だけ前のa点を基準に、車両の進行方位の走行距離から求めた車両の現在位置の軌跡が、図中に破線で示したようにo点をオーバした地点から曲がり始めた軌跡であったとする。

【0057】

この場合には、距離係数Rが真の値より大きかったものと推定することができる。なぜならば、車両がカーブに達したと距離係数Rを用いて推定された時点では、車両は曲がり始めていないので実際には車両はカーブに達していないと考えられるからである。また、距離係数Rが真の値より大きい度合いは、車両が実際に曲がり始めた地点とカーブ開始点oとの距離を表す図中の距離dより推定することができる。

10

【0058】

また、逆に、車両の進行方位の走行距離から求めた車両の現在位置の軌跡が、交叉点oに到る前に曲がり始めた軌跡である場合には、距離係数Rが真の値より小さかったものと推定することができる。なぜならば、車両がカーブoに達したと距離係数Rを用いて推定される前に、車両は曲がり始めているので実際には車両は既にカーブoに達していると考えられるからである。また、この場合も、距離係数Rが真の値より小さい度合いは、車両が実際に曲がり始めた地点とカーブoとの距離を表す図中の距離誤差dより推定することができる。

20

【0059】

したがい、図中のdより距離係数が真値よりのずれの方向と、その度合いを判定し、これを打ち消すために、必要な補正変数Rshの変化量を定めることができる。

【0060】

または、車両方位と道路の方位差から、車両がカーブoに達したと距離係数Rを用いて推定されるより前に、車両が曲がり始めたか、車両がカーブoに達したと距離係数Rを用いて推定された時点より後に車両が曲がり始めたかを判定するようにしてもよい。この場合には、車両方位と道路の方位差の積分値より、距離係数Rが真の値よりずれている度合いを求めることができる。

【0061】

本実施例では、このようにして、距離誤差dが検出された場合に距離誤差dの値に応じて、補正変数Rshの変化量を求め、これによって補正変数Rshを補正することにより距離係数Rを補正する。

30

【0062】

この、補正変数Rshの補正は、具体的には、次のように行う。

【0063】

すなわち、まず、300m走行する間は、Rshの絶対値を正しいRを得るための値より大きくすることにより、この300m走行する間に、現在位置（B）として漸次正しい値が求まるようにする。これは、距離誤差dが求まった時点では、それまでの距離係数が誤差によって、誤った現在位置（B）が求まっていると推定される。しかし、ここで、現在位置（B）を一時に正しい位置に修正すると、前述した現在位置の表示マークは、走行方向と逆方向に移動したり、一気に離れた位置に移動してしまい、運転者に混乱を与えることになる。そこで、300m走行する間に、Rshの変化量の絶対値を正しいRを得るための値より大きくし、この300m走行する間に、現在位置（B）のマークの表示位置が少しずつ正しい位置に修正されていくようにするのである。具体的には、距離誤差dが求まったならば、それまでのRshを変数Rsh0として記憶し、このRsh0に、距離誤差dに正の係数を乗じて定めたRshの変化量Rsh300を加算することにより、Rshを、その絶対値が正しいRを得るための値より大きくなるように補正する。

40

【0064】

そして、この300m走行した後は、再度Rshの絶対値を正しいRを得るための値に修

50

正し、その後は正しい距離係数 R で、現在位置 (B) が求まるようにするのである。この処理は、具体的には、変化量 $Rsh300$ を加算する前の Rsh である $Rsh0$ に、変化量 $Rsh300$ に 1 より小さな正の係数を乗じた値を加算することにより、正しい R を得るための値に修正する。

【0065】

結果、距離誤差 d が検出されてから $300m$ は、より大きな変化量 $Rsh300$ によって補正された Rsh が用いられ、その後は、 $Rsh300$ を一定の割合で小さくした値によって Rsh が用いられることになる。

【0066】

ところで、このような補正を行う場合、一般的にいて、誤った値を有する距離係数 R は、図 8b に示すように、誤った値から、時間と共に真の値に近づいてくることが期待できる。また、この場合の、補正変数 Rsh の変化量 $Rsh300$ は図 8b に示すように 0 に時間と共に近づき、その後値 0 を中心に安定することが期待できる。

10

【0067】

したが、補正変数 $Rsh300$ が、しばらくの間、0 周辺で安定して推移していれば、距離係数 R は真の値に近い値となっていると考えることができる。そこで、本実施例では、補正変数 $Rsh300$ が、しばらくの間、0 周辺で安定して推移している場合には、距離誤差 d として大きな大きな値が求めた場合でも、 $Rsh300$ の値を小さくすることにより、距離係数が真の値に近い値を有している場合に、 Rsh が大きく変化し、これにより距離係数が真の値から大きくはずれる方向に補正されてしまうことを防ぐ。

20

【0068】

以下、マイクロプロセッサ 214 によって実行される補正変数 Rsh を求めていく処理の詳細について説明する。

【0069】

まず、本装置が起動されると、まず、図 9 に示すように、以降の処理で使用する各種パラメータを初期化する。すなわち、距離誤差 d を 0 に、補正変数 Rsh を 0 に、修正量 k を 1 に、ポインタ pt を 0 に、補正レベル $lv(i)$ の全てを 0 に、変数 l を 0 に、 $flag$ を 0 に、補正レベル有効距離 $lv_l(i)$ の全てを 0 にセットする。ここで、 $lv(i)$ 、 $lv_l(i)$ は $imax$ 個の要素を持つ一次元の配列データである。

【0070】

30

これ以降は、図 10 に示す処理が、車両が $20m$ 走行する度に実行される。

【0071】

この処理では、各センサ 201、202、203 を用いて算出した走行距離や進行方位をセンサデータとして読み込む (ステップ 801)。そして次に、CD-ROM 205 からドライバ 206 に読み出された地図データを読み込む (ステップ 802)。

【0072】

次に、補正レベル $lv(i)$ 、補正レベル有効距離 $lv_l(i)$ の更新処理を行う (ステップ 803)。

【0073】

次に、 $flag$ が 0 か否かを判定する (ステップ 804)。 $flag$ は、距離誤差 d が求められ $Rsh300$ が算出されてから $300m$ 走行する間のみ値 1 となるフラグである。

40

【0074】

そして、フラグが 0 であれば、前述した距離誤差 d を求める。一方、 $flag$ が 1 である場合には、 $Rsh300$ が有効である期間、すなわち、距離誤差 d が求められ $Rsh300$ が算出されてから $300m$ 走行していない期間であるので、距離誤差 dno の算出 (ステップ 805) は行わない。

【0075】

次に、ステップ 805 で求めた距離誤差 d が 0 か否かを判定する (ステップ 807)。今回ステップ 805 で求めた距離誤差 d が 0 でない場合、および、 $Rsh300$ が算出されてから $300m$ 走行していない期間は、距離誤差 d が 0 ではないので、ステップ 808 の

50

R s h 算出処理を行う。なお、R s h 3 0 0 が算出されてから 3 0 0 m 走行していない期間は、この R s h 3 0 0 を求めるために用いた距離誤差 d が維持されつづけるので距離誤差 d は 0 とならない。R s h 算出処理（ステップ 8 0 8）では、前述したような、距離誤差 d が求められたときの R s h 3 0 0 の算出と R s h の変更と、R s h 3 0 0 が算出されてから 3 0 0 m 走行した後の R s h の変更の処理等を行う。

【0076】

以下、各処理の詳細について説明する。

【0077】

まず、ステップ 8 0 5 の距離誤差 d の算出処理について説明する。

【0078】

図 1 1 に、この処理の手順を示す。

【0079】

図示するように、この処理では、まず、車両が右左折したか否かを判定し（ステップ 1 1 0 1）、距離誤差が存在するか否かを判定し（ステップ 1 1 0 2）、距離誤差があれば、先に説明した距離誤差 d を算出する（ステップ 1 1 0 3）。ただし、このとき、車両がカーブを内回りしたときに距離誤差 d の符号は負とし、車両がカーブを外回りしたときに距離誤差 d の符号は正とする。

【0080】

一方、車両が右左折していない場合、または、距離誤差が存在しない場合には、そのまま処理を終了する。

【0081】

次に、図 1 0 のステップ 8 0 8 の補正変数 R s h の算出処理について説明する。

【0082】

図 1 2 に、この補正変数 R s h の算出処理の処理手順を示す。

【0083】

この処理では、まず、f l a g が 0 か否かを判定する（ステップ 1 2 0 0）。f l a g が 0 であるときは、R s h 3 0 0 が算出されてから既に 3 0 0 m 以上走行し、かつ、0 でない距離誤差 d が求められたときであるので（図 1 0 参照）、まず、修正量 k を求め（ステップ 1 2 0 1）、距離誤差 d に応じた R s h 3 0 0 を求める。

【0084】

この補正変数 R s h 3 0 0 は、距離誤差 d より次のようにして求める。すなわち、R s h 3 0 0 ' の正負を図 1 3 に示す対応によって定め、R s h 3 0 0 の絶対値を距離誤差 d の絶対値に、予め定めた係数 j を乗じて求める。

【0085】

そして、その後、 $R s h 3 0 0 = k \times R s h 3 0 0 '$ に従って、R s h 3 0 0 ' を修正し、 $R s h = R s h 0 + R s h 3 0 0$ に従い、R s h を補正する（ステップ 1 2 0 3）。ただし、R s h が変更される前に、その時点の R s h を R s h 0 として記憶しておく（ステップ 1 2 0 2）。

【0086】

そして、次に、算出した R s h 3 0 0 の大きさの応じた値を、l v (i) として登録する l v (i) 登録処理を行い（ステップ 1 2 0 4）、前述した f l a g を 1 に、R s h 3 0 0 が算出されてからの走行距離を表す変数 l を 0 に設定し（ステップ 1 2 0 5）、処理を終了する。変数 l は、R s h 3 0 0 が算出されてから 3 0 0 m 走行したか否かを判定するために用いる。

【0087】

さて、図 1 2 に戻り、ステップ 1 2 0 0 で f l a g が 0 でないと判定された場合には、まだ、R s h 3 0 0 が有効な期間であるので、R s h 3 0 0 が算出されてからの走行距離を表す l が 3 0 0 m を超えているかを判定し（ステップ 1 2 0 6）、超えていなければ l は、R s h 3 0 0 が算出されてから現在までの走行距離を表すよう更新し（ステップ 1 2 0 7）、処理を終了する。

10

20

30

40

50

【0088】

一方、 l が300mを超えていれば、距離誤差 d 、 l 、 $flag$ を0にリセットし（ステップ1209）、 Rsh を更新する。 Rsh は、 $Rsh300$ に1より小さな正の定数を乗じた値を、記憶しておいた $Rsh0$ （前回求めた $Rsh300$ 算出前の Rsh ）に加算することにより求める（ステップ1210）。この処理により、 Rsh は、より絶対値の小さな値に変更されることになる。

【0089】

なお、 $Rsh300$ として、 $Rsh300$ が算出されたから300の間以外に用いられる Rsh を無視できる程度に大きな値を用いることとし、 $Rsh300$ を求めてから300m間は、 $R = R0 + Rsh300$ とするようにしてもよい。

10

【0090】

さて、このような処理において、補正変数 $Rsh300$ が、しばらくの間、0周辺で推移している場合には、距離誤差 d として大きな値が求まった場合でも、 Rsh が大きく変化しないようにするための処理が、図12における、ステップ1201の修正量 k 算出処理と、ステップ1204の補正レベル $lv(i)$ の登録処理と、図10のステップ803の補正レベル $lv(i)$ 、補正レベル有効距離 $lv_l(i)$ の更新処理である。これらの処理では、補正変数 $Rsh300$ が、しばらくの間、0周辺で推移している場合には、補正変数 Rsh が大きく変化しないようにするため、ステップ1201の修正量 k 算出処理で算出される修正量 k の値が小さくなるようにする。

【0091】

20

以下、これらの処理の詳細について説明する。

【0092】

図14に、補正レベル $lv(i)$ の登録処理（図12のステップ1204）の処理手順を示す。

【0093】

この処理は、図12のステップ1204において $Rsh300$ が求められた後に、必ず実行される。

【0094】

図示するように、この処理は、まず、現在のポインタ pt の値を i とし（ステップ1000）、 $Rsh300$ に所定の定数 a を乗じた値を、配列データである補正レベル lv の i 番目の要素データ $lv(i)$ とする（ステップ1001）。また、配列データである補正レベル有効距離 lv の i 番目の要素データ $lv_l(i)$ に距離定数 b を設定し（ステップ1002）、その後、 $lv(i)$ 、 $lv_l(i)$ をメモリ215に登録する（ステップ）。ただし、図12ステップ1203で修正量 k を乗じる前の $Rsh300$ に所定の定数 a を乗じた値を、配列データである補正レベル lv の i 番目の要素データ $lv(i)$ としてもよい。

30

【0095】

そして、次の処理のためにポインタ pt を1増加させておく（ステップ1005）。ただし、1増加させた値が、補正レベル lv 、補正レベル有効距離 lv_l の要素数より大きくなってしまったら（ステップ1005）、ポインタ pt を0とする（ステップ1006）。

40

【0096】

以上の処理によって、順次求められた $Rsh300$ に比例した値が、それぞれ、順次、補正レベル $lv(0)$ 、 $lv(1)$ 、 $lv(2)$ 、...、 $lv(imax)$ 、 $lv(0)$ 、 $lv(1)$ 、...の順に格納されることになる。したがって、配列データである補正レベル lv には、最近求められた、最大 $imax$ 個の $Rsh300$ の値に各々比例した、最大 $imax$ 個の補正レベルが格納されることになる。

【0097】

ここで、ある補正レベル有効距離 $lv_l(n)$ は、対応する補正レベル $lv(n)$ が求められてから、現在まで、距離 b 以上走行したか否かを管理するために用いられる。すな

50

わち、補正レベル有効距離 $lv_l(i)$ は、次に述べる補正レベル $lv(i)$ 、補正レベル有効距離 $lv_l(i)$ の更新処理によって、現在より後、どのくらい走行すると補正レベル $lv(i)$ が求められてから、距離 b 走行したことになるかを表すように更新される。また、この処理では、ある補正レベル $lv(n)$ が求められてから、距離 b 以上走行したことを、対応する補正レベル有効距離 $lv_l(n)$ が表すようになったら、補正レベル $lv(n)$ を削除する。

【0098】

以下、この補正レベル $lv(i)$ 、補正レベル有効距離 $lv_l(i)$ の更新処理（図10ステップ707）について説明する。

【0099】

図15に、この処理の処理手順を示す。

【0100】

この処理は、図10に示すように20m走行する毎に必ず実行される。

【0101】

さて、この処理では、まず、 i を0に設定し（ステップ1601）、補正レベル有効距離 $lv_l(i)$ の値を更新する。すなわち、補正レベル有効距離 $lv_l(i)$ の値を20m分減ずる（ステップ1602）。したがい、補正レベル有効距離 $lv_l(i)$ の値は、 b より順次20m分減っていき、補正レベル $lv(i)$ が求められた後に距離 b 以上走行した時点で0以下の値となる。図16は、補正レベル lv 、補正レベル有効距離 lv_l の具体例について示したものである。この例は、補正レベル $lv(0)$ が求められてから現在までに距離 $L0$ 走行しており、補正レベル $lv(1)$ が求められてから現在までに距離 $(L1)$ 走行している場合の例である。この例では、補正レベル $lv(3)$ は今求められたばかりのものである。

【0102】

そして、次に、補正レベル有効距離 $lv_l(i)$ の値が0以下となったか否かを判定し（ステップ1603）、0以下となっていれば、補正レベル $lv(i)$ 、補正レベル有効距離 $lv_l(i)$ を共に0に設定する。一方、0以下となっていなければこの処理は行わない。

【0103】

そして、次に、 i が $imax$ と等しくなったかを判定し、等しければ配列データである補正レベル lv の全ての要素データ、配列データである補正レベル有効距離 lv の全ての要素データについての処理を終了したことになるので処理を終了する。 i が $imax$ と等しくなければ、 i を1増加させ（ステップ1606）、以上の処理を繰り返す。

【0104】

この処理の結果、配列データである補正レベル lv には、常に現在地点より距離 b 走行する前の地点から、現在地点までの間に求められた各補正変数数 $Rsh300$ に各々比例した補正レベルのみが格納されることになる。

【0105】

ただし、近似的には、常に現在までの一定期間中に求められた各補正変数数 Rsh に各々比例した補正レベルのみが格納されるようにしてもよい。

【0106】

したがい、補正レベル lv より、補正変数 $Rsh300$ が、最近、どのような値をとっていたかを判定することができ、その判定結果に応じて修正量 k を定めれば、しばらくの間、0周辺で推移している場合には、 $Rsh300$ を小さくすることにより Rsh が大きく変化しないようにすることができる。

【0107】

このように修正量 k を求めるのが、図12ステップ1201の修正量 k 算出処理である。以下、この修正量 k 算出処理について説明する。

【0108】

図17に、この修正量 k 算出処理の処理手順を示す。

10

20

30

40

50

【0109】

図12に示すように、この処理は、Rsh300を求める場合に、補正変数Rsh300を求める直前に必ず実行される。

【0110】

図示するように、この処理では、まず、配列データである補正レベルlvの各要素データの値の合計addを求める（ステップ1801）。次に、配列データである補正レベルlvの各要素データの値の絶対値の合計add_absを求める（ステップ1802）。

【0111】

そして、これらの値より修正量kを算出する（ステップ1803）。すなわち、合計addの絶対値に応じて図18に示すように値nを求め、 $m = add_abs - n$ に従い値mを求め、このmの値より図19に示すように修正量kを求める。ここで、値nは合計addの絶対値が、しきい値（図19では5）より小さい場合、すなわちRsh300が値0を中心として上下に均等に変動しているとき（Rsh300が安定しているとき）に、より大きな値をとるように定めている。また、add_absは、Rsh300が0周辺で推移しているときに小さい値となる。したがって、mはRsh300が0周辺で安定して推移しているときに、より小さな値となる。また、kはmが小さいほうが小さい値ととるように定めている。したがって、修正量kは、Rsh300が0周辺で安定して推移しているときに小さな値となる。

【0112】

したがって、前述したように修正量kによって、距離誤差dより求まるRsh300'を補正し、補正したRsh300によって補正変数Rshを求めれば、Rsh300が、しばらくの間、0周辺で安定して推移している場合、すなわちRが安定して推移している場合には、Rshが大きく変化することないようにすることができ、距離係数Rが真の値に近い値を有している場合に、距離係数が真の値から大きくはずれる方向に補正されてしまうことを防ぐことができる。

【0113】

【発明の効果】

以上のように、本発明によれば、距離係数が真の値に近い値を有している場合に、距離係数が真の値から大きくはずれる方向に補正されてしまうことを防ぐことができる現在位置算出装置を提供することができる。

【図面の簡単な説明】

【図1】本発明の一実施例に係る現在位置算出装置の構成を示すブロック図である。

【図2】本発明の一実施例において行う地図および現在位置の表示例を示す図である。

【図3】本発明の一実施例において行う進行方位および距離の算出処理の手順を示すフローチャートである。

【図4】本発明の一実施例において行う現在位置の算出処理の手順を示すフローチャートである。

【図5】本発明の一実施例において行う現在位置表示処理の手順を示すフローチャートである。

【図6】本発明の一実施例において用いる地図データにおける道路の表現形式を示す図である。

【図7】本発明の一実施例において用いる距離誤差dを示す図である。

【図8】本発明の一実施例において用いる距離係数と補正変数の変動のようすを示す図である。

【図9】本発明の一実施例において行う補正変数を補正する処理の手順を示す第1のフローチャートである。

【図10】本発明の一実施例において行う補正変数を補正する処理の手順を示す第2のフローチャートである。

【図11】本発明の一実施例において行う補正変数を補正する処理の手順を示す第3のフローチャートである。

10

20

30

40

50

【図 1 2】本発明の一実施例において行う補正変数を補正する処理の手順を示す第 4 のフローチャートである。

【図 1 3】本発明の一実施例において用いる距離誤差 d と補正変数 R_{sh} の正負の関係を示す図である。

【図 1 4】本発明の一実施例において行う補正変数を補正する処理の手順を示す第 5 のフローチャートである。

【図 1 5】本発明の一実施例において行う補正変数を補正する処理の手順を示す第 6 のフローチャートである。

【図 1 6】本発明の一実施例用いる補正レベル $l_v(i)$ とほせいレベル有効距離 $l_{v_1}(i)$ の例を示す図である。

10

【図 1 7】本発明の一実施例において行う補正変数を補正する処理の手順を示す第 7 のフローチャートである。

【図 1 8】本発明の一実施例において用いる変数 n と変数 a_{dd} の絶対値との対応を示す図である。

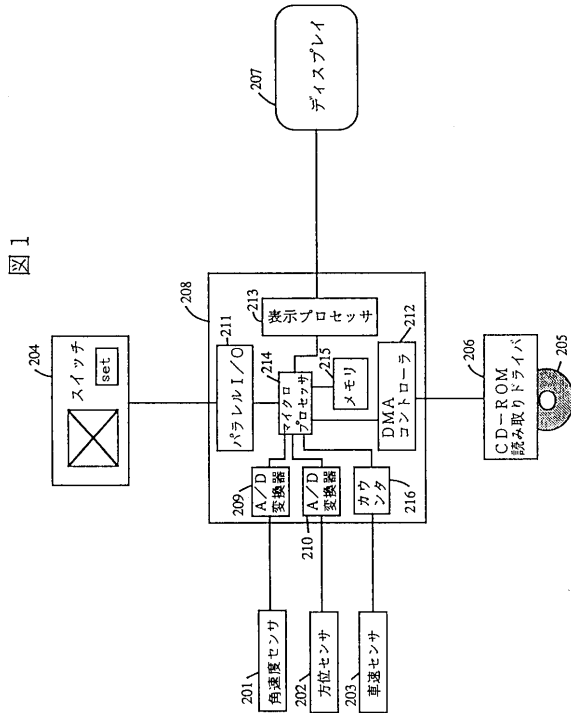
【図 1 9】本発明の一実施例において用いる変数 m と修正量 k との対応を示す図である。

【符号の説明】

201	角速度センサ
202	方位センサ
203	車速センサ
204	スイッチ
205	CD-ROM
206	CD-ROMドライバ
207	ディスプレイ
208	DMAコントローラ
214	マイクロプロセッサ
215	メモリ

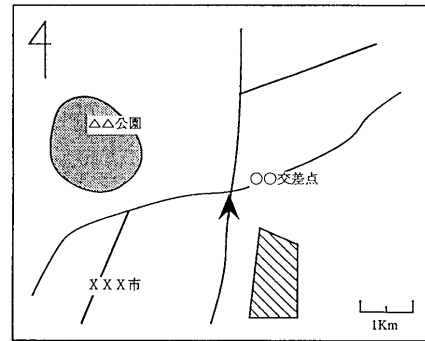
20

【図 1】



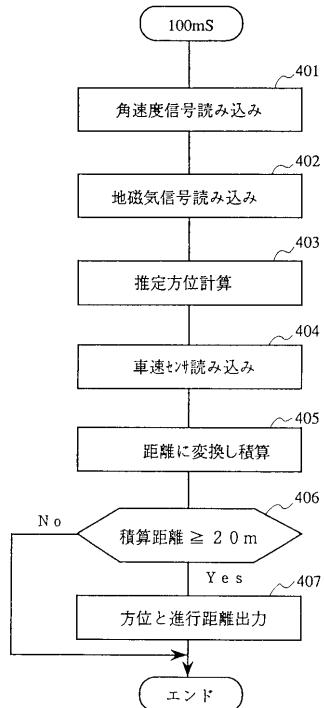
【図 2】

図 2



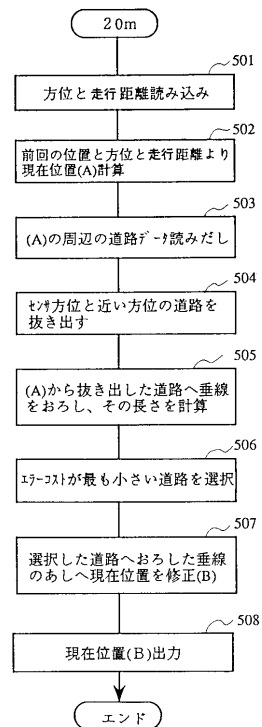
【図 3】

図 3

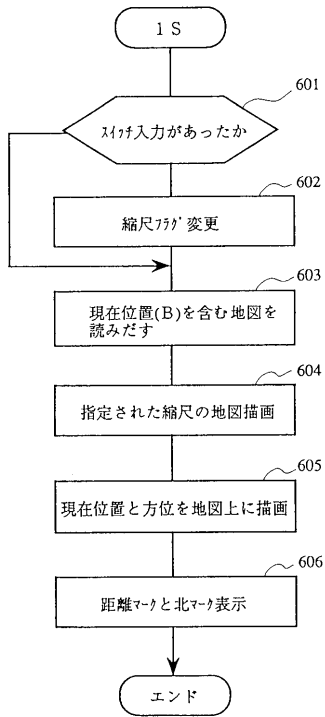


【図 4】

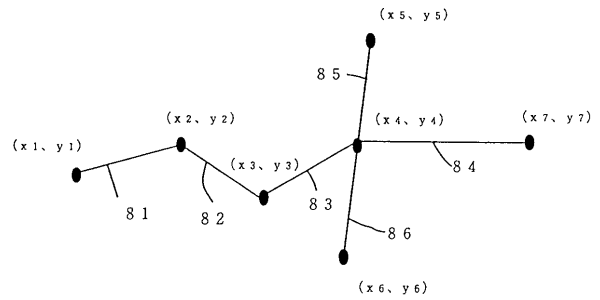
図 4



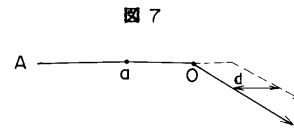
【図 5】
図5



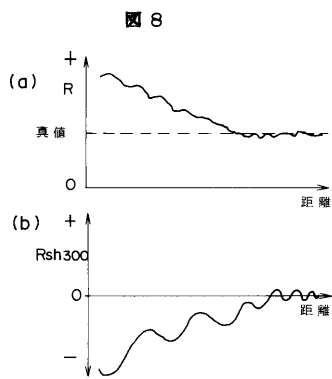
【図 6】
図6



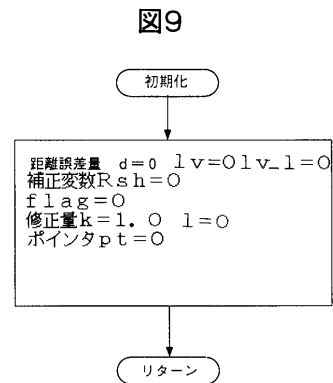
【図 7】



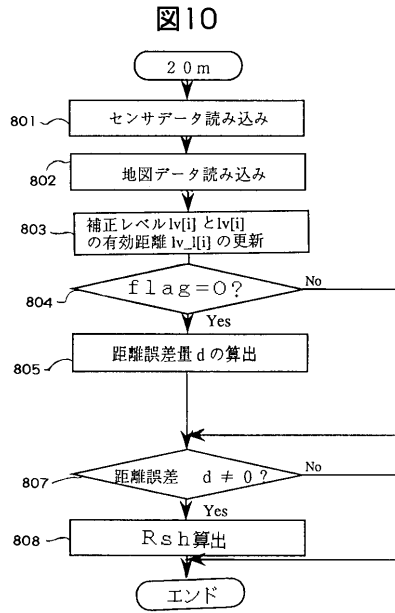
【図 8】



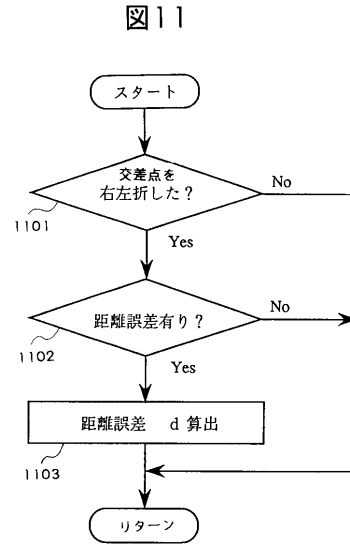
【図 9】



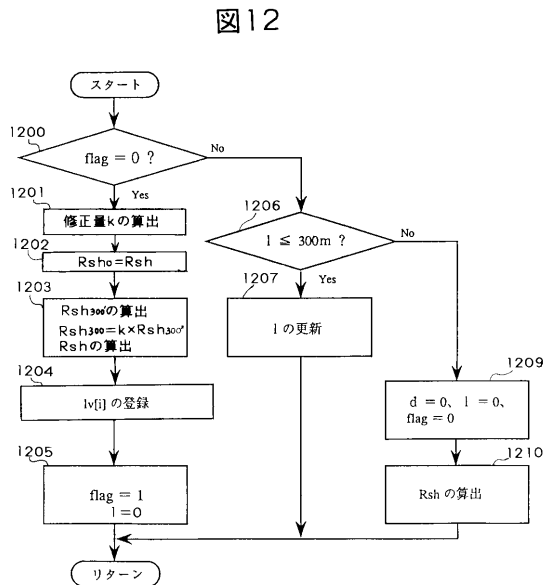
【図 10】



【図 11】



【図 12】



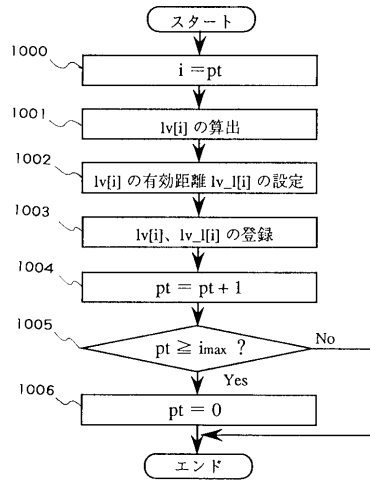
【図 13】

図13

距離誤差量 d	Rsh300'
正	負
負	正

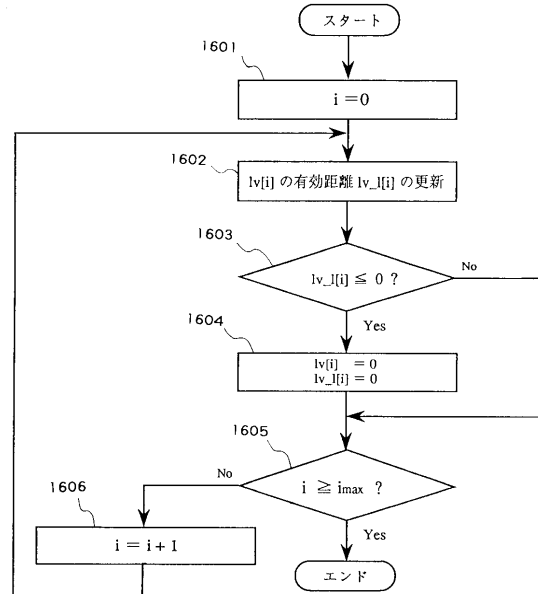
【図 14】

図14



【図 15】

図15



【図 16】

図16

i	lv[i]	lv_l[i]
0	a0	b - L0
1	a1	b - L1
2	a2	b
3	0	0
⋮	⋮	⋮

【図 18】

図18

add (超～以下)	n
0 ~ 5	10
5 ~	0

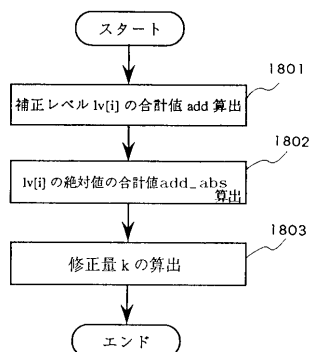
【図 19】

図19

m (超～以下)	k
~ 10	0.4
10 ~ 20	0.6
20 ~ 30	0.8
30 ~ 40	0.9
40 ~	1.0

【図 17】

図17



フロントページの続き

(58)調査した分野(Int.Cl.⁷, DB名)

G01C 21/00 -21/36

G01C 23/00 -25/00

G09B 23/00 -29/14

G08G 1/00 - 9/02