

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5493097号
(P5493097)

(45) 発行日 平成26年5月14日(2014.5.14)

(24) 登録日 平成26年3月14日(2014.3.14)

(51) Int.Cl.

F I

G05D 1/02 (2006.01)

G05D 1/02 H

B25J 19/02 (2006.01)

B25J 19/02

B25J 5/00 (2006.01)

B25J 5/00 A

請求項の数 7 (全 40 頁)

(21) 出願番号 特願2010-87682(P2010-87682)
 (22) 出願日 平成22年4月6日(2010.4.6)
 (65) 公開番号 特開2011-221631(P2011-221631A)
 (43) 公開日 平成23年11月4日(2011.11.4)
 審査請求日 平成25年1月22日(2013.1.22)

特許法第30条第1項適用 「The 2009 IEEE/RSJ International Conference on Robots and Intelligent Systems (IROS 2009)」 日本ロボット学会(共催) 2009年10月11~15日開催

(73) 特許権者 393031586
 株式会社国際電気通信基礎技術研究所
 京都府相楽郡精華町光台二丁目2番地2
 (74) 代理人 100090181
 弁理士 山田 義人
 (72) 発明者 ディラン フェアチャイルド グラス
 京都府相楽郡精華町光台二丁目2番地2
 株式会社国際電気通信基礎技術研究所内
 (72) 発明者 神田 崇行
 京都府相楽郡精華町光台二丁目2番地2
 株式会社国際電気通信基礎技術研究所内

審査官 柿崎 拓

最終頁に続く

(54) 【発明の名称】 ロボット自己位置同定システム

(57) 【特許請求の範囲】

【請求項1】

自身の移動に関して自身の位置データを少なくとも含む状態データを出力できるロボットを、それが存在する空間内において自己位置同定する、ロボット自己位置同定システムであって、

前記ロボットから順次取得した複数の状態データを記憶する状態データ記憶手段、

前記空間内に存在する実体をセンシングする環境センサ、

前記環境センサの出力に基づいて前記実体の位置を検出する位置検出手段、

前記位置検出手段によって検出された実体の複数の位置データを記憶する位置データ記憶手段、

前記状態データ記憶手段によって記憶された前記ロボットの複数の位置データおよび前記位置データ記憶手段によって記憶された前記実体の複数の位置データに基づいて、前記ロボットと前記実体とを関連付ける関連付手段、および

前記ロボットと関連付けられた前記実体の位置データに基づいて、前記状態データに含まれる位置データを補正する補正データを、前記ロボットに送信する送信手段を備え、

前記ロボットは、前記補正データに従って、自身の位置データを補正する、ロボット自己位置同定システム。

【請求項2】

前記状態データは、自身の角度データをさらに含み、

前記状態データ記憶手段は、前記角度データを含む状態データを記憶し、

10

20

前記状態データ記憶手段によって記憶された前記ロボットの複数の状態データおよび前記位置データ記憶手段によって記憶された前記実体の複数の位置データに基づいて、次の位置における角度を推定する角度推定手段、および

前記角度推定手段によって推定された角度および状態データ記憶手段によって記憶された状態データに含まれる角度データに基づいて、前記状態データに含まれる角度データを補正する角度補正データを算出する角度補正算出手段をさらに備え、

前記送信手段は、前記補正データおよび前記角度補正データを、前記ロボットに送信する第1送信手段を含み、

前記ロボットは、前記補正データおよび前記角度補正データに従って、自身の位置データおよび角度データを補正する、請求項1記載のロボット自己位置同定システム。

10

【請求項3】

前記角度推定手段は、角度を推定するときに、推定角度誤差をあわせて算出し、

前記推定角度誤差を補正精度として記憶する記憶手段をさらに備え、

前記送信手段は、前記記憶手段によって記憶された補正精度が低いとき、前記補正データのみを前記ロボットに送信する第2送信手段を含む、請求項2記載のロボット自己位置同定システム。

【請求項4】

前記角度推定手段は、前記状態データ記憶手段によって記憶された前記ロボットの複数の状態データおよび前記位置データ記憶手段によって記憶された前記実体の位置データに基づいて、前記ロボットの角度を観測する角度観測手段、および前記角度観測手段によって観測された角度の観測角度誤差を算出する観測角度誤差算出手段を含み、

20

前記角度推定手段は、前記観測された角度、前記観測角度誤差、前回推定された角度および前回算出された推定角度誤差に基づいて、次の位置の角度を推定し、推定角度誤差を算出する、請求項3記載のロボット自己位置同定システム。

【請求項5】

前記ロボットと前記実体との関連付けを維持するかを判断する判断手段、および

前記判断手段によって維持しないと判断されたとき、前記ロボットと前記実体との関連付けを解除する解除手段をさらに備える、請求項1ないし4のいずれかに記載のロボット自己位置同定システム。

【請求項6】

30

前記空間には、少なくとも2つ以上の実体が存在し、

前記関連付手段は、前記ロボットと前記実体との関連付けが解除されてから所定時間以内のとき、前記ロボットと各実体との第1速度差を算出する第1速度差算出手段、前記第1速度差算出手段によって算出された第1速度差が第1所定速度より大きいかを判断する第1速度判定手段、前記ロボットと各実体との距離が所定距離より離れているかを判定する距離判定手段、および前記第1速度判定手段によって前記ロボットとの第1速度差が前記第1所定速度以下と判定され、かつ前記距離判断手段によって前記ロボットとの距離が所定距離以内と判定された実体のうち、前記ロボットに最も近い実体と前記ロボットとを関連付ける第1関連付手段を含む、請求項5記載のロボット自己位置同定システム。

【請求項7】

40

前記空間には、少なくとも2つ以上の実体が存在し、

前記関連付手段は、前記ロボットと前記実体との関連付けが解除されてから所定時間が経過したとき、前記ロボットと各実体との第2速度差を算出する第2速度差算出手段、前記第2速度差算出手段によって算出された第2速度差が第2所定速度より大きいかを判定する第2速度判定手段、および前記第2速度判定手段によって前記ロボットの第2速度差が前記第2所定速度以下と判定された実体が1つだけであるとき、その実体と前記ロボットとを関連付ける第2関連付手段を含む、請求項5または6記載のロボット自己位置同定システム。

【発明の詳細な説明】

【技術分野】

50

【 0 0 0 1 】

この発明は、ロボット自己位置同定システムに関し、特にたとえば、ロボットが存在する空間においてそのロボットを自己位置同定する、ロボット自己位置同定システムに関する。

【 背景技術 】

【 0 0 0 2 】

非特許文献 1 には、個別の I D が割り振られた RFID タグが発する電波を受信し、電波強度を検出することで、RFID タグを付けた人物の位置を推定し、人物を識別する技術が開示されている。

【 0 0 0 3 】

また、特許文献 1 には、位置検出装置が検出する位置情報を、適正に補正する位置補正装置が開示されている。位置検出装置は、GPS を利用しており、たとえば車両に取り付けられる。そして、位置補正装置は、GPS の測位に基づいて推定される推定車両速度と車両速度とから、車両の位置情報を補正する。

【 非特許文献 1 】 神田 崇行, 塩見 昌裕, 野村 竜也, 石黒 浩, 荻田 紀博, "RFID タグを用いた科学館来館者の移動軌跡の分析" (情報処理学会論文誌 Vol.49 No.5 P.1727-1742)

【 特許文献 1 】 特開 2 0 0 2 - 3 5 0 1 5 7 号公報 [G01C 21/00, G01S 5/14, G08G, 1/0 969, G09B 29/00, G09B 29/10]

【 発明の概要 】

【 発明が解決しようとする課題 】

【 0 0 0 4 】

近年、様々なロボット自己位置同定システムが開発されている。そして、その一例として、非特許文献 1 の技術を応用し、RFID タグをロボットに取り付けることでロボットを自己位置同定するシステムが開発されている。

【 0 0 0 5 】

ところが、上記したシステムではロボットに RFID タグを取り付け、さらには環境に RFID タグを受信するアンテナを設けなければならず、容易にロボットを自己位置同定することができない。また、広く一般的に利用されている GPS を利用すれば位置を検出することができるが、GPS の位置検出精度は低く、特許文献 1 で開示された技術を利用して位置補正を行ったとしても、ロボット自己位置同定システムには利用できない。

【 0 0 0 6 】

それゆえに、この発明の主たる目的は、新規な、ロボット自己位置同定システムを提供することである。

【 0 0 0 7 】

この発明の他の目的は、容易にロボットを自己位置同定することができる、ロボット自己位置同定システムを提供することである。

【 課題を解決するための手段 】

【 0 0 0 8 】

この発明は、上記の課題を解決するために、以下の構成を採用した。なお、括弧内の参照符号および補足説明等は、この発明の理解を助けるために記述する実施形態との対応関係を示したものであって、この発明を何ら限定するものではない。

【 0 0 0 9 】

第 1 の発明は、自身の移動に関して自身の位置データを少なくとも含む状態データを出力できるロボットを、それが存在する空間内において自己位置同定する、ロボット自己位置同定システムであって、ロボットから順次取得した複数の状態データを記憶する状態データ記憶手段、空間内に存在する実体をセンシングする環境センサ、環境センサの出力に基づいて実体の位置を検出する位置検出手段、位置検出手段によって検出された実体の複数の位置データを記憶する位置データ記憶手段、状態データ記憶手段によって記憶されたロボットの位置データおよび位置データ記憶手段によって記憶された実体の位置データに基づいて、ロボットと実体とを関連付ける関連付手段、およびロボットと関連付けられた

10

20

30

40

50

実体の位置データに基づいて、状態データに含まれる位置データを補正する補正データを、ロボットに送信する送信手段を備え、ロボットは、補正データに従って、自身の位置データを補正する、ロボット自己位置同定システムである。

【 0 0 1 0 】

第1の発明では、ロボット自己位置同定システム(100)は、空間(検出領域F)に存在するロボット(18a, 18b, 18c)を、その空間内で自己位置同定する。また、ロボットは、自身の移動に関して自身の位置データを少なくとも含む状態データを出力できる。

【 0 0 1 1 】

状態データ記憶手段(20, S9)は、たとえばメモリ(22)のバッファにロボットから順次取得した状態データを記憶する。環境センサ(12)は、たとえばLRF(Laser Range Finder)などであり、空間内に存在する人間やロボットなどの実体(Ea, Eb, Ec, Ed)をセンシングする。位置検出手段(20, S89)は、環境センサの出力に対して、たとえばパーティクルフィルタを利用することで実体の位置を検出する。位置データ記憶手段(20, S91)は、たとえば状態データ記憶手段と同様、メモリのバッファに、検出された実体の位置データを記憶する。関連付手段(20, S49)は、バッファに記憶されたロボットの複数の位置データおよび実体の複数の位置データから、たとえば速度を求めることで、ロボットと実体とを関連付ける。そして、ロボットの位置データおよび実体の位置データは、空間における同じ平面座標系の値で示される。

【 0 0 1 2 】

送信手段(20, S15, S19)は、ロボットと関連付けられた実体の位置データに基づいて、たとえば空間の位置を示す座標データを、補正データとしてロボットに送信する。そして、ロボットは、たとえば補正データを受け取ると、自身の位置データを補正する。

【 0 0 1 3 】

第1の発明によれば、容易に設置可能な環境センサと、ロボットが一般的に出力可能な状態データとを利用することで、ロボットの現在位置を容易に検出できるようになる。そのため、検出された実体の位置をロボットに送信することで、容易にロボットを自己位置同定することができるようになる。

【 0 0 1 4 】

第2の発明は、第1の発明に従属し、状態データは、自身の角度データをさらに含み、状態データ記憶手段は、角度データを含む状態データを記憶し、状態データ記憶手段によって記憶されたロボットの複数の状態データおよび位置データ記憶手段によって記憶された実体の複数の位置データに基づいて、次の位置における角度を推定する角度推定手段、および角度推定手段によって推定された角度および状態データ記憶手段によって記憶された状態データに含まれる角度データに基づいて、状態データに含まれる角度データを補正する角度補正データを算出する角度補正算出手段をさらに備え、送信手段は、補正データおよび角度補正データを、ロボットに送信する第1送信手段を含み、ロボットは、補正データおよび角度補正データに従って、自身の位置データおよび角度データを補正する。

【 0 0 1 5 】

第2の発明では、状態データ記憶手段は、ロボット自身の角度データをさらに含む、状態データを記憶する。角度推定手段(20, S261 - S277)は、たとえば実体の複数の位置データおよびロボットの複数の状態データからロボットの位置を予測することで、次の位置のロボットの角度を推定する。角度補正算出手段(20, S279)は、たとえば現在の状態データに含まれる角度データと推定された角度(θ^k)との角度差を、角度補正データとして算出する。第1送信手段(20, S19)は、補正データおよび角度補正データを、ロボットに送信する。

【 0 0 1 6 】

第2の発明によれば、ロボットの角度も補正することで、自己位置同定の精度を高めることができる。

【 0 0 1 7 】

第3の発明は、第2の発明に従属し、角度推定手段は、角度を推定するときに、推定角度誤差をあわせて算出し、推定角度誤差を補正精度として記憶する記憶手段をさらに備え、送信手段は、記憶手段によって記憶された補正精度が低いとき、補正データのみをロボットに送信する第2送信手段を含む。

【 0 0 1 8 】

第3の発明では、角度推定手段は、角度を推定すると共に、推定角度誤差 ($\sigma^2_{\theta^k}$) を算出する。記憶手段 (20, S281) は、推定角度誤差を補正精度として、バッファに記憶する。推定された角度の誤差が大きければ、第2送信手段 (20, S15) は補正データのみをロボットに送信する。

10

【 0 0 1 9 】

第3の発明によれば、推定された角度の誤差を、補正精度として利用することで、ロボットの角度の補正が可能であるかを、判断できるようになる。

【 0 0 2 0 】

第4の発明は、第3の発明に従属し、角度推定手段は、状態データ記憶手段によって記憶されたロボットの複数の状態データおよび位置データ記憶手段によって記憶された実体の位置データに基づいて、ロボットの角度を観測する角度観測手段、および角度観測手段によって観測された角度の観測角度誤差を算出する観測角度誤差算出手段を含み、角度推定手段は、観測された角度、観測角度誤差、前回推定された角度および前回算出された推定角度誤差に基づいて、次の位置の角度を推定し、推定角度誤差を算出する。

20

【 0 0 2 1 】

第4の発明では、たとえば、角度観測手段 (20, S261 - S265) は、複数の状態データから次の位置を予測して予測角度を算出し、実体の位置データから検知角度を算出する。そして、角度観測手段は、検知角度から予測角度を減算することで、ロボットの角度を観測 (算出) する。観測角度誤差算出手段 (20, S267) は、たとえば、観測における位置の誤差 ($\sigma^2_{p_o}$) に対する実体の位置データの商に近似することで、観測角度誤差を算出する。

【 0 0 2 2 】

たとえば、角度推定手段は、観測された角度、観測角度誤差、前回推定された角度および前回算出された推定角度誤差に基づいて、カルマンゲイン (K_k) を求めることで、次の位置の角度を推定し、推定角度誤差を算出する。

30

【 0 0 2 3 】

第4の発明によれば、カルマンフィルタを用いて、次の位置の角度を推定することができる。

【 0 0 2 4 】

第5の発明は、第1の発明ないし第4の発明のいずれかに従属し、ロボットと実体との関連付けを維持するかを判断する判断手段、および判断手段によって維持しないと判断されたとき、ロボットと実体との関連付けを解除する解除手段をさらに備える。

【 0 0 2 5 】

第5の発明では、判断手段 (20, S117, S119, S121) は、たとえばロボットと実体との速度差や距離などに基づいて、関連付けを維持するかを判断する。解除手段 (20, S125) は、たとえばロボットと実体との速度差が大きかったり、ロボットと実体との距離が離れていたりすると、ロボットと実体との関連付けを解除する。

40

【 0 0 2 6 】

第5の発明によれば、エラーの要因となるロボットと実体との関連付けを、解除できるようになる。

【 0 0 2 7 】

第6の発明は、第5の発明に従属し、空間には、少なくとも2つ以上の実体が存在し、関連付手段は、ロボットと実体との関連付けが解除されてから所定時間以内のとき、ロボットと各実体との第1速度差を算出する第1速度差算出手段、第1速度差算出手段によっ

50

て算出された第1速度差が第1所定速度より大きいかを判断する第1速度判定手段、ロボットと各実体との距離が所定距離より離れているかを判定する距離判定手段、および第1速度判定手段によってロボットとの第1速度差が第1所定速度以下と判定され、かつ距離判断手段によってロボットとの距離が所定距離以内と判定された実体のうち、ロボットに最も近い実体とロボットとを関連付ける第1関連付手段を含む。

【0028】

第6の発明では、空間には、たとえば4つの実体が存在することがある。第1速度差算出手段(20, S193)は、関連付けが解除されてから所定時間(5秒)以内のとき、たとえば5秒分のロボットの位置データおよび5秒分の実体の位置データから、第1速度差(RMS(Root Mean Square)速度差)を算出する。第1速度判定手段(20, S195)は、第1速度差が第1所定速度より大きいかを判断する。また、距離判定手段(20, S199)は、ロボットの位置データおよび実体の位置データから距離を求め、ロボットと各実体とが所定距離より離れているかを判定する。たとえば、第1関連付手段(20, S209)は、ロボットとの第1速度差が第1所定速度以下であり、かつロボットとの距離が所定距離以下の実体が複数ある場合、ロボットに最も近い実体とロボットとを関連付ける。

10

【0029】

第6の発明によれば、関連付けが解除されて間もなければ、ロボットと実体とを容易に関連付け直すことができる。

【0030】

20

第7の発明は、第5の発明または第6の発明に従属し、空間には、少なくとも2つ以上の実体が存在し、関連付手段は、ロボットと実体との関連付けが解除されてから所定時間が経過したとき、ロボットと各実体との第2速度差を算出する第2速度差算出手段、第2速度差算出手段によって算出された第2速度差が第2所定速度より大きいかを判定する第2速度判定手段、および第2速度判定手段によってロボットの第2速度差が第2所定速度以下と判定された実体が1つだけであるとき、その実体とロボットとを関連付ける第2関連付手段を含む。

【0031】

第7の発明では、第6の発明と同様、空間には4つの実体が存在することがある。たとえば、第2速度差算出手段(20, S233)は、関連付けが解除されてから所定時間が経過していれば、300秒分のロボットの位置データおよび300秒分の実体の位置データから、第2速度差(RMS速度差)を算出する。第2速度判定手段(20, S235)は、第2速度差が第2所定速度より大きいかを判定する。たとえば、第2関連付手段(20, S247)は、ロボットとの第2速度差が第2所定速度以下の実体が1つだけであれば、その実体とロボットとを関連付ける。

30

【0032】

第7の発明によれば、関連付けが解除されて所定時間が経過したとしても、ロボットと実体とを関連付け直すことができる。

【発明の効果】

【0033】

40

この発明によれば、容易に設置可能な環境センサおよびロボットが一般的に出力可能な状態データを利用して、ロボットの位置を検出することができるため、容易にロボットを自己位置同定することができる。

【0034】

この発明の上述の目的、その他の目的、特徴および利点は、図面を参照して行う以下の実施例の詳細な説明から一層明らかとなろう。

【図面の簡単な説明】

【0035】

【図1】図1はこの発明の一実施例のロボット自己位置同定システムの概要を示す図解図である。

50

【図 2】図 2 は図 1 に示す追跡サーバの電氣的な構成の一例を示す図解図である。

【図 3】図 3 は図 1 に示すロボットの外観を正面から見た図解図である。

【図 4】図 4 は図 1 に示すロボットの電氣的な構成の一例を示すブロック図である。

【図 5】図 5 は図 1 および図 2 に示す L R F の計測領域を示す図解図である。

【図 6】図 6 は図 1 および図 2 に示す L R F を利用して取得された実体の移動軌跡の一例を示す図解図である。

【図 7】図 7 は図 2 に示すメモリに記憶された実体の位置履歴データの構成の一例を示す図解図である。

【図 8】図 8 は図 2 に示すメモリに記憶されたロボットの状態履歴データの構成の一例を示す図解図である。

10

【図 9】図 9 は図 1 に示す追跡サーバのメモリに記憶される、R T M (Robot Tracking Model) データの一例を示す図解図である。

【図 10】図 10 は図 6 に示す実体の運動履歴および図 1 に示すロボットの運動履歴の変化を示すグラフである。

【図 11】図 11 は図 10 に示す実体の運動履歴およびロボットの運動履歴の R M S 誤差のグラフである。

【図 12】図 12 は図 2 に示すメモリに一時記憶される追跡実体リストの構成の一例を示す図解図である。

【図 13】図 13 は図 2 に示すメモリに記憶される関連付テーブルの構成の一例を示す図解図である。

20

【図 14】図 14 は図 1 に示すプロセッサによって実行される角度修正処理で利用される各数値の関係を示す図解図である。

【図 15】図 15 は図 1 に示す人間とロボットとの関連付けおよび関連付け解除の一例を示す図解図である。

【図 16】図 16 は図 2 に示すメモリのメモリマップの一例を示す図解図である。

【図 17】図 17 は図 16 に示すデータ記憶領域の一例を示す図解図である。

【図 18】図 18 は図 1 に示す追跡サーバのプロセッサの通信処理の一部を示すフロー図である。

【図 19】図 19 は図 1 に示す追跡サーバのプロセッサの通信処理の一部であって、図 18 に後続するフロー図である。

30

【図 20】図 20 は図 1 に示す追跡サーバのプロセッサの補正処理を示すフロー図である。

【図 21】図 21 は図 1 に示す追跡サーバのプロセッサの追跡処理の一部を示すフロー図である。

【図 22】図 22 は図 1 に示す追跡サーバのプロセッサの追跡処理の一部であって、図 21 に後続するフロー図である。

【図 23】図 23 は図 1 に示す追跡サーバのプロセッサの関連付解除処理を示すフロー図である。

【図 24】図 24 は図 1 に示す追跡サーバのプロセッサの関連付処理を示すフロー図である。

40

【図 25】図 25 は図 1 に示す追跡サーバのプロセッサのローカル関連付処理を示すフロー図である。

【図 26】図 26 は図 1 に示す追跡サーバのプロセッサのグローバル関連付処理を示すフロー図である。

【図 27】図 27 は図 1 に示す追跡サーバのプロセッサの角度補正処理を示すフロー図である。

【発明を実施するための形態】

【0036】

図 1 を参照して、この実施例のロボット自己位置同定システム 100 は、たとえば自律移動型のロボット（以下、単に「ロボット」と言う。）18a、18b、18c を、それ

50

らのロボットが存在する空間内で自己位置同定 (Localization (ローカライゼーション) と言うこともある。) するためのシステムである。なお、本実施例で言う「自己位置同定」とは、ロボット 18a, 18b, 18c (以下、区別しない場合には「ロボット 18」と言う。) が、その空間内における自身の位置、角度および速度などのデータを、自身のセンサおよび環境センサなどを利用して取得することを意味する。

【0037】

ロボット自己位置同定システム 100 は、追跡サーバ 10、遠隔操作装置 14、計画サーバ 16 およびロボット 18 から構成されている。そして、ロボット 18 は、ネットワーク 400 を介して、追跡サーバ 10、遠隔操作装置 14 および計画サーバ 16 に接続される。

10

【0038】

追跡サーバ 10 は、環境センサである、LRF 12a, 12b を含む複数の LRF を備える。LRF 12a, 12b は人間およびロボット 18 が行動できる場所 (環境) に設置される。そして、追跡サーバ 10 は LRF 12a, 12b を利用して、ロボット 18 および人間を含む実体をセンシングすることで、各実体の位置を検出する。たとえば、人間およびロボット 18 が行動できる場所とは、会社のフロア、博物館、ショッピングモールまたはアトラクション会場などである。なお、システムの管理者は、様々な場所において、LRF 12a, 12b を、容易に設置することができる。また、追跡サーバ 10 によって得られた位置データは、遠隔操作サーバ 14 および計画サーバ 16 に出力される。

【0039】

さらに、追跡サーバ 10 は、ローカライゼーション機能 (ローカライゼーションモジュールと言うこともある。) を備えている。そのため、追跡サーバ 10 は、ロボット 18 との通信を確立し、ロボット 18 が出力する状態データを受信 (取得) するとともに、ロボット 18 に対して位置および角度 (向き) を補正するための補正データを送信する。状態データは、ロボット 18 自身の位置データ、角度データおよび速度データを含み、ロボット 18 が備えるセンサの出力に基づいて作成される。

20

【0040】

遠隔操作装置 14 は、追跡サーバ 10 から得た位置データを表示する操作端末を含み、オペレータによって操作される。また、オペレータは、操作端末に表示されるロボット 18 や人間の位置に基づいて、同時に複数のロボット 18a, 18b, 18c を監視することができる。そして、オペレータは、必要に応じてロボット 18 を遠隔操作したり、動作命令を発行したりする。また、計画サーバ 16 は、追跡サーバ 10 から得た位置データを利用して、ロボット 18 が空間内に存在する人間とのコミュニケーション行動を行うように、経路を計算する。

30

【0041】

ロボット 18 は、相互作用指向のロボットでもあり、人間のようなコミュニケーションの対象との間で、身振り手振りのような身体動作および音声の少なくとも一方を含むコミュニケーション行動を実行する機能を備えている。そのため、ロボット 18 は、遠隔操作装置 14 または計画サーバ 16 の命令に基づいて、人間に接近したり、コミュニケーション行動を行ったりする。

40

【0042】

また、ロボット 18 は、先述した状態データを 0.1 秒 ~ 0.3 秒毎に出力する。さらに、各ロボット 18 は、文字列で構成されるロボット ID でそれぞれが識別 (区別) される。

【0043】

なお、ここでは簡単のため人間を 1 人、ロボット 18 を 3 台しか示していないが、追跡サーバ 10 は 2 人以上の人間および 3 台以上のロボット 18 を同時に検出することができる。また、追跡サーバ 10、遠隔操作装置 14 および計画サーバ 16 と、ネットワーク 400 との接続は無線接続であってもよいし、有線接続であってもよい。さらに、各サーバおよび装置間の接続も、有線接続であってもよいし、無線接続であってもよい。

50

【 0 0 4 4 】

図 2 は、追跡サーバ 1 0 の電氣的な構成を示すブロック図である。図 2 を参照して、追跡サーバ 1 0 は、L R F 1 2 a - 1 2 f およびプロセッサ 2 0 を含む。プロセッサ 2 0 は、R T C (Real Time Clock) 2 0 a を含んでおり、検出した位置データに対して、R T C 2 0 a から得られた時刻データを対応付けて、メモリ 2 2 のバッファに記憶する。

【 0 0 4 5 】

プロセッサ 2 0 は、マイクロコンピュータ或いはCPUとも呼ばれ、先述した L R F 1 2 a および L R F 1 2 b に加えて、L R F 1 2 c , L R F 1 2 d , L R F 1 2 e および L R F 1 2 f とそれぞれ接続される。さらに、プロセッサ 2 0 は、メモリ 2 2 、通信 L A N ボード 2 4 、無線通信装置 2 6 および外部出力 2 8 とともに接続される。なお、L R F 1 2 a - 1 2 f を区別する必要がない場合には、まとめて「L R F 1 2」と言う。

10

【 0 0 4 6 】

L R F 1 2 は、レーザーを照射し、物体（実体も含む）に反射して戻ってくるまでの時間から当該物体との距離を計測するものである。たとえば、トランスミッタ（図示せず）から照射したレーザーを回転ミラー（図示せず）で反射させて、前方を扇状に一定角度（たとえば、0 . 5 度）ずつスキャンする。ここで、L R F 1 2 としては、SICK社製のレーザーレンジファインダ（型式 LMS200）を用いることができる。このレーザーレンジファインダを用いた場合には、距離 8 m を ± 1 5 mm 程度の誤差で計測可能である。

【 0 0 4 7 】

メモリ 2 2 は、図示は省略をするが、ROM , HDD および RAM を含み、ROM および HDD には、追跡サーバ 1 0 の動作を制御するための制御プログラムが予め記憶される。たとえば、L R F 1 2 による人間の検出に必要なプログラムなどが記録される。また、RAM は、ワークメモリやバッファメモリとして用いられる。

20

【 0 0 4 8 】

通信 L A N ボード 2 4 は、たとえば DSP で構成され、プロセッサ 2 0 から与えられた送信データを無線通信装置 2 6 に与え、無線通信装置 2 6 は送信データを、ネットワーク 4 0 0 を介してロボット 1 8 に送信する。また、送信データは補正データなどである。一方、通信 L A N ボード 2 4 は、無線通信装置 2 6 を介してデータを受信し、受信したデータをプロセッサ 2 0 に与える。また、受信データはロボット 1 8 が出力する状態データなどである。なお、受信した状態データは、メモリ 2 2 のバッファに記憶される。

30

【 0 0 4 9 】

外部接続 2 8 は、遠隔操作装置 1 4 および計画サーバ 1 6 と接続するためのポートであり、たとえば L A N ケーブル、SCSI ケーブルまたは USB ケーブルなどの有線ケーブルが接続される。そして、プロセッサ 2 0 は、実体の位置データを外部出力 2 8 に与え、外部出力 2 8 は位置データを遠隔操作装置 1 4 および計画サーバ 1 6 に出力する。

【 0 0 5 0 】

図 3 はこの実施例のロボット 1 8 の外観を示す正面図である。図 3 を参照して、ロボット 1 8 は台車 3 0 を含み、台車 3 0 の下面にはロボット 1 8 を自律移動させる右車輪 3 2 a 、左車輪 3 2 b （特に区別しない場合には、「車輪 3 2」と言う）および 1 つの従輪 3 4 が設けられる。右車輪 3 2 a および左車輪 3 2 b は、右車輪モータ 3 6 a および左車輪モータ 3 6 b （図 4 参照）によってそれぞれ独立に駆動され、台車 3 0 すなわちロボット 1 8 を前後左右の任意方向に動かすことができる。また、従輪 3 4 は車輪 3 2 を補助する補助輪である。したがって、ロボット 1 8 は、配置された空間内を自律制御によって移動可能である。

40

【 0 0 5 1 】

台車 3 0 の上には、円柱形のセンサ取り付けパネル 3 8 が設けられ、このセンサ取り付けパネル 3 8 には、多数の赤外線距離センサ 4 0 が取り付けられる。これらの赤外線距離センサ 4 0 は、センサ取り付けパネル 3 8 すなわちロボット 1 8 の周囲の物体（人間や障害物など）との距離を測定するものである。

【 0 0 5 2 】

50

なお、この実施例では、距離センサとして、赤外線距離センサを用いるようにしてあるが、赤外線距離センサに代えて、小型のLRFや、超音波距離センサおよびミリ波レーダなどを用いることもできる。

【0053】

センサ取り付けパネル38の上には、胴体42が直立するように設けられる。また、胴体42の前方中央上部（人の胸に相当する位置）には、上述した赤外線距離センサ40がさらに設けられ、ロボット18の前方の主として人間との距離を計測する。また、胴体42には、その側面側上端部のほぼ中央から伸びる支柱44が設けられ、支柱44の上には、全方位カメラ46が設けられる。全方位カメラ46は、ロボット18の周囲を撮影するものであり、後述する眼カメラ70とは区別される。この全方位カメラ46としては、たとえばCCDやCMOSのような固体撮像素子を用いるカメラを採用することができる。なお、これら赤外線距離センサ40および全方位カメラ46の設置位置は、当該部位に限定されず適宜変更されてもよい。

10

【0054】

胴体42の両側面上端部（人の肩に相当する位置）には、それぞれ、肩関節48Rおよび肩関節48Lによって、上腕50Rおよび上腕50Lが設けられる。図示は省略するが、肩関節48Rおよび肩関節48Lは、それぞれ、直交する3軸の自由度を有する。すなわち、肩関節48Rは、直交する3軸のそれぞれの軸廻りにおいて上腕50Rの角度を制御できる。肩関節48Rの或る軸（ヨー軸）は、上腕50Rの長手方向（または軸）に平行な軸であり、他の2軸（ピッチ軸およびロール軸）は、その軸にそれぞれ異なる方向から直交する軸である。同様にして、肩関節48Lは、直交する3軸のそれぞれの軸廻りにおいて上腕50Lの角度を制御できる。肩関節48Lの或る軸（ヨー軸）は、上腕50Lの長手方向（または軸）に平行な軸であり、他の2軸（ピッチ軸およびロール軸）は、その軸にそれぞれ異なる方向から直交する軸である。

20

【0055】

また、上腕50Rおよび上腕50Lのそれぞれの先端には、肘関節52Rおよび肘関節52Lが設けられる。図示は省略するが、肘関節52Rおよび肘関節52Lは、それぞれ1軸の自由度を有し、この軸（ピッチ軸）の軸廻りにおいて前腕54Rおよび前腕54Lの角度を制御できる。

【0056】

前腕54Rおよび前腕54Lのそれぞれの先端には、人の手に相当する球体56Rおよび球体56Lがそれぞれ固定的に設けられる。ただし、指や掌の機能が必要な場合には、人間の手の形をした「手」を用いることも可能である。また、図示は省略するが、台車30の前面、肩関節48Rと肩関節48Lとを含む肩に相当する部位、上腕50R、上腕50L、前腕54R、前腕54L、球体56Rおよび球体56Lには、それぞれ、接触センサ58（図4で包括的に示す）が設けられる。台車30の前面の接触センサ58は、台車30への人間や他の障害物の接触を検知する。したがって、ロボット18は、その自身の移動中に障害物との接触が有ると、それを検知し、直ちに車輪32の駆動を停止してロボット18の移動を急停止させることができる。また、その他の接触センサ58は、当該各部位に触れたかどうかを検知する。なお、接触センサ58の設置位置は、当該部位に限定されず、適宜な位置（人の胸、腹、脇、背中および腰に相当する位置）に設けられてもよい。

30

40

【0057】

胴体42の中央上部（人の首に相当する位置）には首関節60が設けられ、さらにその上には頭部62が設けられる。図示は省略するが、首関節60は、3軸の自由度を有し、3軸の各軸廻りに角度制御可能である。或る軸（ヨー軸）はロボット18の真上（鉛直上向き）に向かう軸であり、他の2軸（ピッチ軸、ロール軸）は、それぞれ、それと異なる方向で直交する軸である。

【0058】

頭部62には、人の口に相当する位置に、スピーカ64が設けられる。スピーカ64は

50

、ロボット 18 が、その周辺の人間に対して音声ないし音によってコミュニケーションを取るために用いられる。また、人の耳に相当する位置には、マイク 66R およびマイク 66L が設けられる。以下、右のマイク 66R と左のマイク 66L とをまとめてマイク 66 と言うことがある。マイク 66 は、周囲の音、とりわけコミュニケーションを実行する対象である人間の音声を取り込む。さらに、人の目に相当する位置には、眼球部 68R および眼球部 68L が設けられる。眼球部 68R および眼球部 68L は、それぞれ眼カメラ 70R および眼カメラ 70L を含む。以下、右の眼球部 68R と左の眼球部 68L とをまとめて眼球部 68 と言うことがある。また、右の眼カメラ 70R と左の眼カメラ 70L とをまとめて眼カメラ 70 と言うことがある。

【0059】

眼カメラ 70 は、ロボット 18 に接近した人間の顔や他の部分ないし物体などを撮影して、それに対応する映像信号を取り込む。また、眼カメラ 70 は、上述した全方位カメラ 46 と同様のカメラを用いることができる。たとえば、眼カメラ 70 は、眼球部 68 内に固定され、眼球部 68 は、眼球支持部（図示せず）を介して頭部 62 内の所定位置に取り付けられる。図示は省略するが、眼球支持部は、2 軸の自由度を有し、それらの各軸廻りに角度制御可能である。たとえば、この 2 軸の一方は、頭部 62 の上に向かう方向の軸（ヨー軸）であり、他方は、一方の軸に直交しかつ頭部 62 の正面側（顔）が向く方向に直行する方向の軸（ピッチ軸）である。眼球支持部がこの 2 軸の各軸廻りに回転されることによって、眼球部 68 ないし眼カメラ 70 の先端（正面）側が変位され、カメラ軸すなわち視線方向が移動される。なお、上述のスピーカ 64、マイク 66 および眼カメラ 70 の設置位置は、当該部位に限定されず、適宜な位置に設けられてよい。

【0060】

このように、この実施例のロボット 18 は、車輪 32 の独立 2 軸駆動、肩関節 48 の 3 自由度（左右で 6 自由度）、肘関節 52 の 1 自由度（左右で 2 自由度）、首関節 60 の 3 自由度および眼球支持部の 2 自由度（左右で 4 自由度）の合計 17 自由度を有する。

【0061】

図 4 はロボット 18 の電氣的な構成を示すブロック図である。この図 4 を参照して、ロボット 18 は、プロセッサ 80 を含む。プロセッサ 80 は、マイクロコンピュータ或いは CPU とも呼ばれ、バス 82 を介して、メモリ 84、モータ制御ボード 86、センサ入力/出力ボード 88 および音声入力/出力ボード 90 に接続される。プロセッサ 80 は、RTC 80a を含んでおり、作成した状態データに対して時刻データを対応付ける。

【0062】

メモリ 84 は、図示は省略をするが、SSD(Solid State Drive)、ROM および RAM を含む。SSD、ROM には、ロボット 18 の動作を制御するための制御プログラムが予め記憶される。たとえば、各センサの出力（センサ情報）を検知するための検知プログラムや、外部コンピュータ（追跡サーバ 10、遠隔操作装置 14 および計画サーバ 16）との間で必要なデータやコマンドを送受信するための通信プログラムなどが記録される。また、RAM は、ワークメモリやバッファメモリとして用いられる。

【0063】

モータ制御ボード 86 は、たとえば DSP で構成され、各腕や首関節および眼球部などの各軸モータの駆動を制御する。すなわち、モータ制御ボード 86 は、プロセッサ 80 からの制御データを受け、右眼球部 68R の 2 軸のそれぞれの角度を制御する 2 つのモータ（図 4 では、まとめて「右眼球モータ 92」と示す。）の回転角度を制御する。同様に、モータ制御ボード 86 は、プロセッサ 80 からの制御データを受け、左眼球部 68L の 2 軸のそれぞれの角度を制御する 2 つのモータ（図 4 では、まとめて「左眼球モータ 94」と示す。）の回転角度を制御する。

【0064】

また、モータ制御ボード 86 は、プロセッサ 80 からの制御データを受け、肩関節 48R の直交する 3 軸のそれぞれの角度を制御する 3 つのモータと肘関節 52R の角度を制御する 1 つのモータとの計 4 つのモータ（図 4 では、まとめて「右腕モータ 96」と示す。

）の回転角度を制御する。同様に、モータ制御ボード 86 は、プロセッサ 80 からの制御データを受け、肩関節 48 L の直交する 3 軸のそれぞれの角度を制御する 3 つのモータと肘関節 52 L の角度を制御する 1 つのモータとの計 4 つのモータ（図 4 では、まとめて「左腕モータ 98」と示す。）の回転角度を制御する。

【0065】

さらに、モータ制御ボード 86 は、プロセッサ 80 からの制御データを受け、首関節 60 の直交する 3 軸のそれぞれの角度を制御する 3 つのモータ（図 4 では、まとめて「頭部モータ 110」と示す。）の回転角度を制御する。そして、モータ制御ボード 86 は、プロセッサ 80 からの制御データを受け、右車輪 32 a を駆動する右車輪モータ 36 a、左車輪 32 b を駆動する左車輪モータ 36 b の回転角度を、それぞれ個別に制御する。なお、この実施例では、車輪モータ 36 を除くモータは、制御を簡素化するためにステッピングモータ（すなわち、パルスモータ）を用いる。ただし、車輪モータ 36 と同様に直流モータを用いるようにしてもよい。また、ロボット 18 の身体部位を駆動するアクチュエータは、電流を動力源とするモータに限らず適宜変更されてもよい。たとえば、他の実施例では、エアアクチュエータなどが適用されてもよい。

【0066】

センサ入力／出力ボード 88 は、モータ制御ボード 86 と同様に、DSP で構成され、各センサからの信号を取り込んでプロセッサ 80 に与える。すなわち、赤外線距離センサ 40 のそれぞれからの反射時間に関するデータがこのセンサ入力／出力ボード 88 を通じてプロセッサ 80 に入力される。また、全方位カメラ 46 からの映像信号が、必要に応じてセンサ入力／出力ボード 88 で所定の処理を施してからプロセッサ 80 に入力される。眼カメラ 70 からの映像信号も、同様に、プロセッサ 80 に入力される。また、上述した複数の接触センサ 58（図 4 では、まとめて「接触センサ 58」と示す。）からの信号がセンサ入力／出力ボード 88 を介してプロセッサ 80 に与えられる。

【0067】

また、センサ入力／出力ボード 88 には、右車輪 32 a の回転速度（車輪速）を検出する右車輪速センサ 112 a、左車輪 32 b の回転速度を検出する左車輪速センサ 112 b がさらに接続される。そして、右車輪速センサ 112 a および左車輪速センサ 112 b からのそれぞれから、一定時間における右車輪 32 a および左車輪 32 b の回転数（車輪速）に関するデータが、センサ入力／出力ボード 88 を通じて、プロセッサ 80 に入力される。

【0068】

音声入力／出力ボード 90 も、他の入力／出力ボードと同様に DSP で構成され、プロセッサ 80 から与えられる音声合成データに従った音声または声がスピーカ 64 から出力される。また、マイク 66 からの音声入力が、音声入力／出力ボード 90 を介してプロセッサ 80 に与えられる。

【0069】

また、プロセッサ 80 は、バス 82 を介して通信 LAN ボード 114 に接続される。通信 LAN ボード 114 は、たとえば DSP で構成され、プロセッサ 80 から与えられた送信データを無線通信装置 116 に与え、無線通信装置 116 は送信データを、ネットワーク 400 を介して外部コンピュータ（追跡サーバ 10、遠隔操作装置 14、計画サーバ 16）に送信する。また、通信 LAN ボード 114 は、無線通信装置 116 を介してデータを受信し、受信したデータをプロセッサ 80 に与える。たとえば、送信データとしては、全方位カメラ 46 および目カメラ 70 によって撮影された周囲の映像データや、状態データであったりする。

【0070】

ここで、状態データに含まれる位置データは、右車輪速センサ 112 a および左車輪速センサ 112 b が出力する車輪 32 の回転数に基づいて求められる。また、状態データに含まれる角度データは、右車輪速センサ 112 a および左車輪速度センサ 112 b の回転比率に基づいて求められる。さらに、状態データに含まれる速度データは、右車輪速セン

サ 1 1 2 a が出力する車輪速データおよび左車輪速センサ 1 1 2 b が出力する車輪速データから構成される。つまり、ロボット 1 8 は、状態データに含まれる速度データとして、右車輪 3 2 a の車輪速データおよび左車輪 3 2 b の車輪速データを追跡サーバ 1 0 に送信する。そして、追跡サーバ 1 0 では、受信した左右の車輪速データに基づいて、ロボット 1 8 の速度を算出する。

【 0 0 7 1 】

次に L R F 1 2 について詳細に説明する。図 5 を参照して、L R F 1 2 の計測範囲は、半径 R (R = 8 m) の半円形状 (扇形) で示される。つまり、L R F 1 2 は、その正面方向を中心とした場合に、左右 9 0 ° の方向を所定の距離 (R) 以内で計測可能である。

【 0 0 7 2 】

また、使用しているレーザーは、日本工業規格 JIS C 6802 「レーザー製品の安全基準」におけるクラス 1 レーザーであり、人の眼に対して影響を及ぼさない安全なレベルである。また、この実施例では、L R F 1 2 のサンプリングレートを 3 7 Hz とした。これは、移動したり、停止したりする実体の位置を連続して検出するためである。

【 0 0 7 3 】

さらに、先述したように、L R F 1 2 は、様々な場所に配置される。具体的には、L R F 1 2 a - 1 2 f の各々は、計測領域が重なるように配置され、図示は省略するが、床面から約 9 0 cm の高さに固定される。この高さは、人間の胴体と腕 (両腕) とを検出可能とするためであり、たとえば、日本人の成人の平均身長から算出される。したがって、遠隔操作装置 1 4 を設ける場所 (地域ないし国) や人間の年齢ないし年代 (たとえば、子供、大人) に応じて、L R F 1 2 を固定する高さを適宜変更するようにしてよい。なお、本実施例では、設定される L R F 1 2 は 6 台としたが、2 台以上であれば、任意の台数の L R F 1 2 を設置してもよい。

【 0 0 7 4 】

このような構成の追跡サーバ 1 0 では、プロセッサ 2 0 が L R F 1 2 からの出力 (距離データ) に基づいて、パーティクルフィルタを用いて、人間およびロボット 1 8 の現在位置の変化を推定する。なお、本実施例では、人間とロボット 1 8 とでは速度や形状が異なるため、人形状のモデル (人形状モデル) またはロボット形状のモデル (ロボット形状モデル) を採用して、パーティクルフィルタの尤度の計算量を軽減している。

【 0 0 7 5 】

たとえば、L R F 1 2 によってスキャンされると、人間などの実体が存在しない可視領域 (o p e n) 、実体が存在する陰領域 (s h a d o w) および実体のエッジ (e d g e) が検出される。

【 0 0 7 6 】

ここで、空間に均等にばら撒くパーティクルを識別する変数を m とし、パーティクルの状態 (位置と速度) を X とし、或る時刻 t の状態 X t のときに観測される観測値ベクトルを Z t としたとき、各 R L F 1 2 の尤度 p i (Z t | X t [m]) は数 1 に従って算出される。ただし、[m] はパーティクルを個別に識別するための添え字である。また、表現の都合上、数式以外では [m] を上付きにしないで記載する。以下、同様である。

【 0 0 7 7 】

[数 1]

$$p_i(Z_t | X_t^{[m]}) = \begin{cases} p_{shadow} + p_{edge}(Z_t | X_t^{[m]}) & | \text{shadow} \\ p_{open} & | \text{open} \end{cases}$$

【 0 0 7 8 】

つまり、尤度 p i (Z t | X t [m]) は、可視領域では一定値 (p o p e n) とし、陰領域では一定値 (p s h a d o w) とエッジの尤度 p e d g e (Z t | X t [m]) との和となる。

【 0 0 7 9 】

さらに、L R F 1 2 の数を $n_{s e n s o r s}$ としたとき、数 1 で求めた各 L R F 1 2 の尤度を統合した統合尤度 $p(Z_t | X_t[m])$ は、数 2 に従って算出される。なお、陰領域に 2 人の人間が存在することがあるので、2 つのパーティクルが同じ実体を追跡しないようにするために、各 R L F 1 2 の尤度 $p_i(Z_t | X_t[m])$ から補正值 $P_{c o l l o c a t i o n}$ が減算される。

【 0 0 8 0 】

[数 2]

$$p(Z_t | X_t^{[m]}) = \sum_{i=1}^{n_{sensors}} p_i(Z_t | X_t^{[m]}) - P_{collocation}$$

10

【 0 0 8 1 】

このようにして求められた統合尤度 $p(Z_t | X_t[m])$ に基づいて、各パーティクルが更新され、人間またはロボット 1 8 の現在位置の変化が推定される。そして、推定された現在位置の変化に基づいて、ロボット 1 8 または人間の位置データを求め、その位置データがバッファに記憶される。また、本実施例では、バッファに記憶された全ての位置データをまとめて「位置履歴データ」と言う。

【 0 0 8 2 】

なお、パーティクルフィルタを利用した人物追跡については、特開 2 0 0 8 - 6 1 0 5 号公報に詳細が開示されている。

【 0 0 8 3 】

20

図 6 は、L R F 1 2 a - 1 2 f が設置された或る環境の地図を示す図解図である。図 6 を参照して、地図が表す場所は或るショッピングモールである。L R F 1 2 a , 1 2 c , 1 2 d の 3 台は地図の上側の位置に対応して設置されており、L R F 1 2 b , 1 2 e , 1 2 f の 3 台は地図の下側の位置に対応して設定されている。そして、2 台以上の L R F 1 2 の計測領域が重なる領域は、検出領域 F として示され、図 6 では斜線の領域である。なお、検出領域 F における位置は、図 6 における左下を原点とする平面座標系で示される。

【 0 0 8 4 】

検出領域 F (空間) 内では、ロボット 1 8 a、ロボット 1 8 b、ロボット 1 8 c および人間は、実体 E a、実体 E b、実体 E c および実体 E d (区別しない場合は「実体 E 」と言う。) として検出される。また、図 6 では、各実体 E に対応する位置履歴データに基づいて、移動軌跡 M を示す。つまり、実体 E a の位置履歴データは移動軌跡 M a で示され、実体 E b の位置履歴データは移動軌跡 M b で示され、実体 E c の位置履歴データは移動軌跡 M c で示され、実体 E d の位置履歴データは移動軌跡 M d で示される。

30

【 0 0 8 5 】

図 7 は追跡サーバ 1 0 によって検出された複数の位置データ、つまり位置履歴データを示す図解図である。図 7 を参照して、各位置データは、位置を検出した時刻 (T) および検出領域 F の平面座標を示す数値 (X , Y) から構成される。そして、各位置履歴データは、実体 E 毎に分けて記憶される。たとえば、追跡サーバ 1 0 のメモリ 2 2 のバッファには、実体 E a の位置履歴データ、実体 E b の位置履歴データ、実体 E c の位置履歴データおよび実体 E d の位置履歴データが記憶される。

40

【 0 0 8 6 】

図 8 は追跡サーバ 1 0 が受信した複数の状態データ、つまり状態履歴データを示す図解図である。図 8 を参照して、各状態データは、受信した時刻 (t)、検出領域 F における平面座標を示す数値 (x , y)、ロボット 1 8 の向きを示す角度 (d)、ロボット 1 8 の右車輪 3 2 a の速度 (r v) およびロボット 1 8 の左車輪 3 2 b の速度 (l v) から構成される。そして、位置履歴データと同様に、各状態履歴データは、ロボット 1 8 毎に分けて記憶される。たとえば、メモリ 2 2 のバッファには、ロボット 1 8 a の状態履歴データ、ロボット 1 8 b の状態履歴データおよびロボット 1 8 c の状態履歴データが記憶される。

【 0 0 8 7 】

50

なお、位置データおよび状態データは、一定時間（たとえば、360秒）が経つと、古いデータから削除（上書き）される。つまり、位置履歴データおよび状態履歴データがメモリ22のメモリ容量を圧迫することは無い。

【0088】

また、上記した平面座標を示す数値（ X ， Y ）および（ x ， y ）において、1座標の変化は1cmの距離変化に対応する。たとえば、実体Eの位置データが（ X ， Y ）である場合に、実体Eが右に1cm移動すると、実体Eの座標が（ $X+1$ ， Y ）に変化する。

【0089】

図9は、各ロボット18のRTMから構成されたRTMデータの図解図である。この実施例では、追跡サーバ10が状態データを受信するために各ロボット18との通信を確立すると、ロボット18毎にRTMを作成する。そして、各ロボット18に対して送信される補正データがこのRTMに基づいて決められる。そのため、各RTMは現在のロボット18の位置、角度および速度の値を含む。

【0090】

図9を参照して、RTMデータは、各ロボット18に割り振られたロボットIDに対応づけて記憶される。たとえば、ロボットIDが「001」であるロボット18aのRTMは、（ RX_1 ， RY_1 ）が位置を示し、 RD_1 が角度を示し、 RV_1 が速度を示す。

【0091】

ここで、追跡サーバ10がロボット18から状態データを受信すると、そのロボット18に対応するRTMの位置、角度および速度が更新される。たとえば、ロボット18aから受信した状態データにおいて、位置が（ x_1 ， y_1 ）、角度が d_1 、速度が rv_1 ， lv_1 であれば、ロボットID「001」のRTMにおいて、（ RX_1 ， RY_1 ）， RD_1 ， RV_1 が、受信した状態データの数値に更新される。なお、速度 RV_1 は、 rv_1 および lv_1 に基づいて算出された速度に更新される。

【0092】

ところが、ロボット18の車輪32は、ロボット18が停止したり、曲がったりする度に、地面に対して僅かに滑るため、車輪32の回転から求める位置および角度と、実際の位置および角度とに誤差が生じる。そして、ロボット18が移動するにつれて、その誤差は増大するため、ロボット18が移動するにつれて、状態データに含まれる位置データおよび角度データの信頼性は低くなる。そのため、ロボット18が出力する状態データだけに基づいて、そのロボット18に対するRTMを更新すると、RTMが正しいロボット18の状態（位置、角度および速度）を示せなくなってしまう。

【0093】

そこで、本実施例では、検出領域F内で検出された実体Eと検出領域F内を移動するロボット18とを関連付け、LRF12を利用して検出された実体Eの位置データおよび関連付けられたロボット18の状態データに基づいてRTMを更新する。これにより、各RTMは、ロボット18の正しい状態を示すことができるようになる。

【0094】

まず、ロボット18と実体Eとの関連付けについて説明する。本願発明では、実体Eおよびロボット18の1秒毎の移動距離、つまり速度に着目する。なお、各実体Eおよびロボット18aの速度は、三平方の定理に基づいて、或る時刻の座標と1秒後の座標とから距離（mm）を求めることで、速度（mm/s）を求めることができる。

【0095】

図10の運動履歴グラフは、縦軸に距離（mm）をとり、横軸に時間（秒：s）をとるグラフである。そして、運動履歴グラフは、実体Ea - 実体Edの15秒間の速度の変化と、ロボット18aの速度の変化とを示す。この運動履歴グラフにおいて、実体Ea、実体Eb、実体Ecおよび実体Edの速度に対応するグラフは、実線La、実線Lb、実線Lcおよび実線Ldで示される。また、ロボット18aの速度に対応するグラフは点線Dlで示される。

【0096】

10

20

30

40

50

たとえば、運動履歴グラフからは、点線 D L（ロボット 18 a の速度）の変化は、実線 L a（実体 E a の速度）、実線 L c（実体 E c の速度）および実線 L d（実体 E d の速度）の変化とは異なっているが、実線 L b（実体 E b の速度）の変化とは類似していることが分かる。

【0097】

そこで、ロボット 18 a の速度と各実体 E の速度との R M S（Root Mean Square：二乗平均）誤差を、以下の数 3 に示す式に基づいて求めると、図 11 に示す R M S 誤差グラフを求めることができる。なお、本実施例における R M S 誤差は、R M S 速度差と言う場合もある。

【0098】

[数 3]

$$\text{RMS 誤差} = \sqrt{\frac{\sum_{i=1}^{RT} (\text{ロボットの移動速度}_i - \text{実体の移動速度}_i)^2}{RT}}$$

(RTは所定の時間)

【0099】

図 11 を参照して、R M S 誤差グラフは、縦軸が R M S 誤差、横軸が実体 I D のグラフであり、各 R M S 誤差は数 3 の R T を 15（秒）として算出されている。まず、R M S 誤差グラフにおいて、実体 E a、実体 E c および実体 E d の R M S 誤差に着目すると、実体 I D が「001」の実体 E a とロボット 18 a との R M S 誤差は 330 mm である。また、実体 I D が「003」の実体 E c とロボット 18 a との R M S 誤差は 500 mm であり、実体 I D が「004」の実体 E d とロボット 18 a との R M S 誤差は 600 mm である。これに対して、実体 I D が「002」の実体 E b とロボット 18 a との R M S 誤差は 30 mm である。つまり、速度の変化が類似するロボット 18 a と実体 E b との R M S 誤差は、他の実体 E a、E c、E d との R M S 誤差に比べて小さいことが分かる。

【0100】

そのため、本実施例では、或るロボット 18 に対して、R M S 誤差、つまり R M S 速度差（第 1 速度差）が 350 mm/s（第 1 所定速度）以下であり、かつ現在の速度差が 350 mm/s 以下の実体 E が 1 つだけであれば、或るロボット 18 とその実体 E とを関連付ける。

【0101】

たとえば、図 11 を参照して、ロボット 18 a において、R M S 速度差が 350 mm/s 以下の実体 E は、実体 E a および実体 E b である。さらに、図 10 を参照して、15 秒の時点を経現在の速度として考えると、現在の速度差が 350 mm/s 以下の実体 E は、実体 E b だけである。そのため、このような場合にはロボット 18 a と実体 E b とが関連付けられる。そして、プロセッサ 20 は、各ロボット 18 に対して上述した判断を行うことで、他のロボット 18 と他の実体 E とを関連付ける。

【0102】

また、プロセッサ 20 が上述した判断を実行する際には、図 12 に示す追跡実体リストを用いる。図 12 を参照して、追跡実体リストは、実体 I D およびフラグの列から構成されている。そして、フラグの列には、先述した条件を満たしていれば「1」が設定され、満たしていなければ「0」が設定される。

【0103】

たとえば、プロセッサ 20 は、ロボット 18 a に対して、実体 E 毎に条件を満たすか否かを判断し、その判断結果をフラグの列に記録する。そして、全ての実体 E に対して判断が終了すれば、プロセッサ 20 は、追跡実体リストのフラグの列において「1」が設定された実体 I D が 1 つだけであるか否かを判断する。このとき、フラグが「1」に設定された実体 I D が「002」、つまり実体 E b だけであれば、プロセッサ 20 はロボット 18 a と実体 E b とを関連付ける。

【0104】

10

20

30

40

50

また、プロセッサ 20 は、関連付けた結果を図 13 に示す関連付テーブルに記録する。関連付けテーブルは、ロボット ID を示す列と実体 ID を示す列とから構成される。たとえば、プロセッサ 20 は、関連付テーブルにおいて、ロボット 18 a のロボット ID 「001」に対して、実体 E b の実体 ID 「002」を設定する。

【0105】

本実施例では、このような処理を経てロボット 18 と実体 E とを関連付ける処理を、「グローバル関連付処理」と言う。なお、上記説明では、簡単のために RT を 15 として RSM 速度差を算出していたが、実際のグローバル関連付処理では、RT が 300 に設定される。

【0106】

次に、RTM の更新について説明する。RTM の位置は、関連付けられた実体の位置データに基づいて更新される。たとえば、ロボット 18 a に関連付けられた実体 E b の最新の位置データが、(30, 40) であれば、ロボット 18 a に対応する RTM の位置は (30, 40) に更新される。

【0107】

また、RTM の角度は、実体 E の位置データおよびロボット 18 の状態データから、カルマンフィルタを利用して次の位置で推定される角度（推定角度）および推定角度誤差量を求めることで、更新できる。具体的には、図 14 を参照して、受信したロボット 18 の状態データから次の位置を予測し、予測された位置（予測位置）と前回位置とから予測角度を計算する。また、追跡サーバ 10 から現在の位置を取得し、前回位置と現在位置とから検出角度を計算する。そして、検出角度および予測角度から、観測角度 Z_k を算出（観測）する。ただし、「 Z_k 」が下付き添え字である場合については、表現の都合上、数式以外では「 k 」を「 Z 」の添え字とせず、まとめて下付きの添え字にして記載する。以下、同様である。

【0108】

さらに、追跡サーバ 10 の観測には常に位置誤差量 σ^2_p が含まれている。そのため、観測角度誤差量 $\sigma^2_{Z_k}$ は、数 4 に従って近似することができる。

【0109】

[数 4]

$$\sigma^2_{Z_k} \approx \frac{\sigma^2_{po}}{d_s}$$

【0110】

ここで、数 4 において、変位量 d_s は位置履歴データにおける前回位置から現在位置までの距離であり、角度誤差量 $\sigma^2_{Z_k}$ はその変位量 d_s が大きい場合に小さくなる。つまり、実体 E の移動距離が大きいほど、角度誤差量 $\sigma^2_{Z_k}$ が小さくなる。

【0111】

また、前回の推定角度誤差量 $\sigma^2_{\theta-k}$ および角度誤差量 $\sigma^2_{Z_k}$ から、カルマンゲイン K_k を数 5 に従って算出する。そして、そのカルマンゲイン K_k 、観測角度 Z_k および前回の推定角度 $\theta-k$ を、数 6 に従う式に代入することで、推定角度 θ^k を算出できる。

【0112】

ただし、「 $\theta-k$ 」が下付きの添え字である場合については、表現の都合上、数式以外では「 $-$ 」を「 θ 」の添え字とせず、まとめて下付きの添え字にして記載する。さらに、添え字「 $\wedge$ 」は、数式では「 θ 」の上に付されるが、表現の都合上、数式以外では「 θ 」の上付き添え字として記載する。以下、同様である。

【0113】

[数 5]

10

20

30

40

50

$$K_k = \frac{\sigma_{z_k}^2}{\sigma_{z_k}^2 + \sigma_{\theta_k^-}^2}$$

【 0 1 1 4 】

[数 6]

$$\hat{\theta}_k = \theta_k^- + K_k(Z_k - \theta_k^-)$$

【 0 1 1 5 】

さらに、カルマンゲイン K_k 、前回の推定角度 θ_k^- および時間毎のプロセスノイズ σ_{pr}^2 を数 7 に従う式に代入することで、推定角度誤差量 $\sigma_{\theta_k}^2$ を算出することができる。なお、本実施例ではノイズ σ_{pr}^2 を 0.0001 ラジアンとして計算する。

10

【 0 1 1 6 】

[数 7]

$$\sigma_{\theta_k}^2 = \sigma_{\theta_k^-}^2 + K_k(\sigma_{\theta_k}^2) + \sigma_{pr}^2$$

【 0 1 1 7 】

このようにして、推定角度 $\hat{\theta}_k$ が算出されると、現在の R T M の角度との角度差を算出して、R T M の角度を推定角度に補正し、算出した角度差を補正角度データとして記憶する。

20

【 0 1 1 8 】

なお、前回値が存在しない場合、たとえば 1 回目の角度補正処理では、前回の推定角度 θ_k^- および推定角度誤差量 $\sigma_{\theta_k}^2$ の代わりに所定値を、数 5 - 7 の数式に代入して推定角度 $\hat{\theta}_k$ および推定角度誤差量 $\sigma_{\theta_k}^2$ を算出する。また、所定値は、「0」であってもよい、ロボット 18 の初期状態に応じた値であってもよい。

【 0 1 1 9 】

そして、本実施例では、ロボット 18 から状態データを受信（取得）したときに、追跡サーバ 10 はロボット 18 に対して記憶した補正角度データと R T M が示す位置データを、補正データとしてロボット 18 に送信する。そして、ロボット 18 は、受信した位置データおよび補正角度データに従って、自身の位置および角度を補正する。これにより、追跡サーバ 10 は、ロボット 18 を、検出領域 F 内において自己位置同定することができる。

30

【 0 1 2 0 】

このように、ロボット自己位置同定システム 100 は、位置だけでなく、ロボット 18 の角度も補正することで、自己位置同定の精度を高めることができる。また、本実施例では、カルマンフィルタを用いて、推定角度 $\hat{\theta}_k$ を求めることができる。

【 0 1 2 1 】

なお、追跡サーバ 10 は、算出した推定角度誤差量 $\sigma_{\theta_k}^2$ を補正精度として記憶して置き、この補正精度が低い場合、つまり推定角度誤差量 $\sigma_{\theta_k}^2$ が大きい場合には、角度の補正データをロボット 18 に送信しない。つまり、ロボット 18 には、位置を補正するための補正データのみが送信される。このように、推定角度誤差量 $\sigma_{\theta_k}^2$ を補正精度として利用することで、角度の補正が可能であるかを判断できるようになる。

40

【 0 1 2 2 】

また、本実施例では、ロボット 18 と実体 E とを関連付けた後に、ローカル追跡エラーまたはグローバル追跡エラーが生じることが有る。ローカル追跡エラーは、追跡サーバ 10 に問題が起きたり、実体 E を追跡するパーティクルフィルタが少しの間、消えてしまったりしたときに生じる。また、グローバル追跡エラーは、特定の環境下で、ロボット 18 と実体 E とが誤って関連付けられたときに生じる。

【 0 1 2 3 】

50

そこで、本実施例では、ローカル追跡エラーおよびグローバル追跡エラーを除去するために、次の3つの条件のうち1つでも満たされる場合に、1組のロボット18と実体Eとの関連付けを解除する。すなわち、3つの条件は、(1) 関連付けられた実体Eがすでに追跡されていない、(2) ロボット18と関連付けられた実体Eとが離れている(たとえば、2m)、(3) ロボット18の速度と関連付けられた実体Eの速度が異なる(たとえば、0.35mm/s)である。

【0124】

つまり、ロボット自己位置同定システム100は、エラーの要因となるロボット18と実体Eとの関連付けを解除できる。

【0125】

そして、ロボット18と実体との関連付けが解除されて、所定時間(たとえば、5秒)以内であれば、追跡サーバ10は「ローカル関連付処理」を実行して、関連付けが解除されたロボット18を、他の実体Eと関連付ける。

【0126】

ローカル関連付処理は、RMS速度差(第2速度差)が350mm/s(第2所定速度)以下であり、かつ現在の速度差が350mm/s以下であり、かつロボット18と実体Eとの距離が1m(所定距離)以下の実体Eが複数存在していれば、ロボット18と最も近い実体Eが、ロボット18と関連付けられる。

【0127】

また、ローカル関連付処理は、数3に示すRTを5(秒)として計算するため、先述したグローバル関連付処理に比べて処理の速度が速い。以下、関連付けの解除およびローカル関連付処理について、具体例を示して説明する。

【0128】

図15(A)は、人間とロボット18との位置関係を示す図解図である。また、図15(A)に示す状態では、グローバル関連付処理が実行されていない。検出領域F内において、ロボット18は人間に対してコミュニケーション行動を行っている。このとき、追跡サーバ10は、(200, 300)の位置でロボット18を検出し、(250, 300)の位置で人間を検出する。そして、ロボット18は、位置データ(240, 300)および速度データ(0mm/s)を含む、状態データを出力する。

【0129】

この状態でグローバル関連付処理が実行されると、図15(B)に示すように、人間に対応する実体Eとロボット18とが誤って関連付けられる。つまり、コミュニケーション行動が行われていれば、ロボット18と人間とは動かないため、停止している人間またはロボット18が他に存在しなければ、図15(B)に示すように、人間に対応する実体Eとロボット18とが誤って関連付けられる。そして、ロボット18におけるRTMの位置は、人間の位置(250, 300)に補正される。

【0130】

ところが、コミュニケーション行動が終了して、人間がロボット18の前から立ち去ると、図15(C)に示すような状態になる。つまり、ロボット18が出力する状態データは位置データが(250, 300)となるが、人間に対応する実体Eの位置データが(500, 300)となり、RTMの位置と人間の位置とで、距離が250cm(=500-250)離れていることとなる。このとき、上記した関連付け解除の条件(2)を満たすことになるため、人間に対応する実体Eとロボット18との関連付けは解除される。

【0131】

そして、関連付けが解除され、所定時間以内であれば、ローカル関連付処理が実行される。たとえば、図15(C)に示す状態では、ロボット18に対応する実体Eがローカル関連付けの条件(1)-(3)を満たすため、その実体Eとロボット18とが正しく関連付けられる。このように、関連付けが解除されて間もなければ、ロボット18と実体Eとを容易に関連付け直すことができる。

【0132】

10

20

30

40

50

また、ローカル関連付処理を実行しても、5秒以内にロボット18と実体Eとを関連付けられなければ、グローバル関連付処理が実行される。つまり、関連付けが解除されて所定時間が経過したとしても、ロボット18と実体Eとを関連付け直すことができる。

【0133】

図16は、図2に示すメモリ22のメモリマップ300の一例を示す図解図である。図16に示すように、メモリ22はプログラム記憶領域302およびデータ記憶領域304を含む。プログラム記憶領域302には、追跡サーバ10を動作させるためのプログラムとして、通信プログラム312および補正プログラム314などが記憶される。なお、通信プログラム312および補正プログラム314などの処理によって、ローカライゼーション機能が有効になる。

10

【0134】

通信プログラム312は、ロボット18から状態データを受信し、補正データをロボット18に送信するためのプログラムである。補正プログラム314は、RTMデータを補正するためのプログラムであり、追跡プログラム314a、関連付解除プログラム314b、関連付プログラム314cおよび角度補正プログラム314dのサブルーチンを含む。

【0135】

追跡プログラム314aは、パーティクルフィルタの尤度を更新し、実体Eの位置を検出するためのプログラムである。関連付解除プログラム314bは、ロボット18と実体Eとの関連付けを解除するためのプログラムである。関連付プログラム314cは、ロボット18と実体Eとを関連付けるためのプログラムであり、さらにローカル関連付プログラム316およびグローバル関連付プログラム318のサブルーチンを含む。そして、角度補正プログラム314dは、上述したカルマンフィルタによって角度を推定し、角度の補正データを求めるためのプログラムである。

20

【0136】

なお、図示は省略するが、追跡サーバ10を動作させるためのプログラムとしては、関連付けなどが解除されたときに時間をカウントするプログラム、追跡サーバ10とロボット18との時刻を同期するためのプログラムなどを含む。

【0137】

また、図17を参照して、データ記憶領域304には、LRFバッファ330、位置履歴バッファ332、状態履歴バッファ334、追跡実体リストバッファ336、補正角度バッファ338、補正精度バッファ340および前回値バッファ342が設けられる。また、データ記憶領域340には、関連付テーブルデータ344およびRTMデータ346が記憶されるとともに、初回フラグ348、初期起動カウンタ350および解除カウンタ352も設けられる。

30

【0138】

LRFバッファ330は、各LRF12によって計測された距離データが一時記憶されるバッファである。位置履歴バッファ332は、図7に示す位置履歴データが一時的に記憶されるバッファである。状態履歴バッファ334は、図8に示す状態履歴データが一時的に記憶されるバッファである。追跡実体リストバッファ336は、図12に示す追跡実体リストのデータが一時的に記憶されるバッファである。補正角度バッファ338は、推定角度 $\theta_k^{\wedge}$ とRTMの角度との角度差が、補正角度データとして一時的に記憶されるバッファである。補正精度バッファ340は、推定角度誤差量 $\sigma^2_{\theta_k^{\wedge}}$ を補正精度データとして一時的に記憶するためのバッファである。前回値バッファ342は、角度補正処理で算出された推定角度 $\theta_k^{\wedge}$ および推定角度誤差量 $\sigma^2_{\theta_k^{\wedge}}$ を、前回の推定角度 $\theta_k^{\wedge}$ および推定角度誤差量 $\sigma^2_{\theta_k^{\wedge}}$ として一時的に記憶するためのバッファである。

40

【0139】

関連付けテーブルデータ344は、図13に示す関連付テーブルのデータである。RTMデータ346は、図9に示すように、各ロボット18に対応する複数のRTMから構成されるデータである。

50

【 0 1 4 0 】

初回フラグ 3 4 8 は、追跡サーバ 1 0 が実行されてから一定時間が経過したか否かを判断するためのフラグである。たとえば初回フラグ 3 4 8 は 1 ビットのレジスタで構成される。初回フラグ 3 4 8 がオン（成立）されると、レジスタにはデータ値「 1 」が設定される。一方、初回フラグ 3 4 8 がオフ（不成立）されると、レジスタにはデータ値「 0 」が設定される。また、初回フラグ 3 4 8 は、追跡サーバ 1 0 が起動されるとオンになり、追跡サーバ 1 0 が起動してから一定時間が経過するとオフになる。

【 0 1 4 1 】

初期起動カウンタ 3 5 0 は、追跡サーバ 1 0 が起動してから一定時間が経過するまで時間をカウントするためのカウンタである。解除カウンタ 3 5 2 は、ロボット 1 8 と実体と E との関連付けが解除されてからの時間をカウントするためのカウンタである。

10

【 0 1 4 2 】

なお、図示は省略するが、データ記憶領域 3 0 4 には、遠隔操作装置 1 4 および計画サーバ 1 6 とのデータ通信の結果を一時的に記憶するバッファや、カルマンフィルタの計算結果が一時的に記憶されるバッファなどが設けられると共に、追跡サーバ 1 0 の動作に必要な他のカウンタやフラグなども設けられる。

【 0 1 4 3 】

以下、追跡サーバ 1 0 によって実行される本願発明のフロー図について説明する。また、図 1 8 および図 1 9 のフロー図は、通信プログラム 3 1 2 による処理を示し、図 2 0 のフロー図は補正プログラム 3 1 4 による処理を示す。また、図 2 1 および図 2 2 のフロー図は追跡プログラム 3 1 4 a による処理を示し、図 2 3 のフロー図は関連付解除プログラム 3 1 4 b による処理を示す。さらに、図 2 4 のフロー図は関連付プログラム 3 1 4 c による処理を示し、図 2 5 のフロー図はローカル関連付プログラム 3 1 6 による処理を示し、図 2 6 のフロー図はグローバル関連付プログラム 3 1 8 による処理を示し、図 2 7 のフロー図は角度補正プログラム 3 1 4 d による処理を示す。

20

【 0 1 4 4 】

図 1 8 を参照して、たとえば、追跡サーバ 1 0 の電源がオンにされると、通信処理が行われる。そして、プロセッサ 2 0 は、ステップ S 1 では各ロボット 1 8 との通信を確立する。たとえば、検出領域 F 内にロボット 1 8 a - 1 8 c が存在していれば、それらのロボット 1 8 との通信を確立する。続いて、ステップ S 3 では、通信を確立しているロボット 1 8 の数を変数 R n に設定する。たとえば、3 台のロボット 1 8 (1 8 a , 1 8 b , 1 8 c) との通信が確立されていれば、変数 R n には「 3 」が設定される。なお、変数 R n は、通信が確立されているロボット 1 8 の総数を示す変数として定義されている。続いて、ステップ S 5 では、R n 個の R T M を作成する。たとえば、変数 R n に「 3 」が設定されていれば、3 つの R T M を作成する。つまり、R T M データ 3 4 6 には、作成された 3 つの R T M が含まれる。

30

【 0 1 4 5 】

続いて、図 1 9 を参照して、ステップ S 7 では、ロボット 1 8 が出力した状態データを取得する。たとえば、ロボット 1 8 a が状態データを発信していれば、その状態データを受信する。続いて、ステップ S 9 では、ロボット I D に基づいて状態データをバッファに記憶する。たとえば、ロボット I D が「 0 0 1 」である、ロボット 1 8 a の状態データを取得した場合、その状態データを状態履歴バッファ 3 3 4 に記憶する。また、ロボット 1 8 a の状態データは、ロボット 1 8 a の状態履歴データ（図 8 参照）に追加される。なお、ステップ S 9 の処理を実行するプロセッサ 2 0 は、状態データ記憶手段として機能する。

40

【 0 1 4 6 】

続いて、ステップ S 1 1 では、補正できるか否かを判断する。つまり、ロボット 1 8 と実体 E とが関連付けられているか否かを判断する。また、本実施例では、追跡サーバ 1 0 は起動してから一定時間が経過しなければ、ロボット 1 8 と実体 E とをまだ関連付けることができない。そのため、ステップ S 1 1 では、起動してから一定時間が経過したことを

50

示す初回フラグ 3 4 8 がオフであるか否かを判断する。ステップ S 1 1 で “ N O ” であれば、つまり追跡サーバ 1 0 が起動してから一定時間が経過していなければ、ステップ S 1 7 に進む。

【 0 1 4 7 】

また、ステップ S 1 1 で “ Y E S ” であれば、つまり追跡サーバ 1 0 が起動してから一定時間が経過し、ロボット 1 8 と実体 E とが関連付けられていれば、ステップ S 1 3 で補正精度が高いか否かを判断する。つまり、補正精度バッファ 3 4 0 に記憶された推定角度誤差量 $\sigma^2_{\theta \wedge k}$ が閾値（たとえば、2 度）以下であるか否かを判断する。ここで、ステップ S 1 3 で “ N O ” であれば、つまり補正精度バッファ 3 4 0 に一時記憶された推定角度誤差量 $\sigma^2_{\theta \wedge k}$ が閾値以下でなければ、ステップ S 1 5 で R T M の位置データをロボット I D に基づいて送信する。たとえば、ロボット I D 「 0 0 1 」 のロボット 1 8 a から状態データを取得していれば、同じロボット I D のロボット 1 8 a に対して、R T M の位置データを送信する。続いて、ステップ S 1 7 では、ロボット I D に基づいて R T M を更新する。つまり、取得された状態データに含まれる角度データに基づいて、R T M の角度および速度を更新する。たとえば、ロボット 1 8 a の状態データにおいて、角度データが 3 0 度であれば、ロボット 1 8 a の R T M の角度が 3 0 度に更新される。

10

【 0 1 4 8 】

また、ステップ S 1 3 で “ Y E S ” であれば、つまり補正精度バッファ 3 4 0 に一時記憶された推定角度誤差量 $\sigma^2_{\theta \wedge k}$ が閾値以下であれば、ステップ S 1 9 で R T M の位置データおよび補正角度データをロボット I D に基づいて送信する。たとえば、状態データを受信したロボット 1 8 のロボット I D に基づいて、R T M の位置データおよび補正角度データが送信される。なお、ロボット 1 8 に送信される補正角度データは、補正角度バッファ 3 3 8 から読み出される。

20

【 0 1 4 9 】

続いて、ステップ S 2 1 では、取得した角度および補正角度データから現在の角度を計算する。たとえば、状態データの角度データが 2 5 度であり、補正角度データが - 2 度であれば、現在の角度が 2 3 度（= 2 5 - 2 ）になる。そして、ステップ S 1 7 では、R T M の角度はステップ S 2 1 で計算された角度に更新される。たとえば、R T M の角度が 3 0 度であれば、ステップ S 2 1 で算出された 2 3 度に更新される。

【 0 1 5 0 】

30

なお、ステップ S 1 5 またはステップ S 1 9 の処理を実行するプロセッサ 2 0 は、送信手段として機能する。また、特にステップ S 1 5 の処理を実行するプロセッサ 2 0 は第 2 送信手段として機能し、ステップ S 1 9 の処理を実行するプロセッサ 2 0 は第 1 送信手段として機能する。

【 0 1 5 1 】

ここで、ステップ S 1 1 で “ N O ” と判断された場合には、ステップ S 1 7 では、状態データに含まれる位置データ、角度データおよび速度データに基づいて、R T M の位置、角度および速度が更新される。つまり、ロボット 1 8 と実体 E とが関連付けられていない状態では、ロボット 1 8 が出力する状態データに基づいて R T M が更新される。

【 0 1 5 2 】

40

ステップ S 2 3 では、各ロボット 1 8 との通信が継続しているか否かを判断する。たとえば、ロボット 1 8 a - 1 8 d との通信が継続しているか否かを判断する。ステップ S 2 3 で “ N O ” であれば、つまりロボット 1 8 との通信が途切れていれば、ステップ S 2 5 で通信が途切れたロボット I D に基づいて R T M を削除する。たとえば、ロボット I D が 「 0 0 4 」 のロボット 1 8 d との通信が途切れていれば、R T M データ 3 4 6 を構成する、ロボット I D 「 0 0 4 」 の R T M が削除される。そして、ステップ S 2 5 の処理が終了すれば、ステップ S 3 に戻る。つまり、変数 R n をディクリメントして、R T M データ 3 4 6 に含まれる R T M の数と変数 R n の数値とを一致させる。

【 0 1 5 3 】

また、ステップ S 2 3 で “ Y E S ” であれば、つまり各ロボット 1 8 との通信が継続し

50

ていれば、ステップS 2 7で新しいロボット1 8との通信が確立されたか否かを判断する。たとえば、5 台目のロボット1 8が検出領域Fに入ったか否かを判断する。ステップS 2 7で“ N O ”であれば、つまり新たなロボット1 8との通信が確立されなければ、ステップS 7に戻る。一方、ステップS 2 7で“ Y E S ”であれば、たとえば5 台目のロボット1 8との通信が確立されれば、ステップS 3に戻る。つまり、変数R nを更新して、R T Mデータ3 4 6に新しいR T Mを追加する。

【 0 1 5 4 】

図2 0には補正プログラム3 1 4の処理を示すフロー図が示される。たとえば、追跡サーバ1 0の電源がオンにされると、プロセッサ2 0は、ステップS 4 1で追跡処理を実行する。この追跡処理については、図2 1および図2 2に示すフロー図を用いて後述するため、ここでの詳細な説明は省略する。

10

【 0 1 5 5 】

続いて、ステップS 4 3では、1 度目の処理か否かを判断する。つまり追跡サーバ1 0が起動してから、補正処理が実行されるのが1 度目であるか否かを判断する。また、具体的には、初回フラグ3 4 8がオンであるか否かを判断する。ステップS 4 3で“ N O ”であれば、つまり初回フラグ3 4 8がオフであれば、ステップS 4 7に進む。一方、ステップS 4 3で“ Y E S ”であれば、つまり初回フラグ3 4 8がオンであれば、ステップS 4 5で一定時間が経過したか否かを判断する。たとえば、一定時間は3 0 0 秒であり、ステップS 4 5では追跡サーバ1 0が起動してから3 0 0 秒経過したか否かを判断する。また、具体的には、追跡サーバ1 0が起動してからの時間をカウントする初期起動カウンタ3 5 0の値が3 0 0 秒を示す値を超えているか否かを判断する。

20

【 0 1 5 6 】

ステップS 4 5で“ N O ”であれば、つまり追跡サーバ1 0が起動してから一定時間が経過していなければ、ステップS 4 1に戻る。つまり、位置データおよび状態データが記憶されるまでは、ロボット1 8と実体Eとを関連付けることができないため、ステップS 4 1 - S 4 5の処理が繰り返し実行される。

【 0 1 5 7 】

また、ステップS 4 5で“ Y E S ”であれば、つまり追跡サーバ1 0が起動してから一定時間が経過すれば、初回フラグ3 4 8をオフにしてステップS 4 7で関連付解除処理を実行する。なお、ステップS 4 7の関連付解除処理については、図2 3に示すフロー図を用いて後述するため、ここでの詳細な説明は省略する。また、1 度目の処理では、関連付解除処理が実行されたとしても、ロボット1 8と実体Eとの関連付けが解除されることはない。

30

【 0 1 5 8 】

続いて、ステップS 4 9では関連付処理が実行される。この関連付処理については、図2 4に示すフロー図を用いて後述するため、ここでの詳細な説明は省略する。また、ステップS 4 9の処理を実行するプロセッサ2 0は、関連付手段として機能する。

【 0 1 5 9 】

続いて、ステップS 5 1では、変数i xを初期化する。この変数i xは、ロボットI Dを指定するための変数である。そのため、変数i xが初期化されると「 1 」が設定される。なお、以下の説明では、ロボットI Dが「 i x 」のロボット1 8は「ロボット_{i x}」と記述する。たとえば、ロボットI Dが「 0 0 1 」のロボット1 8は「ロボット₁」と記述される。これは、図面でも同じである。

40

【 0 1 6 0 】

続いて、ステップS 5 3では、ロボット_{i x}に対応するR T Mの位置を更新する。たとえば、変数i xが「 1 」であれば、ロボットI Dが「 0 0 1 」のロボット1 8 aのR T Mの位置が更新される。また、更新するための位置データは、位置履歴バッファ3 3 2に記憶される、ロボット₁の位置履歴データから読み出される。従って、ロボット₁の位置履歴データにおいて最新の位置が(3 0 0 , 4 0 0)であれば、ロボット₁のR T Mの位置が(3 0 0 , 4 0 0)に更新される。続いて、ステップS 5 5では、角度補正処理を実行

50

する。なお、この角度補正処理については、図 27 に示すフロー図を用いて後述するため、ここでの詳細な説明は省略する。

【0161】

続いて、ステップ S57 では、変数 i_x をインクリメントする。つまり、次のロボット ID を指定するために、変数 i_x はインクリメントされる。続いて、ステップ S59 では、変数 i_x が変数 R_n より大きいか否かを判断する。つまり、ロボット 18 の総数を示す変数 R_n よりも、ロボット ID を指定する変数 i_x が大きいか否かを判断する。これにより、全てのロボット 18 に対して、RTM の位置および角度が更新されたか否かを判断することができる。ステップ S59 で “NO” であれば、つまり変数 i_x が変数 R_n 以下であれば、ステップ S53 に戻る。つまり、他のロボット 18 の RTM の位置および角度を
10 更新するためにステップ S53 に戻る。一方、ステップ S59 で “YES” であれば、つまり変数 i_x が変数 R_n より大きければ、ステップ S41 に戻る。つまり、全てのロボット 18 の RTM の位置および角度が更新されると、補正処理が再び実行される。

【0162】

図 21 には追跡プログラム 314a の処理を示すフロー図が示される。プロセッサ 20 は、ステップ S41 で追跡処理が実行されると、ステップ S71 で LRF データを取得する。つまり、LRF パッファ 330 に一時記憶される、各 LRF 12 によって計測された距離のデータを取得する。続いて、ステップ S73 では、新しい実体 E を検出したか否かを判断する。たとえば、検出領域 F 内で、実体 $E_a - E_d$ とは別の実体 E が検出されたか否かを判断する。ステップ S73 で “NO” であれば、つまり新たな実体 E が検出されて
20 いなければ、ステップ S79 に進む。一方、ステップ S73 で “YES” であれば、つまり検出領域 F 内で新たな実体 E が検出されれば、ステップ S75 で実体 E の数を変数 E_n に設定する。たとえば、検出領域 F 内で、4 つの実体 E_a, E_b, E_c, E_d が検出されていれば、変数 E_n には「4」が設定される。なお、変数 E_n は、検出されている実体 E の総数を示す変数として定義されている。

【0163】

続いて、ステップ S77 では、 E_n 個の位置履歴データを作成する。たとえば、変数 E_n に「4」が設定されていれば、図 7 に示すように、位置履歴パッファ 332 に 4 つの位置履歴データが作成される。続いて、ステップ S79 では、変数 I_x を初期化する。この変数 I_x は、実体 ID を指定するための変数である。そのため、変数 I_x が初期化されると「1」が設定される。なお、以下の説明では、実体 ID が「 I_x 」の実体 E は「実体 I_x 」と記述する。たとえば、実体 ID が「001」の実体 E は、「実体 $_1$ 」と記述される。
30 これは、図面でも同じである。

【0164】

続いて、図 22 を参照して、ステップ S81 では、実体 I_x にロボット 18 が関連付けられているか否かを判断する。たとえば、変数 I_x に「1」が設定されていれば、関連付けテーブルを参照し、実体 ID が「001」の実体 E_a にロボット 18 が関連付けられているか否かを判断する。ステップ S81 で “YES” であれば、つまり実体 I_x にロボット 18 が関連付けられてい
40 れば、ロボット形状モデルを利用して、パーティクルフィルタの尤度を更新する。つまり、ステップ S83 では、ロボット 18 に適したモデルを使用して、数 1 および数 2 に示す式に従って、パーティクルフィルタの尤度を更新する。一方、ステップ S81 で “NO” であれば、つまり実体 I_x にロボット 18 が関連付けられていなければ、ステップ S85 で人形状モデルを利用して、パーティクルフィルタの尤度を更新する。つまり、ステップ S85 では、人間に適したモデルを利用して、数 1 および数 2 に示す式に従って、パーティクルフィルタの尤度を更新する。

【0165】

続いて、ステップ S87 では、パーティクルの位置を更新する。つまり、ステップ S83 およびステップ S85 の処理で、算出されたパーティクルフィルタの尤度に基づいてパーティクルの位置が更新される。続いて、ステップ S89 では、実体 I_x の位置を検出する。つまり、移動したパーティクルの位置に基づいて、実体 I_x の位置、つまり検出領域
50

F内の平面座標を検出する。なお、ステップS89の処理を実行するプロセッサ20は、位置検出手段として機能する。

【0166】

続いてステップS91では、検出された実体 I_x の位置をバッファに記憶する。つまり、位置データは、位置履歴バッファ332に記憶される位置履歴データにおいて、実体ID「IX」に基づいて追加される。たとえば、変数IXに「1」が設定されていれば、実体IDが「001」の位置履歴データに、実体 I_1 の位置データが追加される。なお、ステップS91の処理を実行するプロセッサ20は、位置データ記憶手段として機能する。

【0167】

続いて、ステップS93では、変数IXをインクリメントする。つまり、次の実体IDを指定するために、変数IXはインクリメントされる。続いて、ステップS95では、変数IXが変数Enよりも大きいかなかを判断する。つまり、実体Eの総数を示す変数Enよりも、実体IDを指定する変数IXが大きいかなかを判断する。これにより、全ての実体Eに対して位置が検出されたかなかを判断することができる。ステップS95で“NO”であれば、つまり全ての実体Eに対して位置が検出されていないならば、ステップS81に戻る。一方、ステップS95で“YES”であれば、つまり全ての実体Eに対して位置が検出されれば、追跡処理を終了して、補正処理に戻る。

【0168】

図23には、関連付解除プログラム314bの処理を示すフロー図が示される。ステップS47で関連付処理が実行されると、プロセッサ20は、ステップS111でロボット18が関連付けられているかなかを判断する。つまり、関連付テーブルデータ344が読み出され、少なくとも1台のロボット18が実体Eと関連付けられているかなかを判断する。ステップS111で“NO”であれば、つまり全てのロボット18が実体Eと関連付けられていなければ、関連付解除処理を終了して、補正処理に戻る。たとえば、追跡サーバ10が起動し、ステップS49の関連付処理が一度も実行されていない状態では、ステップS111で“NO”と判断される。

【0169】

一方、ステップS111で“YES”であれば、つまり少なくとも1台のロボット18が実体Eと関連付けられていれば、ステップS113で関連付けられたロボット18の数を変数ARnに設定する。たとえば、実体Eと関連付けられているロボット18が3台であれば、変数ARnには「3」が設定される。また、変数ARnは、実体Eと関連付けられているロボット18の総数を示す変数として定義されている。

【0170】

続いて、ステップS115では、変数Dxを初期化する。たとえば、関連付け解除処理では、関連付けられているロボット18には、第1ローカルIDが付与される。そして、変数Dxは、その第1ローカルIDを指定するための変数である。そのため、変数Dxが初期化されると、第1ローカルIDの最初のIDを示す「1」が設定されている。

【0171】

なお、第1ローカルIDは、関連付解除処理が実行される毎に付与される。つまり、同じロボット18であっても、必ずしも同じ第1ローカルIDが付与されるとは限らない。また、ロボットIDと第1ローカルIDとも必ずしも一致するとは限らない。そして、変数ixと同様に、以下の説明では、第1ローカルIDが「Dx」のロボット18は、「ロボット D_x 」と記述する。そして、この記述は図23でも同じである。

【0172】

次のステップS117では、ロボット D_x と関連付けられた実体Eがまだ追跡されているかなかを判断する。つまり、関連付テーブルデータ344に基づいて、ロボット D_x に対応する実体Eの実体IDを読み出す。そして、位置履歴バッファ332を参照して、読み出した実体IDに対応する位置履歴データが存在するかなかを判断する。ステップS117で“NO”であれば、つまりロボット D_x と関連付けられた実体Eが検出領域F内に存在していなければ、ステップS125に進む。

【 0 1 7 3 】

また、ステップ S 1 1 7 で “ Y E S ” であれば、つまりロボット D_x と関連付けられた実体 E が検出領域 F 内に存在していれば、ステップ S 1 1 9 で、ロボット D_x と関連付けられた実体 E が離れているか否かを判断する。たとえば、ステップ S 1 1 7 と同様、ロボット D_x に対応する実体 I D を読み出し、読み出した実体 I D に対応する位置履歴データの最新の位置データを、位置履歴バッファ 3 3 2 から取得する。さらに、ロボット I D が「 D_x 」の状態データから最新の状態データに含まれる位置データを、状態履歴バッファ 3 3 4 から取得する。そして、読み出された状態データの位置データと、位置履歴データの位置データとを比較して、たとえば 2 m 以上離れているか否かを判断する。ステップ S 1 1 9 で “ Y E S ” であれば、つまりロボット D_x と関連付けられた実体 E が離れてい

10

【 0 1 7 4 】

また、ステップ S 1 1 9 で “ N O ” であれば、たとえばロボット D_x と関連付けられた実体 E が 2 m 以内に存在していれば、ステップ S 1 2 1 でロボット D_x と関連付けられた実体 E との速度が異なるか否かを判断する。ステップ S 1 1 9 と同様に、位置履歴データから現在の位置および 1 秒前の位置を取得して、実体 E の速度を算出する。また、ステップ S 1 1 9 と同様に、状態履歴データから現在の位置および 1 秒前の位置を取得して、ロボット D_x の速度を算出する。そして、ロボット D_x の速度と、関連付けられた実体 E の速度との差が、たとえば 0 . 3 5 mm/s 以上であるか否かを判断する。ステップ S 1 2 1 で “ Y E S ” であれば、たとえば 0 . 3 5 mm/s 以上の速度差があれば、ステップ S 1 2 5 に進む。また、ステップ S 1 2 1 で “ N O ” であれば、たとえば 0 . 3 5 mm/s 以上の速度差がなければ、ステップ S 1 2 3 でロボット D_x の関連付けを維持する。たとえば、ステップ S 1 2 3 では、ロボット D_x の関連付けを解除する必要がないことを示すフラグをオンにする。そして、ステップ S 1 2 3 の処理が終了すると、ステップ S 1 2 7 に進む。

20

【 0 1 7 5 】

ここで、ステップ S 1 1 7 で “ N O ”、ステップ S 1 1 9 で “ Y E S ” またはステップ S 1 2 1 で “ Y E S ” と判断され、ステップ S 1 2 5 の処理が実行されると、ロボット D_x の関連付けを解除する。つまり、ステップ S 1 2 5 では、関連付テーブルデータ 3 4 4 において、ロボット D_x と関連付けられている実体 I D を削除する。また、ステップ S 1 2 5 で関連付けが解除されると、解除カウンタ 3 5 2 によって、関連付けが解除されてからの時間が計測される。

30

【 0 1 7 6 】

なお、ステップ S 1 1 7、S 1 1 9、S 1 2 1 の処理を実行するプロセッサ 2 0 は判断手段として機能する。また、ステップ S 1 2 5 の処理を実行するプロセッサ 2 0 は、解除手段として機能する。

【 0 1 7 7 】

続いて、ステップ S 1 2 7 では、変数 D_x をインクリメントする。つまり次の第 1 ローカル I D を指定するために、変数 D_x はインクリメントされる。続いて、ステップ S 1 2 9 では、変数 D_x が変数 $A R n$ より大きいか否かを判断する。つまり、実体 E と関連付けられたロボット 1 8 の総数を示す変数 $A R n$ よりも、第 1 ローカル I D を指定する変数 D_x が大きいか否かを判断する。これにより、実体 E と関連付けられた全てのロボット 1 8 に対して、関連付けの解除が必要であるか否かを判断することができる。

40

【 0 1 7 8 】

ステップ S 1 2 9 で “ N O ” であれば、つまり変数 D_x が変数 $A R n$ 以下であれば、ステップ S 1 1 7 に戻る。つまり、他のロボット 1 8 において、関連付けの解除が必要であるか否かを判断するためにステップ S 1 1 7 に戻る。一方、ステップ S 1 2 9 で “ Y E S ” であれば、つまり変数 D_x が変数 $A R n$ より大きければ、関連付解除処理を終了して、補正処理に戻る。

【 0 1 7 9 】

なお、解除カウンタ 3 5 2 は、関連付けが解除される度にリセットされて、時間が計測

50

される。たとえば、関連付解除処理において、2つのロボット18の関連付けが解除された場合には、後に関連付けが解除されたロボット18に対してのみ、関連付けが解除されてからの時間が計測される。

【0180】

図24には関連付プログラム314cの処理を示すフロー図が示される。ステップS49で関連付処理が実行されると、プロセッサ20は、ステップS151で関連付けられていないロボット18があるか否かを判断する。つまり、関連付テーブルデータ344が読み出され、実体Eが関連付けられていないロボット18があるか否かを判断する。ステップS151で“NO”であれば、つまり全てのロボット18が実体Eと関連付けられていれば、関連付処理を終了して、補正処理に戻る。

10

【0181】

一方、ステップS151で“YES”であれば、つまり実体Eと関連付けられていないロボット18があれば、ステップS153で関連付けられていないロボット18の数を変数DRnに設定する。たとえば、実体Eと関連付けが解除されたロボット18が1台であれば、変数DRnには「1」が設定される。なお、変数DRnは、実体Eと関連付けが解除されたロボット18の総数を示す変数として定義されている。

【0182】

続いて、ステップS155では、変数axを初期化する。たとえば、関連付処理では、関連付けられていないロボット18には、第2ローカルIDが付与される。そして、変数axは、第2ローカルIDを指定するための変数である。そのため、変数axが初期化されると、第2ローカルIDの最初のIDを示す「1」が設定される。

20

【0183】

なお、第2ローカルIDは、第1ローカルIDと同様に、関連付処理が実行される毎に付与される。つまり、同じ第2ローカルIDが、同じロボット18に付与されるとは限らない。また、変数Dxと同様に、以下の説明では、第2ローカルIDが「ax」のロボット18を「ロボット_ax」と記述する。そして、この記述は図面でも同じである。

【0184】

続いて、ステップS157では、追跡実体リストを作成する。たとえば、図12に示す追跡実体リストを、実体Eの総数を示す変数Enに基づいて作成する。そして、作成した追跡実体リストを追跡実体リストバッファ336に格納する。続いて、ステップS159では、5秒(所定時間)以内に関連付けが解除されたか否かを判断する。つまり、5秒以内に実体Eとロボット18との関連付けが解除されたか否かを判断する。また、具体的には、解除カウンタ352の値が5秒を示す値以下であるか否かを判断する。ステップS159で“NO”であれば、つまり実体Eとロボット18との関連付けが解除されて5秒を超えていれば、ステップS171に進む。

30

【0185】

一方、ステップS159で“YES”であれば、つまり実体Eとロボット18との関連付けが解除されてから5秒以内であれば、ステップS161で変数IXを初期化する。つまり、後述するローカル関連付処理で、全ての実体Eを識別するため、変数IXを初期化する。なお、初期化された変数IXには「1」が設定される。続いて、ステップS163では、追跡実体リストを初期化する。つまり、追跡実体リストのフラグの列の全てを「1(オン)」に設定する。

40

【0186】

続いて、ステップS165では、ローカル関連付処理を実行する。なお、ステップS165のローカル関連付処理については、図25に示すフロー図を用いて後述するため、ここでの詳細な説明は省略する。続いて、ステップS167では、変数axをインクリメントする。つまり、次の第2ローカルIDを指定するために、変数axはインクリメントされる。続いて、ステップS169では、変数axが変数DRnより大きいかなんかを判断する。つまり、関連付けられていないロボット18の総数を示す変数DRnよりも、第2ローカルIDを指定する変数axの方が大きいかなんかを判断する。これにより、関連付けら

50

れていない全てのロボット 18 に対して、ローカル関連付処理が実行されたか否かを判断することができる。

【0187】

ステップ S 169 で “NO” であれば、つまり変数 a_x が変数 DR_n 以下であれば、ステップ S 161 に戻る。つまり、関連付けられていない他のロボット 18 に対してローカル関連付け処理を実行するために、ステップ S 161 に戻る。一方、ステップ S 169 で “YES” であれば、つまり変数 a_x が変数 DR_n より大きければ、関連付け処理を終了して、補正処理に戻る。

【0188】

ここで、ロボット 18 と実体 E との関連付けが 5 秒以内に解除されていなければ、ステップ S 171 - S 179 の処理を実行する。なお、ステップ S 171 - S 179 で、ステップ S 161 - S 169 と同じ処理については、詳細な説明を省略する。

10

【0189】

ステップ S 171 では、変数 IX を初期化し、ステップ S 173 では追跡実体リストを初期化する。続いて、ステップ S 175 では、グローバル関連付処理を実行する。このグローバル関連付け処理については、図 26 に示すフロー図を用いて後述するため、ここでの詳細な説明は省略する。

【0190】

続いて、ステップ S 177 では、変数 a_x をインクリメントし、ステップ S 179 では、変数 a_x が変数 DR_n より大きいかなんかを判断する。つまり、ステップ S 179 では、関連付けられていない全てのロボット 18 に対して、グローバル関連付処理が実行されたか否かを判断する。ステップ S 179 で “NO” であれば、つまり実体 E と関連付けられていない全てのロボット 18 に対してグローバル関連付処理が実行されていなければ、ステップ S 171 に戻る。一方、ステップ S 179 で “YES” であれば、つまり実体 E と関連付けられていない全てのロボット 18 に対してグローバル関連付処理が実行されてい

20

【0191】

図 25 にはローカル関連付プログラム 316 の処理を示すフロー図が示される。ステップ S 165 でローカル関連付処理が実行されると、プロセッサ 20 は、ステップ S 191 で実体 IX が他のロボット 18 に関連付けられているかなんかを判断する。つまり、関連付テーブル 344 が読み出され、実体 IX と他のロボット 18 とが関連付けられているかなんかを判断する。ステップ S 191 で “YES” であれば、つまり実体 IX が他のロボット 18 と関連付けられていれば、ステップ S 201 に進む。一方、ステップ S 191 で “NO” であれば、つまり実体 IX が他のロボット 18 と関連付けられていなければ、ステップ S 193 で実体 IX とロボット a_x との 5 秒間（第 1 時間）の RMS 速度差（第 1 速度差）を算出する。つまり、ステップ S 193 では、位置履歴バッファ 332 から、5 秒分の実体 IX の位置履歴データを読み出し、状態履歴バッファ 334 から、5 秒分のロボット a_x の状態履歴データを読み出す。そして、数 3 において RT を 5 として、RMS 速度差を算出する。なお、ステップ S 193 の処理を実行するプロセッサ 20 は、第 1 速度差算出手段として機能する。

30

40

【0192】

続いて、ステップ S 195 では、実体 IX とロボット a_x との RMS 速度差が大きいかなんかを判断する。たとえば、ステップ S 193 で算出された RMS 速度差が第 1 所定速度（350 mm/s）より大きいかなんかを判断する。ステップ S 195 で “YES” であれば、つまり RMS 速度差が第 1 所定速度より大きければ、ステップ S 201 に進む。なお、ステップ S 195 の処理を実行するプロセッサ 20 は第 1 速度判定手段として機能する。

【0193】

一方、ステップ S 195 で “NO” であれば、つまり RMS 速度差が第 1 所定速度以下であれば、ステップ S 197 で実体 IX とロボット a_x との現在の速度差が大きいかなんかを判断する。つまり、ステップ S 193 で読み出された実体 IX の位置履歴データおよび

50

ロボット a_x の状態履歴データから、実体 I_x の速度およびロボット a_x の速度を算出して、現在の速度差を求める。そして、現在の速度差が、たとえば 350 mm/s より大きいかな否かを判断する。ステップ S 197 で “YES” であれば、つまり現在の速度差が 350 mm/s より大きければ、ステップ S 201 に進む。

【0194】

また、ステップ S 197 で “NO” であれば、つまり現在の速度差が 350 mm/s 以下であれば、ステップ S 199 で実体 I_x とロボット a_x とが離れているかな否かを判断する。つまり、ステップ S 199 では、ステップ S 193 で読み出された実体 I_x の位置履歴データおよびロボット a_x の状態履歴データから、実体 I_x の位置データおよびロボット a_x の位置を取得する。そして、実体 I_x とロボット a_x との距離が、たとえば 1 m より大きいかな否かをステップ S 199 で判断する。ステップ S 199 で “NO” であれば、つまり実体 I_x とロボット a_x とが離れていなければ、ステップ S 203 に進む。なお、ステップ S 199 の処理を実行するプロセッサ 20 は、距離判定手段として機能する。

10

【0195】

一方、ステップ S 199 で “YES” であれば、つまり実体 I_x とロボット a_x とが離れていれば、ステップ S 201 で追跡実体リストにおける、実体 I_x のフラグをオフにする。たとえば、変数 I_x が「1」であれば、実体 E_a に対応するフラグの欄において、「0 (オフ)」が設定される。

【0196】

続いて、ステップ S 203 では、変数 I_x をインクリメントする。つまり、次の実体 I_D を指定するために、変数 I_x はインクリメントされる。続いて、ステップ S 205 では、変数 I_x が変数 E_n より大きいかな否かを判断する。つまり、全ての実体 E に対して、ステップ S 191 からステップ S 201 の処理が実行されたかな否かを判断する。ステップ S 205 で “NO” であれば、つまり全ての実体 E に対して、ステップ S 191 からステップ S 201 の処理が実行されていないならばステップ S 191 に戻る。一方、ステップ S 205 で “YES” であれば、つまり全ての実体 E に対して、ステップ S 191 からステップ S 201 の処理が実行されていれば、ステップ S 207 でフラグがオンの実体 E があるかな否かを判断する。つまり、追跡実体リストにおいて、フラグの列に「1」が設定された実体 E があるかな否かを判断する。

20

【0197】

ステップ S 207 で “YES” であれば、つまり追跡実体リストにおいて、フラグの列に「1」が設定された実体 E があれば、ステップ S 209 で最も近い実体 E をロボット a_x に関連付ける。つまり、位置履歴バッファ 332 から、追跡実体リストにおいてフラグがオンの実体 E の現在の位置データを読み出す。また、状態履歴バッファ 334 から、ロボット a_x の最新の位置データを読み出す。そして、フラグがオンの実体 E のそれぞれと、ロボット a_x の距離を算出し、最も距離が短い実体 E とロボット a_x とを関連付ける。また、ステップ S 209 で関連付けられた結果が関連付テーブルに記録される。

30

【0198】

たとえば、実体 E_a および実体 E_b のフラグがオンであり、ロボット a_x がロボット 18a であれば、実体 E_a および実体 E_b のそれぞれと、ロボット 18a との距離が算出される。そして、ロボット 18a と実体 E_b との距離の方が短ければ、ロボット 18a と実体 E_b とが関連付けられる。これにより、図 13 に示す関連付テーブルのように、ロボット 18a のロボット I_D に対して、実体 E_b の実体 I_D が記録される。なお、ステップ S 209 の処理を実行するプロセッサ 20 は第 1 関連付手段として機能する。

40

【0199】

一方、ステップ S 207 で “NO” であれば、つまりフラグがオンの実体 E が追跡実体リストになければ、ステップ S 211 でロボット a_x に実体 E を関連付けない。つまりステップ S 211 が実行されると、関連付テーブルに対して実体 I_D が記録されない。

【0200】

そして、ステップ S 209 またはステップ S 211 の処理が終了すれば、ローカル関連

50

付処理を終了して、関連付処理に戻る。

【0201】

図26にはグローバル関連付プログラム318の処理を示すフロー図が示される。ここで、グローバル関連付処理において、ローカル関連付処理と同じ処理については、詳細な説明を省略する。

【0202】

ステップS175でグローバル関連付け処理が実行されると、ステップS231では、実体 I_x が他のロボット18と関連付けられているか否かを判断する。ステップS231で“YES”であれば、つまり実体 I_x が他のロボット18と関連付けられていれば、ステップS239に進む。一方、ステップS231で“NO”であれば、つまり実体 I_x が他のロボット18と関連付けられていなければ、ステップS233で実体 I_x とロボット a_x との300秒間(第2時間)のRMS速度差(第2速度差)を算出する。なお、ステップS233の処理を実行するプロセッサ20は第2速度差算出手段として機能する。

10

【0203】

続いて、ステップS235では実体 I_x とロボット a_x とのRMS速度差が大きいか否かを判断する。たとえば、実体 I_x とロボット a_x とのRMS速度差が第2所定速度(350mm/s)より大きいか否かを判断する。ステップS235で“YES”であれば、つまり実体 I_x とロボット a_x とのRMS速度差が第2所定速度より大きければ、ステップS239に進む。なお、ステップS235の処理を実行するプロセッサ20は、第2速度判定手段として機能する。

20

【0204】

一方、ステップS235で“NO”であれば、つまり実体 I_x とロボット a_x とのRMS速度差が第2所定速度以下であれば、ステップS237で実体 I_x とロボット a_x との現在の速度差が大きいか否かを判断する。たとえば、現在の速度差が350mm/sより大きいか否かを判断する。

【0205】

ステップS237で“YES”であれば、つまり現在の速度差が350mm/sより大きければ、ステップS239で追跡実体リストにおける実体 I_x のフラグをオンにして、ステップS241に進む。一方、ステップS237で“NO”であれば、つまり現在の速度差が350mm/s以下であれば、ステップS241で変数IXをインクリメントし、ステップS243で変数IXが変数Enより大きいか否かを判断する。ステップS243で“NO”であれば、つまり変数IXが変数En以下であれば、ステップS231に戻る。一方、ステップS243で“YES”であれば、つまり変数IXが変数Enよりも大きければ、ステップS245でフラグがオンの実体Eが1つだけか否かを判断する。つまり、ステップS245では、追跡実体リストのフラグの列において、「1」が設定された実体Eが1つだけであるか否かを判断する。

30

【0206】

ステップS245で“NO”であれば、ステップS247でフラグがオンの実体Eにロボット a_x を関連付ける。たとえば、ロボット a_x がロボット18bを示し、追跡実体リストで実体Ecのフラグのみがオンであれば、図13に示す関連付テーブルのように、ロボットID「002」に対して、実体ID003が記録される。なお、ステップS247の処理を実行するプロセッサ20は第2関連付手段として機能する。

40

【0207】

一方、ステップS245で“NO”であれば、つまり、フラグがオンの実体Eが1つだけでなければ、ステップS211と同様に、ステップS249でロボット a_x に実体Eを関連付けない。

【0208】

そして、ステップS247またはステップS249の処理が終了すれば、グローバル関連付け処理を終了して、関連付け処理に戻る。

【0209】

50

図 27 には角度補正プログラム 314d の処理を示すフロー図が示される。ステップ S55 の角度補正処理が実行されると、プロセッサ 20 は、ステップ S261 で次の位置を予測する。具体的には、ロボット i_x の状態履歴データから、最新の位置および 1 秒前の位置を読み出し、縦 (Y) 軸および横 (X) 軸方向の位置変位量を算出する。そして、ロボット i_x の RTM において、前回の位置 (前回位置) に対して、算出した位置変位量を加えることで、図 14 に示すような予測位置を求める。

【0210】

続いて、ステップ S263 では、予測角度および検出角度を算出する。具体的には、予測角度は、ステップ S261 で求めた予測位置と前回位置との角度を、横軸を基準 (0 度) にして算出する。また、検出角度については、まず位置履歴バッファ 332 から、ロボット i_x に関連付けられた実体 E における、現在の位置 (データ) を読み出す。次に、ロボット i_x に関連付けられた実体 E における、現在の位置と前回位置との角度を検出角度とし、横軸を基準にして算出する。続いて、ステップ S265 では、検出角度および予測角度から観測角度 Z_k を算出する。たとえば、図 14 に示すように、検出角度から予測角度を減算することで、観測角度 Z_k を算出 (観測) する。なお、ステップ S261 - S265 の処理を実行するプロセッサ 20 は角度観測手段として機能する。

【0211】

続いて、ステップ S267 では、観測における位置誤差量 σ^2_p および変位量 d_s に基づいて、観測角度誤差量 $\sigma^2_{Z_k}$ を近似する。つまり、数 4 に従って、角度誤差量 $\sigma^2_{Z_k}$ を、位置誤差量 σ^2_p および変位量 d_s の商から近似する。なお、ステップ S267 の処理を実行するプロセッサ 20 は観測角度誤差算出手段として機能する。

【0212】

続いて、ステップ S269 では、前回値があるか否かを判断する。つまり、前回値バッファ 342 に前回の推定角度 θ_k および前回の推定角度誤差量 $\sigma^2_{\theta-k}$ が格納されているか否かを判断する。ステップ S269 で “NO” であれば、たとえば追跡サーバ 10 が起動してから角度補正処理が実行されるのが 1 度目であれば、ステップ S271 で観測角度誤差量 $\sigma^2_{Z_k}$ と所定値とに基づいてカルマンフィルタを更新する。つまり、数 5 において、前回の推定角度誤差量 $\sigma^2_{\theta-k}$ にかえて所定値 (たとえば、0) を代入して、カルマンゲイン K_k を算出する。

【0213】

続いて、ステップ S273 では、所定値を利用して、推定角度誤差量 $\sigma^2_{\theta-k}$ および推定角度 θ_k を算出し、ステップ S279 に進む。つまり、ステップ S273 では、数 6 に示す式において、前回の推定角度 θ_k に代えて所定値を代入して、推定角度 θ_k を算出する。また、数 7 に示す式において、前回の推定角度誤差量 $\sigma^2_{\theta-k}$ に代えて所定値を代入して、推定角度誤差量 $\sigma^2_{\theta-k}$ を算出する。

【0214】

また、ステップ S269 で “YES” であれば、つまり前回値バッファ 342 に前回の推定角度 θ_k および前回の推定角度誤差量 $\sigma^2_{\theta-k}$ が格納されていれば、ステップ S275 で観測角度誤差量 $\sigma^2_{Z_k}$ と前回値とに基づいてカルマンフィルタを更新する。つまり、数 5 において、前回値バッファ 342 から読み出された前回の推定角度誤差量 $\sigma^2_{\theta-k}$ を代入して、カルマンゲイン K_k を算出する。続いて、ステップ S277 では、前回値を利用して推定角度誤差量 $\sigma^2_{\theta-k}$ および推定角度 θ_k を算出する。つまり、ステップ S277 では、数 6 に示す式において、前回値バッファ 342 から読み出された前回の推定角度 θ_k を代入して推定角度 θ_k を算出する。また、数 7 に示す式において、前回値バッファ 342 から読み出された前回の推定角度誤差量 $\sigma^2_{\theta-k}$ を代入して推定角度誤差量 $\sigma^2_{\theta-k}$ を算出する。

【0215】

なお、ステップ S261 - S277 の処理を実行するプロセッサ 20 は、角度推定手段として機能する。

【0216】

10

20

30

40

50

続いて、ステップS279では、推定角度 $\theta^{\wedge}_k$ と、ロボット i_x のRTMの角度との角度差を算出する。たとえば、ロボット i_x のRTMの角度が32度であり、推定角度 $\theta^{\wedge}_k$ が30度であれば、角度差として-2度(=30-32)が算出される。なお、ステップS279の処理を実行するプロセッサ20は、角度補正算出手段として機能する。

【0217】

続いて、ステップS281では、推定角度誤差量 $\sigma^2_{\theta^{\wedge}_k}$ を補正精度とし、算出された角度差を補正角度データとして一時記憶する。つまり、推定角度誤差量 $\sigma^2_{\theta^{\wedge}_k}$ は補正精度バッファ340に格納され、算出された角度差は補正角度バッファ338に格納される。なお、ステップS281の処理を実行するプロセッサ20は記憶手段として機能する。

10

【0218】

続いて、ステップS283では、推定角度 $\theta^{\wedge}_k$ に基づいて、ロボット i_x のRTMの角度を更新する。たとえば、推定角度 $\theta^{\wedge}_k$ が30度であれば、ロボット i_x のRTMの角度が30度にされる。そして、ステップS283の処理が終了すれば、角度補正処理を終了する。

【0219】

なお、角度差および推定角度誤差量 $\sigma^2_{\theta^{\wedge}_k}$ には、ロボット i_x のロボットIDが対応付けられる。つまり、補正角度バッファ338および補正精度バッファ340には複数の角度補正データおよび補正精度が格納されるため、それぞれを識別可能にするため、ロボットIDが対応付けられる。

20

【0220】

また、角度補正処理では、前回の補正角度データが今回の状態データに含まれる角度データに反映される。たとえば、前回の角度補正処理で補正角度データが-2度と算出され、かつロボット18に送信されていなければ、今回の処理では、状態データに含まれる角度データから-2度が反映されたうえで実行される。ただし、補正角度データがロボット18に送信された場合には、補正角度バッファ338はリセットされる。つまり、補正角度データがロボット18に送信された場合に限り、前回の補正角度データが今回の状態データに含まれる角度データに反映される。

【0221】

この実施例によれば、ロボット自己位置同定システム100は、ネットワーク400を介して接続された追跡サーバ10およびロボット18を含む。追跡サーバ10はLRF12によって検出領域F内の実体Eをセンシングすることで実体Eを追跡し、ロボット18は状態データを出力する。

30

【0222】

追跡サーバ10は、ロボット18が出力する状態データおよび検出された実体Eの位置データをメモリ22のバッファに記憶する。また、追跡サーバ10は関連付処理を実行し、記憶した位置履歴データおよび状態履歴データに基づいて、各ロボット18と各実体Eとを関連付ける。さらに、追跡サーバ10は、各ロボット18に対応するRTMを、位置履歴データおよび状態履歴データに基づいて更新し、そのRTMに基づく補正データをロボット18に送信する。そして、ロボット18は、追跡サーバ10が送信した補正データに基づいて、自身の位置および角度を補正する。

40

【0223】

このように、本実施例では、容易に設置可能なLRF12およびロボット18が一般的に出力可能な状態データを利用して、ロボット18と実体Eとを関連付けることができるため、容易にロボット18を自己位置同定することができる。

【0224】

なお、ロボット自己位置同定システム100では、各サーバおよびロボット18に設定される時刻が一定時間(たとえば、20秒)毎に同期される。つまり、ロボット18が出力する状態データおよび追跡サーバ10が記憶する位置データに対応付けられる時刻データは、同時刻を示す。

50

【 0 2 2 5 】

また、遠隔操作装置 1 4 のオペレータが、ロボット 1 8 の位置データを修正してもよい。この場合、ロボット 1 8 は、追跡サーバ 1 0 に対してリセット信号を送信する。そして、ロボット 1 8 は、リセット信号を送信したときには、追跡サーバ 1 0 が送信した補正データに基づいて位置および角度を補正しない。さらに、追跡サーバ 1 0 は、リセット信号を受信した場合には、リセット信号が途絶した直後に取得した状態データに基づいて R T M を補正する。つまり、オペレータによってロボット 1 8 の位置が修正された場合には、追跡サーバ 1 0 が送信する補正データが、一時的に無効にされる。

【 0 2 2 6 】

また、ロボット自己位置同定システム 1 0 0 には、ローカライゼーション機能専用のサーバが含まれていてもよい。さらに、ロボット自己位置同定システム 1 0 0 は、遠隔操作装置 1 4 および計画サーバ 1 6 を含んでいなくてもよい。

10

【 0 2 2 7 】

また、本実施例のロボット自己位置同定システム 1 0 0 は、本実施例のロボット 1 8 だけに限らず、様々な種類のロボットを同時に自己位置同定することができる。そのため、ロボット 1 8 が出力する状態データの位置および角度などは、車輪速センサ 1 1 2 だけでなく、様々なセンサを利用して検出されてもよい。

【 0 2 2 8 】

また、本実施例の数値（所定時間、第 1 所定速度、第 2 所定速度、第 1 速度差、第 2 速度差、第 1 時間、第 2 時間、所定距離など）は、ロボット自己位置同定システム 1 0 0 が設置される環境に応じて、任意に変更されてもよい。

20

【 0 2 2 9 】

また、カルマンフィルタ以外にも、パーティクルフィルタおよび拡張カルマンフィルタなど、ロボット 1 8 の状態を推定することができるフィルタ（状態推定フィルタ）を利用して角度を補正することができる。

【 0 2 3 0 】

また、他の実施例では、R M S の代わりに相関係数を利用して、ロボット 1 8 と実体 E とを関連付けてもよい。

【 符号の説明 】

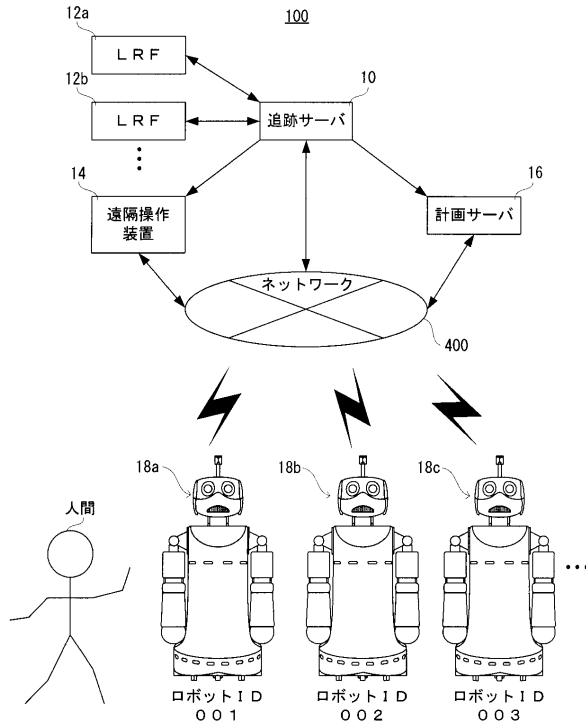
【 0 2 3 1 】

30

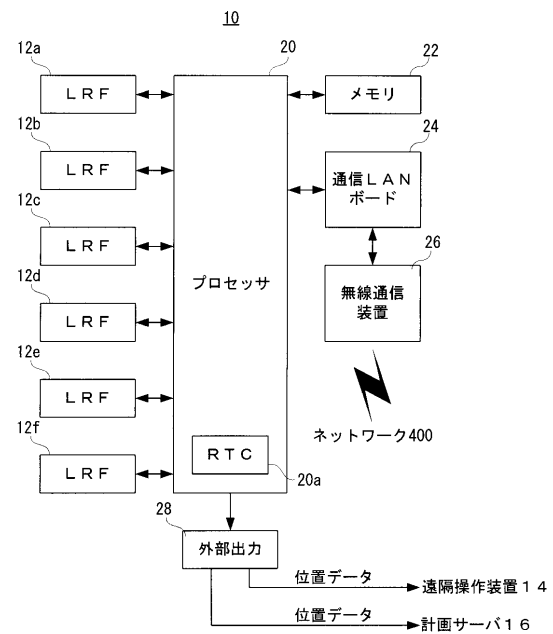
- 1 0 ... 追跡サーバ
- 1 2 a - 1 2 f ... L R F
- 1 8 ... ロボット
- 2 0 ... プロセッサ
- 2 2 ... メモリ
- 2 4 ... 通信 L A N ボード
- 2 6 ... 無線通信装置
- 3 2 a , 3 2 b ... 車輪
- 3 6 a , 3 6 b ... 車輪モータ
- 1 0 0 ... ロボット自己位置同定システム
- 1 1 2 ... 車輪速センサ
- 1 1 4 ... 通信 L A N ボード
- 1 1 6 ... 無線通信装置
- 4 0 0 ... ネットワーク

40

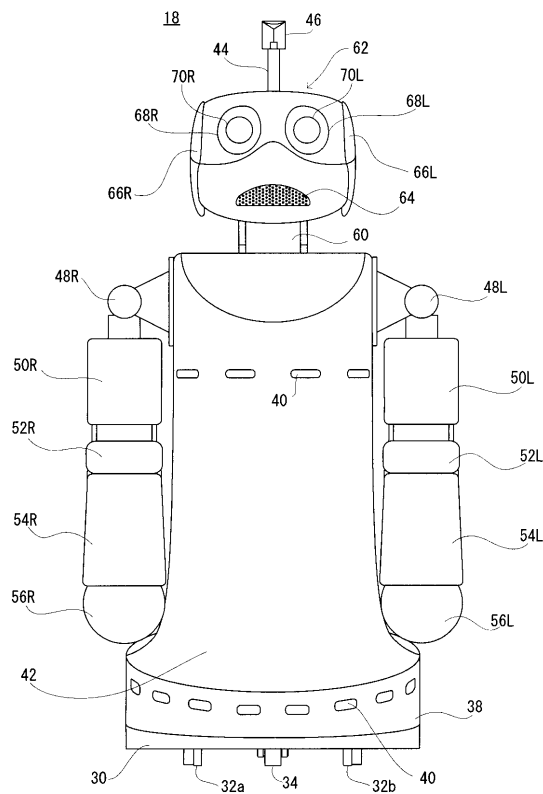
【図 1】



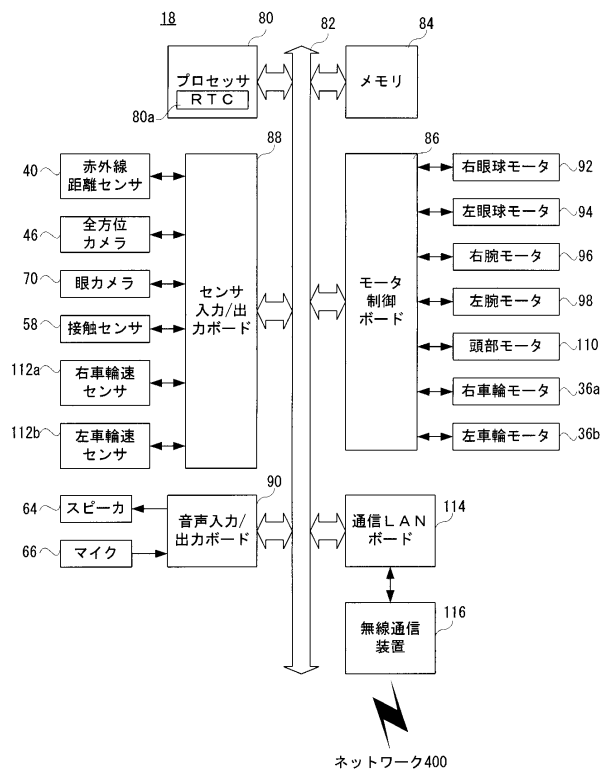
【図 2】



【図 3】

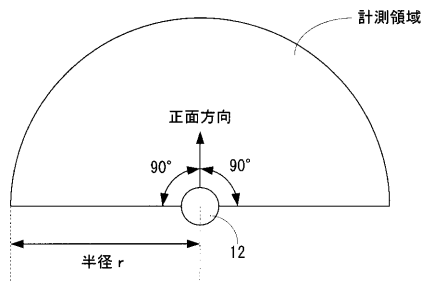


【図 4】



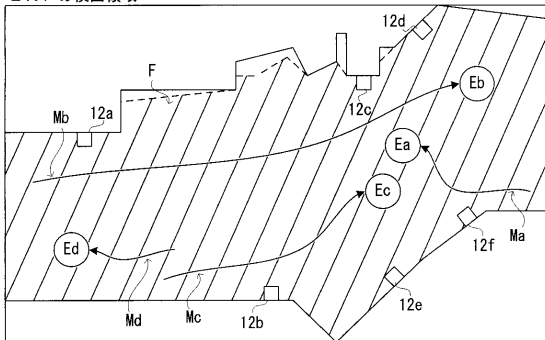
【図 5】

L R F の計測領域



【図 6】

L R F の検出領域



【図 7】

位置履歴データ

時刻	X	Y
T_1	X_1	Y_1
T_2	X_2	Y_2
T_3	X_3	Y_3
T_4	X_4	Y_4
T_5	X_5	Y_5
$\vdots$	$\vdots$	$\vdots$

実体 E a
実体 E b
実体 E c
実体 E d

【図 8】

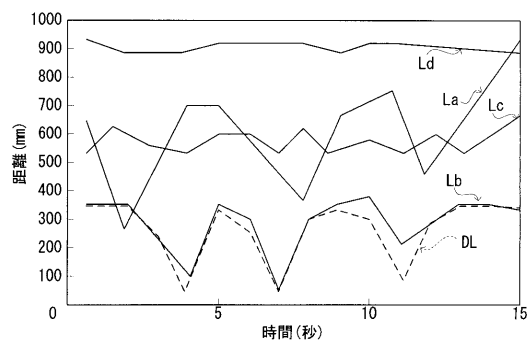
状態履歴データ

時刻	x	y	角度	速度 r	速度 l
t_1	x_1	y_1	d_1	rv_1	lv_1
t_2	x_2	y_2	d_2	rv_2	lv_2
t_3	x_3	y_3	d_3	rv_3	lv_3
t_4	x_4	y_4	d_4	rv_4	lv_4
t_5	x_5	y_5	d_5	rv_5	lv_5
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

ロボット 18 a
ロボット 18 b
ロボット 18 c

【図 10】

運動履歴グラフ



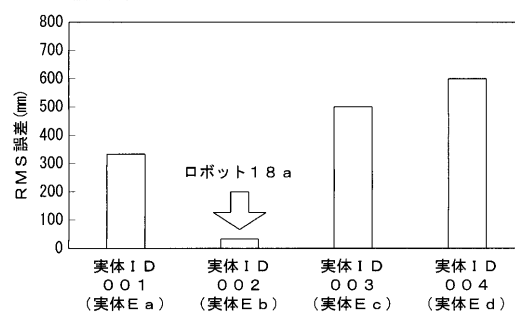
【図 9】

RTMデータ

		座標		角度	速度
		RX	RY		
ロボット I D	0 0 1	RX_1	RY_1	RD_1	RV_1
	0 0 2	RX_2	RY_2	RD_2	RV_2
	0 0 3	RX_3	RY_3	RD_3	RV_3
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

【図 11】

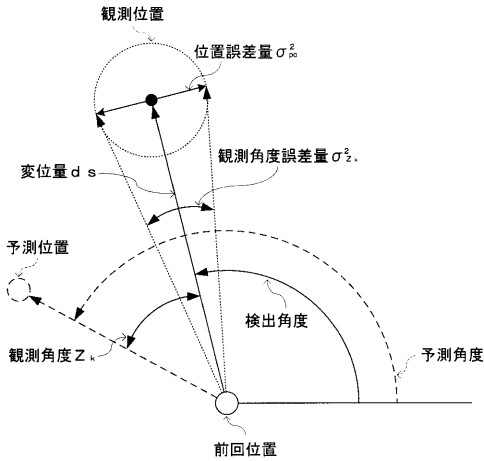
RMS 誤差グラフ



【図 1 2】

追跡実体リスト	
実体 I D	フラグ
0 0 1	0
0 0 2	1
0 0 3	0
0 0 4	0
⋮	⋮

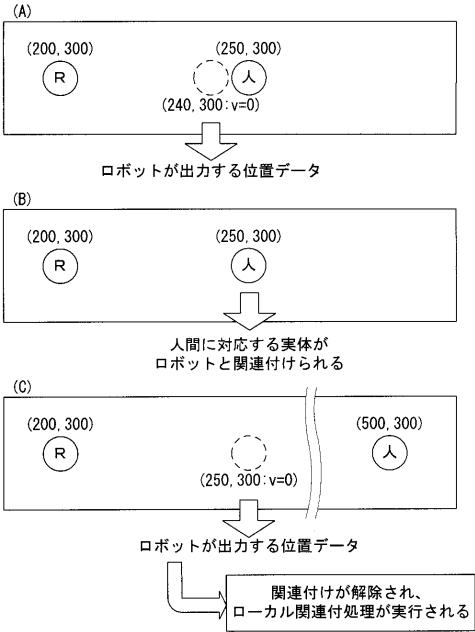
【図 1 4】



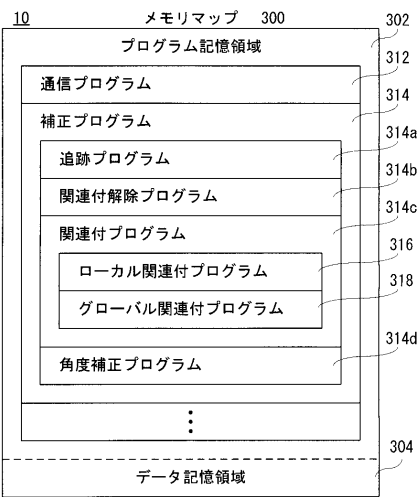
【図 1 3】

関連付テーブル		実体 I D
ロボット I D	0 0 1	0 0 2
	0 0 2	0 0 3
	0 0 3	0 0 1
	⋮	⋮

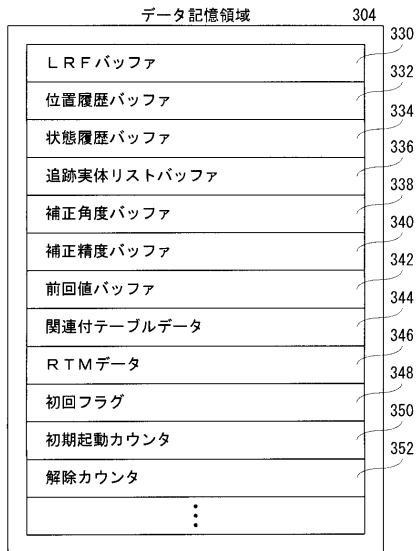
【図 1 5】



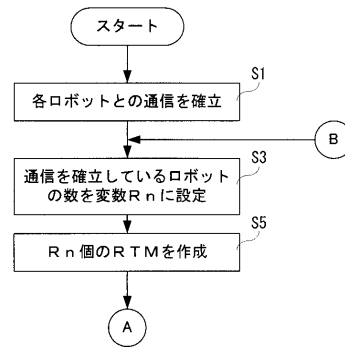
【図 1 6】



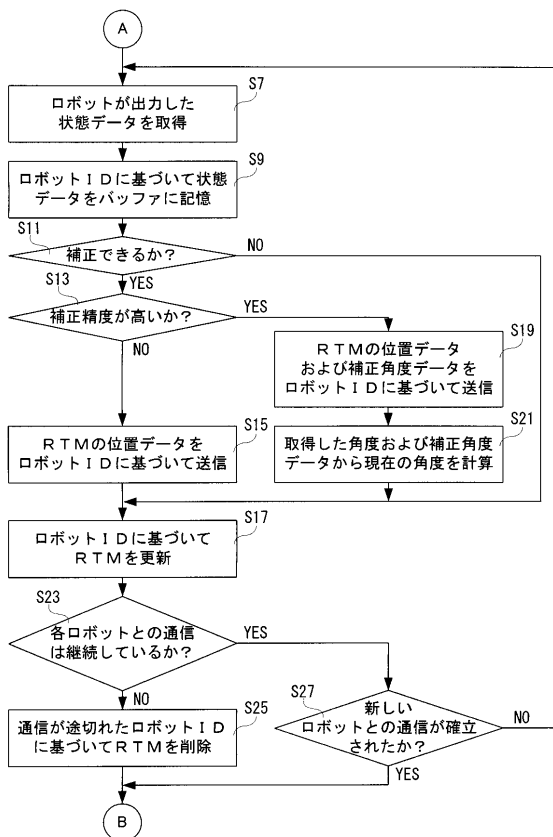
【図 17】



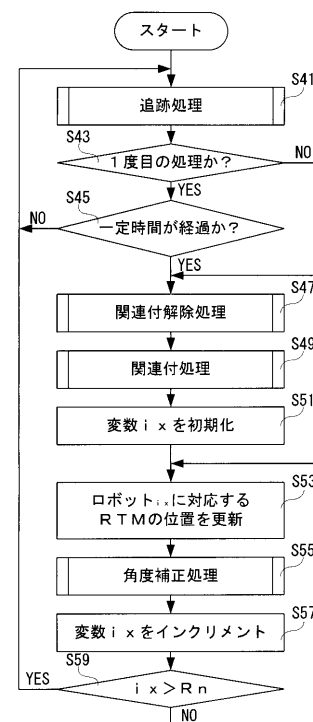
【図 18】



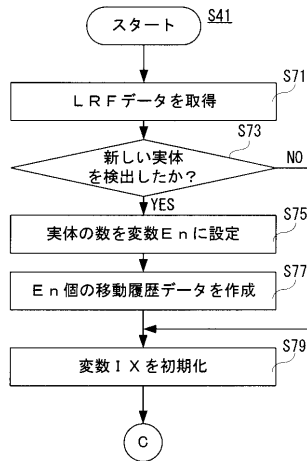
【図 19】



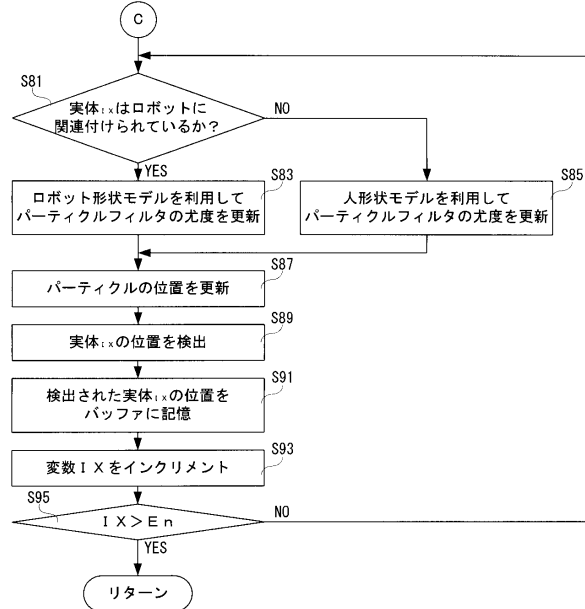
【図 20】



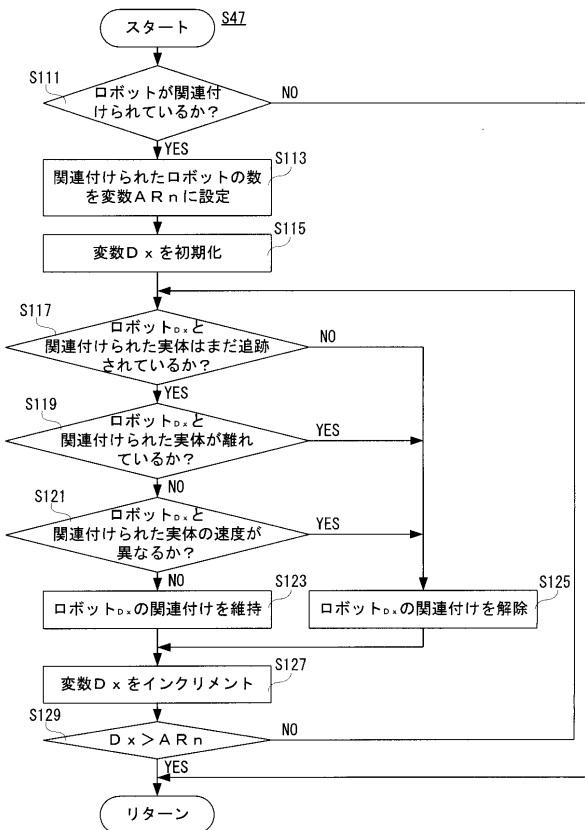
【図 2 1】



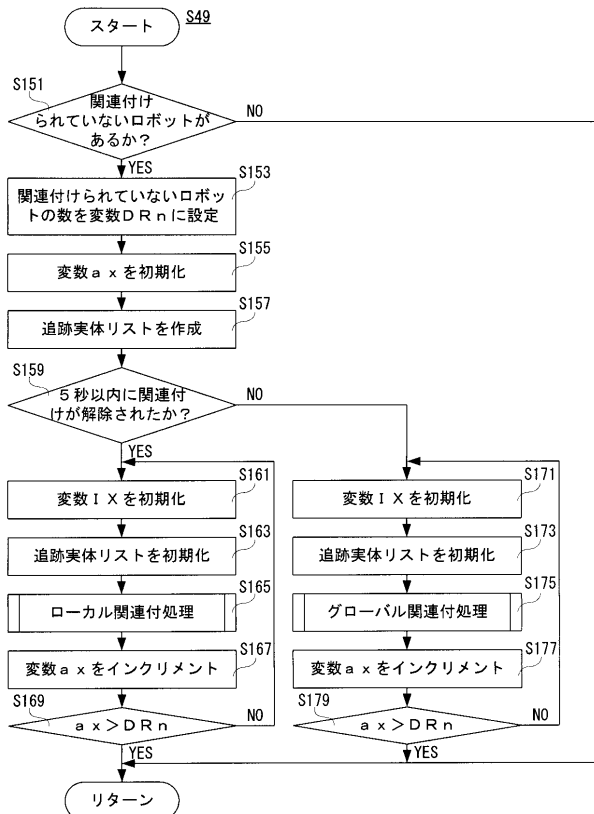
【図 2 2】



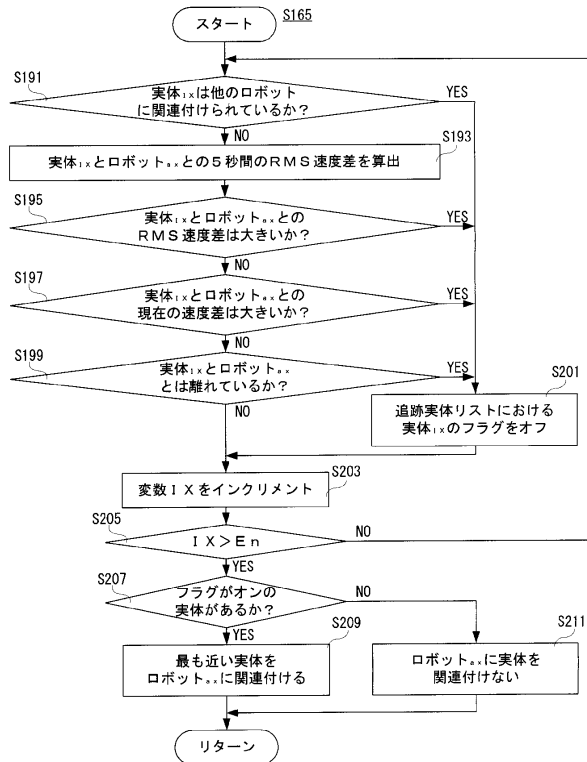
【図 2 3】



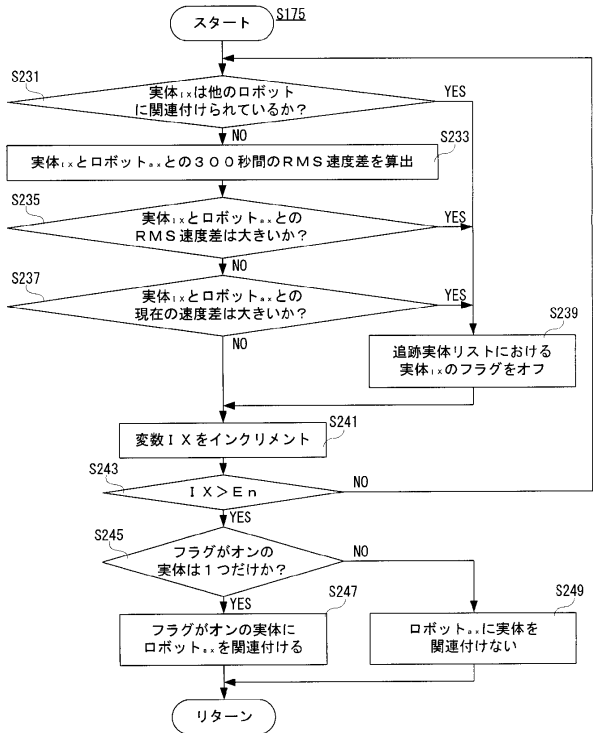
【図 2 4】



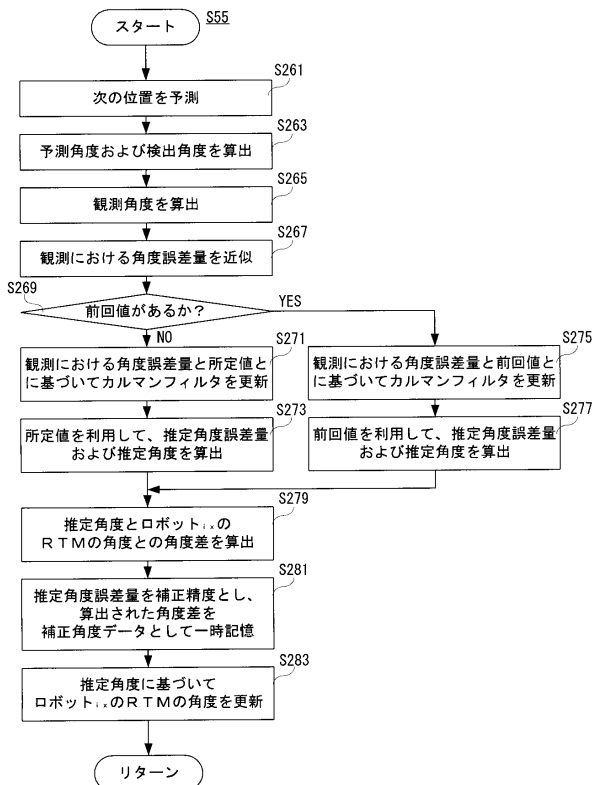
【 図 2 5 】



【 図 2 6 】



【 図 2 7 】



フロントページの続き

(出願人による申告)平成21年6月19日付け、支出負担行為担当官 総務省大臣官房会計課企画官、研究テーマ「高齢者・障害者のためのユビキタスネットワークロボット技術の研究開発」に関する委託研究、産業技術力強化法第19条の適用を受ける特許出願

(56)参考文献 特開2003-295951(JP,A)
特開2000-275321(JP,A)
特開2008-260107(JP,A)
国際公開第02/023297(WO,A1)
特開平02-244306(JP,A)
特開昭63-045610(JP,A)
特開2010-224755(JP,A)

(58)調査した分野(Int.Cl., DB名)

G05D 1/02
B25J 1/00-21/02