解説

車輪型移動ロボット制御とサンプリング時間

Wheeled Mobile Robot Control and Sampling Time

坪内 孝司* *筑波大学

Takashi Tsubouchi* *University of Tsukuba

1. はじめに

本稿の表題を推して本稿の内容を察すれば、一般的な車輪型移動ロボットの非ホロノーム的な制約下での走行制御との関連を読者はご想像になるのではないだろうか。一方、本稿執筆の依頼趣旨は、「移動ロボットの制御と位置認識、および同時に入ってくる外界センサとの情報のインテグレーションをどのようにするのか」の解説を求めるものであった。その背景の一つは、近年盛んに研究されている環境地図と自己位置の同時推定問題 (SLAM) にある。例えば、「制御が 1 [ms] ごと、外界センシングによる位置認識が 100 [ms] ごと、地図作成が数秒ごとというように時間的に異なる次元で行われるなかでどのようにシステムインテグレーションされるのか」という視点での解説を求む、というものである。

確かに、移動ロボットの動きだけに着目した制御の話題は多数あり、これに関する理論的な考察の歴史は長い。一方、環境認識などを含めた全体を見渡し、そのなかで制御系やロボットのシステムをいかに構築したらよいのか、という問題も、動く (働く) ロボットの実現を目指すならば極めて重要な、しっかりおさえておくべき問題である。ここでは、ハードウェア、ソフトウェア両面からの実装をどうするのかという視点もクローズアップされ、きれいな理論体系というよりは、事例の積み重ねによる経験に重きが置かれる性格が強い。

本稿では、移動ロボットに搭載された外界センサからの情報とロボットが推定する自己位置を関連付けて移動する場合を想定したシステムインテグレーションについて、サンプリング時間をキーワードにして筆者らの限られた経験を紹介したい。本稿は、いわゆる制御理論の枠組みでの話題ではなく、むしろ移動ロボットの制御系 (コントローラ) の設計上で考慮すべき内容を含む。本稿の表題はそこに指向するものとしてとらえていただければ幸いである。

2. ロボットの構成要素とサンプリング時間

ディジタル計算機でロボットを動かす以上、連続時間系ではなく離散時間系での制御となる。ロボットに搭載可能な組み込み系の計算機もかなり高速な演算が可能になった昨今、ロボットの運動時定数に対して十分早いサンプリング時間が取れ、連続時間系の制御系設計の近似的援用も可能である。電動モータの回転制御に関するサンプリング時間は、移動ロボットの場合、1 ~ 10 [ms] 程度の周期に選ばれることが多い。電動モータの出力軸にはパルスエンコーダも取り付けられ、この周期でパルスカウントを行い、モータ出力軸の回転数が計測され、回転数制御のフィードバックを行うために使われる。さらに、車輪とモータ出力軸の間に取り付けられる減速機の減速比と車輪径をパラメータとして回転数の累積計算を行えば、ロボットの自己位置が推定できる。これはオドメトリと呼ばれる。オドメトリによる推定自己位置は制御のサーボ周期ごとに得られる。

一方、外界センサとして用いられる走査型レーザ距離計 (以下では「測域センサ」と表記する) やディジタルビデオカメラを用いた視覚の外界に対するサンプリング周期は、上で述べたサーボ周期に比べれば「遅い」。例えば、北陽電機社製の測域センサの一つである URG-04LX は、公称計測周期 100 [ms] であるが、実際には周囲 240° の走査に約 67 [ms] かけ、残りの約 33 [ms] 内で、その測距データを USB あるいはシリアル通信で送り出す。一方、カメラを用いた視覚でもディジタルビデオカメラを使うことが主流になった。映像信号のアナログ伝送時代のいわゆるビデオレート (33 [ms]) の束縛からは開放されたが、それでも、ビデオでサンプリング周期 1 [ms] を実現するのは、実現例はあるにせよまだ一般的ではなく、速くても 10 [ms] 台、一般的には 20 ~ 30 [ms] 程度というところである。さらにデータ取得後の認識処理にも時間を要するから、センサが環境をセンシングしている時間だけでなく、計算機が認識結果を出すまでの間にもロボットは動いてしまう。

以上のようにサンプリング周期の相違もさることながら、ロボットに搭載される外界センサやオドメトリのサンプリングは非同期なのが一般的である。これまでのセンサは外

原稿受付 2009 年 3 月 25 日

キーワード: Mobile Robot, Navigation, Data Sharing, Sampling Period

*〒305-8573 つくば市天王台 1-1-1

*Tsukuba-shi, Ibaraki

部同期のための入出力を持たないものも多く、我々はこれらの既製品を購入するしかすべがなかった。一方で、北陽電機は早くからこの問題に気づき、同社の測域センサは計測タイミングを示す同期信号出力やセンサ内にもたせたりアルタイムクロックによるタイムスタンプを計測データに追加して送出する仕様が標準になっている [1] ~ [3]。

以下、本稿では、測域センサが出力する同期信号を利用した環境地図構築のシステム例と、多種多様なセンサが非同期に計測データを継続的に出力することを前提にしたシステム構築例を紹介する。

3. センサ同期信号を利用する計測システムと地図生成

本例のシステムは、文献 [1] [4] に紹介された、移動ロボットのハードウェア資源は、図 1 に示すように、測域センサの計測データとオドメトリデータを受け取るノート PC、エンコーダのパルスカウンタを処理する組込型計算機からなる。この組込型計算機には測域センサからの同期信号も入力される。この実装では、組込型計算機におけるエンコーダパルスのサンプリング周期は 5 [ms] であり、オドメトリによるロボットの自己位置も 5 [ms] ごとに得ることができる。

このようなシステムを構築した背景をまず説明する。ノート PC 上のプログラムにおける直感的な実装は、測域センサからの 1 走査分の計測データをすべて受信し終わった時点で、オドメトリによる自己位置を組込型計算機に問い合わせ、その位置から相対的に測域センサからの計測データを二次元平面上にプロットすることを繰り返す、というものであろう。このような実装をとると、図 2 (a) にある環境でロボットを走行させたとき、図 2 (b) にあるようなマップが作られる。実はこのケースではロボットが蛇行しており、ロボットの速度に回転角速度成分があるのである。

これを単純化して考えると、移動ロボットの並進速度 0.3 [m/s] 、半径 1 [m] の円弧上の走行で、回転角速度は 0.3 [rad/s] となる。もし、測域センサのある方向の測距時点とロボットの自己位置の計測時点との間に 30 [ms] の遅れがあると、角度方向に約 0.01 [rad] (約 0.5°) の差が出る。その測距方向の対象物までの距離が 3 [m] あるとすれば、この遅れて計測された自己位置により作られた地図では、その対象物点に約 3 [cm] の誤差を生む。図 2 (b) は、毎回の走査によるデータの取得時点とロボットの自己位置計測時点の遅れを考慮していないことにより、本来一枚の壁が重ならない。

そこで、図 1 のシステムにより、図 3 のように 1 走査時間をカバーできるオドメトリによるサンプリング時刻ごと (この実装では 5 [ms] おき) の自己位置を記憶しておき、測域センサから 1 走査終了の同期信号を受けたとき、その時点から過去のオドメトリによる自己位置をすべてノート PC 側に送る。測域センサの各角度方向の測距値が得られた時

図 1 移動測域センサシステム

(a) 計測環境

(b) 時刻同期を考慮しない二次元平面へのプロット (c) 時刻同期を考慮した二次元平面へのプロット

図 2 計測データの例 [4]

のロボット位置をこれらの自己位置から内挿することにより推定して二次元平面上にプロットすると、図 2 (c) となる。自己位置推定と測域センサの測距サンプリング時間の同期関係を考慮・計測して、位置あわせをすることの効果非常な大きいことが分かる。

図3 測域センサとオドメトリのデータ取得タイミング

4. 非同期にデータを送る多種多数のセンサに対応するシステム実装

本章では、マルチプロセス環境を提供するOSを搭載した一つのPCで、移動ロボットの動作全体を記述するモデルを扱う。本章に述べるアイデアは文献[5]～[7]等で紹介しており、文献[8]に詳細が記されている。実装例は、文献[8]のほか、文献[9][10]に紹介している。

文献[9][10]で紹介した移動ロボットは、屋外遊歩道環境で1[km]を自律走行を行う技術チャレンジである「つくばチャレンジ」に出場するために開発した。特に文献[10]では、オドメトリはもちろん、IMUとGPSを各1台、測域センサを5台など多種多様なセンサを移動ロボットに搭載した。この場合、IMUやオドメトリは5[ms]ごとにこれらに関するデータを取り込み、測域センサは、5台のうち3台の北陽電機社製UTM-30LXは1走査あたり25[ms]、2台のURG-04LXは1走査あたり100[ms]の計測周期をもつ。これらのセンサからの計測データはすべて非同期にPCに送られる。このような場合、一般にシステム設計レベルでは次の点について考慮を要する。

- (1) センサの計測時刻を毎回記録できること。前章でも記したように、複数センサのデータを融合するセンサ情報処理を行うときには、それらのセンサデータ間での観測時刻の同一性が重要である。センサごとにデータの取り込み時刻は異なるので、計測データの計測時刻を同時に記録し(タイムスタンプをつけ)、時刻間でのデータ補間ができるようにしたい。
- (2) 一つのセンサからのデータを複数の実行プロセスから読めること。例えば、測域センサのデータを使って、片や障害物検知、片やランドマーク抽出というように、異なる目的をもったプログラムを並行して走らせ、一つ

図4 SSMの構造

のセンサデータを複数の実行プロセスから読みたい。この場合、一つの実行プロセスが一つのセンサデバイスを占有しては困るので、このデータを複数の実行プロセスで共有できるようにしたい。

- (3) センサのデータ送出周期が短くてもセンサデータを取りこぼさないこと。センサデータを読み込む実行プロセスの計算量が多いと、OSのコンテキストスイッチの周期が遅くなり、データ送出周期の短いセンサデータを取りこぼす懸念がある。このためできるだけ軽い実行プロセスでセンサデータを読み、これをバッファにためてデータを取りこぼさない(取りこぼしにくい)ようにしたい。

これらの要請を満足する一つの方法として、共有メモリによる方法を筆者らは採用している[8](図4)。ここでは、センサごとの通信チャンネル(USB, シリアル, TCP/IPなど)ごとにこのデバイスを占有し、センサデータの一つの計測単位ごとにタイムスタンプをつけるSensor Handler[†]、これらのセンサデータを一定時間記録するリングバッファ形式の共有メモリ、指定時刻付近のセンサデータを高速に取り出すための時刻テーブルを含むフィルタリングプロセスな

[†]文献[5][6]などでは、当初、Sensor HandlerはSensor Driverと名づけていた。

どからなる．共有メモリへのデータ入出力をつかさどるプロセス群全体を SSM (Sensor Sharing Manager)†と名づけた．実装上，共有メモリへの書き込みは Sensor Handler のみで書き込みの競合はない．Sensor Handler は計測データの取り込みと共有メモリへの書き込みだけの非常に軽い実行プロセスである．また，共有メモリからの読み込みは現在の最新データまでを読み，書き込み中のデータは読み込まないよう工夫している．SSM には，共有メモリへのデータアクセスに時刻指定ができるライブラリを用意したので，あるセンサ A の計測データの計測時刻を別のセンサ B の計測データを取得するときに指定することにより，その時刻近傍のセンサ B のデータを読むことができ，これにより線形補間を行って，センサ A の時刻におけるセンサ B の計測データを推定できる．このようにしても，図 2(c) に得られたのと同様な結果を得ている [8]．ちなみに文献 [10] では，移動ロボットのナビゲーションに直接関係するプロセスだけで，Sensor Handler プロセスは 6 個前後，SSM からデータを利用するプロセスを 10 個前後を Panasonic 社製 Tough-Book (CPU: Intel(R) Core(TM)2 Duo CPU L7500, 1.60 [GHz] 主記憶 2 [GB]) 上の Linux OS で稼働させた．またこの例では測域センサからの情報に付加されたタイムスタンプを利用し，測域センサの同期信号は利用していない．

逐次更新・記録されるセンサデータを共有するアイデア自体は，文献 [11] などを一例として 1980 年代後半から 1990 年代前半に多数のアイデアが出た．文献 [11] は自動車を改造した例であり，当時は移動体自体が大掛かりな例が多かった．現在の計算機能力の一層の向上・小型化，測域センサの小型軽量化が，小規模な移動ロボットでもこのような実装が可能となったもいえる．

5. お わ り に

以上，移動ロボットを，障害物回避や目標経路に沿った動きをさせる際のセンサ系のサンプリング周期の相違や多種多数のセンサを用いる場合の実装に関して，筆者らの限られた経験を紹介した．

参 考 文 献

- [1] H. Kawata, W. Santosh, T. Mori, A. Ohya and S. Yuta: "Development of ultra-small lightweight optical range sensor system," Proceedings 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.3277-3282, 2005.
- [2] 測域センサ URG-04LX 製品仕様書．北陽電機株式会社，2005.9.9 版．
- [3] SCIP2.0 準拠 "URG" シリーズ 通信仕様書．北陽電機株式会社，2006.10.10 版．
- [4] 上田達朗，川田浩彦，富沢哲雄，大矢晃久，油田信一："移動測域センサ—走行中でも正しい計測が可能な移動ロボット用測域センサシステム—"，計測自動制御学会第 6 回システムインテグレーション部門講演会 (SI2005) 予稿集，pp.574-575, 2005.
- [5] E. Takeuchi, T. Tsubouchi and S. Yuta: "Integration and Synchronization of External Sensor Data for a Mobile Robot," Proceedings of SICE Annual Conference 2003, pp.1019-1020, 2003.
- [6] 竹内栄二朗，坪内孝司，油田信一："センサ情報を統合する機能を有する移動ロボットソフトウェアシステム"，第 4 回 SICE システムインテグレーション部門講演会 (SI2003)，1A4-3 (CD-ROM)，2003.
- [7] E. Takeuchi and T. Tsubouchi: "Sensory Data Processing Middleware for Service Mobile Robot Applications," SICE-ICASE International Joint Conference 2006, pp.1201-1206, 2006.
- [8] 竹内栄二朗：移動ロボットの基本機能のモジュール化と環境地図生成に関する研究，筑波大学大学院システム情報工学研究科 博士論文 第 2 章および第 6 章，2008.
- [9] 坪内孝司，Yoichi Morales，徳永航，竹内栄二朗，他："筑波大学知能ロボット研究室『屋外組』における屋外走行ロボットのシステムインテグレーション"，第 8 回 SICE システムインテグレーション部門講演会 (SI2007)，3C3-6 (CD-ROM)，2007.
- [10] 坪内孝司，Yoichi MORALES，Alexander CARBALLO，原祥亮，他："つくばチャレンジ 2008 における筑波大学知能ロボット研究室『屋外組』の取組み"，第 9 回 SICE システムインテグレーション部門講演会 (SI2008)，114-6 (CD-ROM)，2008.
- [11] S. Shafer, A. Stentz and C. Thorpe: "An architecture for sensor fusion in a mobile robot," Proceedings of 1986 IEEE International Conference on Robotics and Automation, Volume 3, pp.2002-2011, 1986.
- [12] 竹内栄二朗，坪内孝司："異種複数センサに対応した移動ロボット用ナビゲーションソフトウェアの公開"，ロボティクス・メカトロニクス講演会 '08，1P1-E22 (CD-ROM)，2008.

坪内孝司 (Takashi Tsubouchi)

現在，筑波大学大学院システム情報工学研究科知能機能システム専攻教授．移動ロボット技術をベースとした屋内外で実用に供されるサービスロボットに関する研究に従事．IEEE member，日本機械学会，人工知能学会などの正会員． (日本ロボット学会正会員)

†SSM のソフトウェア公開については文献 [12]，<http://sourceforge.jp/projects/pc-ymbc/>，<http://sourceforge.jp/projects/pc-ymbc/wiki/FrontPage> 参照．