自律移動ロボットシミュレータの動作プログラム開発支援機能の拡張

○小坂 賢範 (岡山大学)　永谷 圭司 (岡山大学)　五福 明夫 (岡山大学)

Expansion of Programming Support Function for Mobile Robots' Simulator

*Masanori Kosaka (University of okayama), Keiji Nagatani (University of okayama), Akio Gofuku(University of okayama)

Abstract— To develop motion programs for multiple autonomous mobile robots, programming simulator has developed in our laboratory. By using this simulator, a cost (i.e. debugging time for programming) can be reduced. In this research, we add several functions on the simulator, such as a distance sensor simulation, display of robot's locus, and analysis of positioning error for mobile robot. In this paper, we describe details of these functions, and report application examples of this simulator.

Key Words: Simulator, Distance sensor, Robot's locus, Positioning error, Monte Carlo method

1. はじめに

移動ロボットの研究を行う際，実機による実験は，実験のセッティングに時間がかかるなど，コストが大きい．また，ロボットの故障によって，実験自体ができなくなる可能性もある．このような問題に対し，移動ロボットの動作プログラムの動作検証をシミュレーション上で行うことが提案されてきた [1)2)]．

筆者らの研究グループにおいても，これまでに移動ロボットの動作プログラムの動作検証を行うことができるシミュレータの開発を行ってきた [3)]．このシミュレータの導入によって，動作プログラムのバグをシミュレータ上で発見できるようになり，実機の暴走によって周囲の人や物に危害がおよぶ危険性を回避できるようになった．また，動作プログラムの開発が，ロボットの状態に依存することなく行えることから，研究の進度向上にも貢献した．

本研究では，このシミュレータの機能拡張を行い，より多くのタスクに対応できるように改良した．本稿では，このシミュレータの機能拡張である，距離センサのシミュレーション，ロボットの軌跡の表示，およびロボットの衝突危険性のシミュレーションについて述べ，その機能の有効性を報告する．

2. シミュレータの概要

2·1 対象とするロボットのシステム構成

本研究室では，複数台の自律移動ロボットを所有している (Fig.1)．これらのロボットはすべて,「System Program」と「User Program」という 2 つのプロセスによって制御されている．このうち,「System Program」はロボットの制御を行うプログラムであり,「User Program」は各ユーザが作成する動作プログラムである．これら 2 つのプロセスが共有メモリを用いたプロセス間通信を行うことによって，ロボットの制御を行っている (Fig.2)．

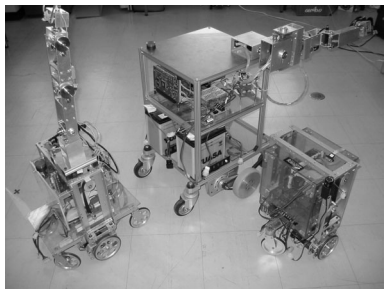

Fig.1 Autonomous mobile robots

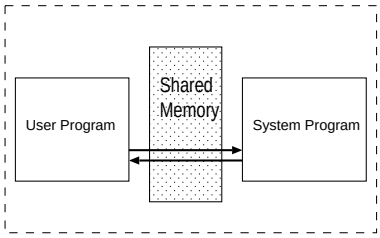

Fig.2 Each robot's system configuration

2·2 シミュレータのシステム構成

本研究室で開発中のシミュレータは，実機制御を行う「System Program」のシミュレーションを行うものである．これにより，シミュレーションで動作検証を行った「User Program」は，そのまま実機に搭載可能である．

本シミュレータは，先の 2 つのプログラム以外に「環境模擬プログラム」を有する．この「環境模擬プログラム」は，Gtk+および OpenGL を用いて作られており，3 次元でロボットや環境のモデルを表示することができる (Fig.3)．これらのプロセスと共有メモリによってシミュレータを構成し (Fig.4)，複数台の自律移動ロボットの同時シミュレーションを可能とした．

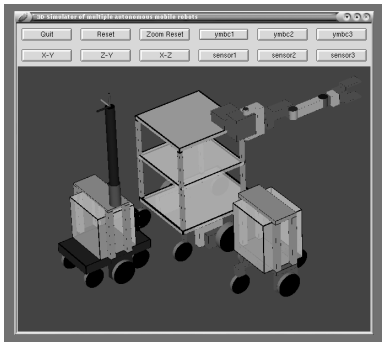

Fig.3 Display of simulator

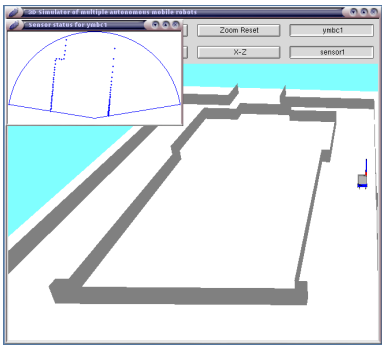

Fig.6 PB9 simulation

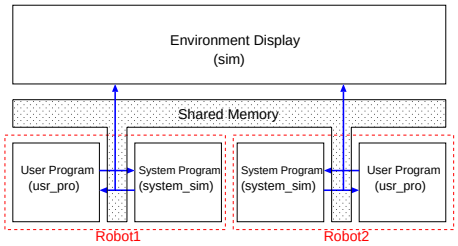

Fig.4 Simulator's configuration

3. 距離センサのシミュレーション

3·1 距離センサのシミュレーション機能の構成

本研究室のロボットには，北陽電機社製の「測距式障害物検出センサ PB9-01」(以下，PB9) という距離センサを実装している (Fig.5)．この PB9 は，LED 光線により半円状のフィールドを 91 ステップ (162 °) でスキャンし，エリア内の障害物までの距離を計測することができる．本研究では，このセンサをシミュレータ内でも利用できるように，本シミュレータに PB9 のシミュレーション機能の追加を行った．

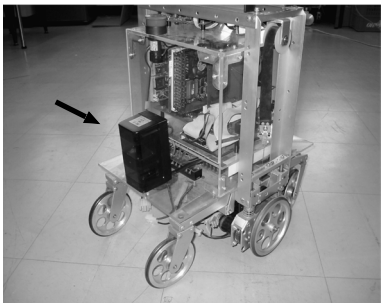

Fig.5 A robot with PB9

本シミュレータ内における環境は，ベクターマップによってユーザが設定することができる．このベクターマップのデータとシミュレータ上のロボットの位置，姿勢を基に，センサ情報を作成することができる．本シミュレータでは，このセンサ情報を，Fig.6 に示すように視覚的に確認できるものとした．

3·2 シミュレーション例 1：長距離ナビゲーション

距離センサシミュレーション機能を用いた本シミュレータの使用例を以下に示す．

本年度，筆者らの研究室では，ロボットの動作プログラミングを習得する目的で，「幅約 2m，一周約 60m の廊下を，PB9 を利用し，壁に接触することなく自律走行させる」という課題が 4 年生に与えられた．4 年生は，研究室に配属されたばかりであり，動作プログラムのバグによってロボットを暴走させる可能性が高いと考えられた．そのため，このタスクを実現する動作プログラムの開発をシミュレータ上で行った．

4 年生らは，シミュレータを利用し，試行錯誤を繰り返してプログラム開発を行った結果，上記のタスクをシミュレータ上で実現することができた．また，シミュレータで動作検証を行ったプログラムを実機に搭載し，実際の環境でロボットの走行実験を行ったところ，ロボットは，壁に接触することなく 60m の距離を移動することができた．

以上のことから，PB9 を使った動作プログラムを，実機を使うことなく開発できることが確かめられ，シミュレータに追加した機能の有効性を確認することができた．

4. 軌跡の表示

4·1 軌跡表示機能の構成

本研究室のシミュレータは，3 次元でロボットや環境のモデルを表示できる．このため，このシミュレータ中にロボットの走行軌跡，およびロボット搭載のマニピュレータの手先の軌跡を残すことで，プログラムの動作の経過確認を行うことができる．本研究では，ロボットの走行軌跡，およびロボット搭載のマニピュレータの手先の軌跡を残す機能を，シミュレータに追加した．

4·2 シミュレーション例 2：移動マニピュレータによる直線描き動作

軌跡表示機能を用いた本シミュレータの使用例を以下に示す．本研究室で行われている研究の一つに，「可操作度を考慮した移動マニピュレータの動作計画と動作の実現」[4] がある．この研究は，マニピュレータの手先に固定したペンを用いて，垂直な壁面に斜めの直線を描くというタスクを設定し，User program の開発を行っている．この User program のシミュレータによる動作検証の際に，この軌跡表示機能を使用した．

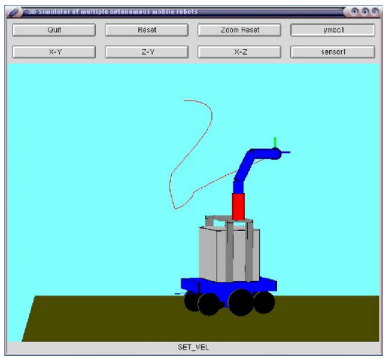

Fig.7 Display of trace line

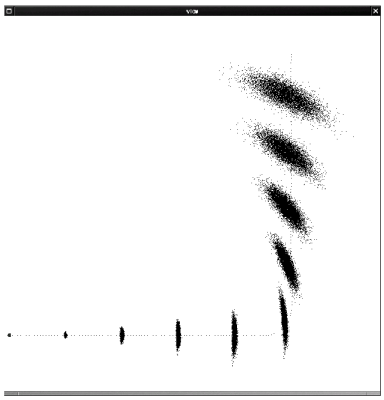

Fig.8 Error expression using Monte Carlo method

動作検証の結果，直線を描くというタスクは，描き始めに多少曲線になることを除けば，ほぼ達成されていることが確かめられた (Fig.7)．しかし，手先の軌跡が壁に追従できていないことが，シミュレーションにより確認された．これにより，実ロボットを動作させることなく，動作プログラムのデバッグを進めることができ，軌跡表示機能の有効性が確認された．

5. 衝突危険性のシミュレーション

5·1 衝突危険性のシミュレーションの目的

本研究室のロボットは，オドメトリを用いた自己位置推定を行っている．しかし，オドメトリによる推定位置は，タイヤの滑りや推定に用いるパラメータの誤差などで，実際の位置，姿勢との間に誤差が生じる．この誤差によって，狭路を通過させるようなタスクの場合，実機で壁に衝突してしまうということが起こる．このような場合，シミュレータ上で予め壁に衝突する危険性がどれだけあるかが分かれば，経路の変更や外界センサによる自己位置修正によって，走行安全性を向上させることが可能となる．

本研究では，シミュレータ内で自己推定位置の不確実性の可視化を行い，これをシミュレータ上に表示できるようにした．また，衝突の危険性の指標として，ロボットが動き始めてから現在の地点までの生存確率 (障害物に衝突しない確率) をシミュレータ上に表示できるようにした．

5·2 推定位置誤差表現

本研究では，推定位置誤差の表現を行うために，モンテカルロ法を採用した [5)6)]．このモンテカルロ法による自己推定位置の表現方法は，まずロボットモデルの車輪径とトレッドに正規分布の誤差を持たせ，ロボットの走行シミュレーションを複数回繰り返す．次に，各走行毎にロボットの位置を画面上にプロットする．これにより，複数回の走行が終わったときの自己位置のばらつきを見ることで，ロボットの推定位置誤差を表現する手法である．ある条件の下で，移動ロボットがL 字の走行経路を走行した場合のモンテカルロ法を用いた誤差表現は，Fig.8 のようになる．ただし，ここでの試走回数は，10000 回とした．

5·3 生存確率の計算

ロボットが環境に近付くと，ロボットの誤差が大きい場合，衝突の可能性が出てくる．このことを表現するために，ロボットの複数回の試走結果のうち，壁からの距離が一定値以内に進入したものについては，壁に衝突したと見なすこととした．また，この衝突したものについては，以降の試走を行わないこととした．これにより，全試走回数 10000 回のうち，壁に衝突しない試走の割合が r であるとき，この r を環境と接触せずに現在地点まで到達することのできる確率として，シミュレータ上に表示させるようにした．

一方，実機では，外界センサを使って推定位置の修正を行い，自己位置の不確実性を低減させることができる．そのため，シミュレータでも，このセンシングによる自己位置修正により，自己位置のばらつきを一定範囲内に収めることとした．

5·4 シミュレーション例 3：狭路の通り抜け動作

衝突危険性のシミュレーション例として,「研究室内から，研究室の外に通じるドアを通り，研究室の外の廊下まで走行する」という動作を設定した．環境および経路は Fig.9 に示す通りである．この動作を，シミュレータと実機によって行い，その結果を比較した．また，Fig.9 の A 地点でセンシングを行う場合と行わない場合で，ロボットの生存確率がどのように変化するかを検討した．

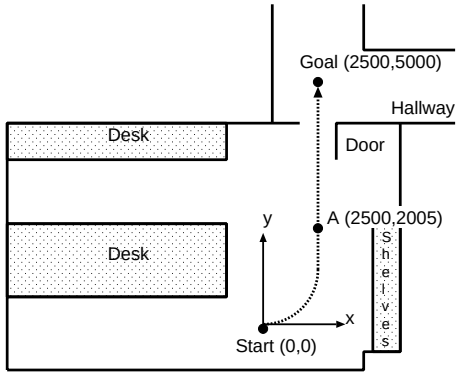

Fig.9 Environment in laboratory

実験に使ったロボットは，本研究室所有の車輪型自律移動ロボット「NOBU」(Fig.1 右端) である.「NOBU」は，CCD カメラと日立製作所の画像処理ボード IP5000

を搭載しており，天井画像を用いたテンプレートマッチングによって自己位置修正を行うことができる．

Fig.9 に示す環境をシミュレータ内に設定し，センシングを行う場合と行わない場合で生存確率の比較を行った．その結果，センシングを行わない場合の生存確率は約 73 %，センシングを行った場合の生存確率は約 99 %であった．

実機を用いた実験では，天井画像を用いた位置修正を行うこととした．具体的には，A 地点の天井画像を予めテンプレートとして登録しておき，このテンプレートと CCD カメラの映像とのマッチングを行うことで，自己位置修正を行う．

この，センシングを行う場合と行わない場合での，目標経路の走行を 30 回ずつ行った．その結果，どちらの場合も障害物に衝突することはなかった．しかしながら，ゴール到達時のロボットの位置を計測した結果，センシングなしの場合は Fig.10，センシングありの場合は Fig.12 のようになった．

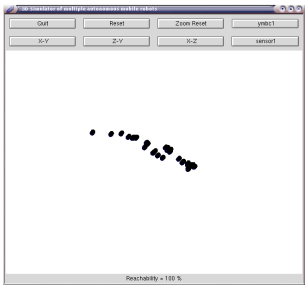

Fig.10 Variance of actual robot position without sensing

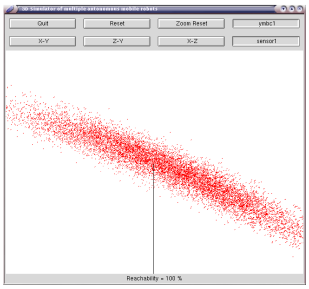

Fig.11 Variance of simulated position without sensing

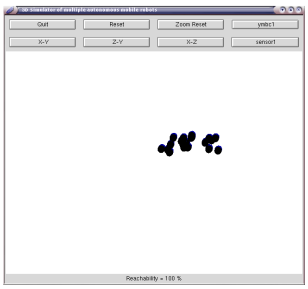

Fig.12 Variance of actual robot position with sensing

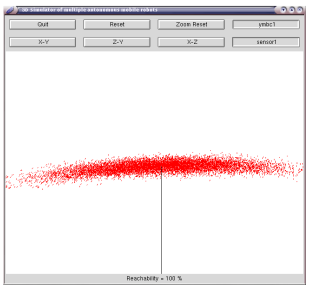

Fig.13 Variance of simulated position with sensing

なお，環境との衝突が起こらないと仮定してシミュレーションを行った場合，自己位置のばらつきは，Fig.11，Fig.13 となった．Fig.10 と Fig.11，Fig.12 と Fig.13 の比較を行うと，多少のずれはあるものの，Fig.10 は Fig.11 を縮小したもの，Fig.12 は Fig.13 を縮小したものとなっている．

以上より，シミュレーションでは，初期誤差やロボットのモデル誤差を大きめに設定したため，環境との衝突が起こる可能性が高くなり，生存確率が下がったが，誤差のモデル自体は合致していたと考えられる．よって，車輪径およびトレッドの誤差に関するパラメータチューニングを行えば，より実機に近いシミュレーションを行うことが可能となる．これによりこの機能は，衝突確率の改善による動作プログラムの信頼性向上に利用できると期待できる．

6. まとめと今後の展開

本稿の機能追加によって，シミュレータは，様々なタスクに対応できるようになった．

PB9 のシミュレーションでは，センサを利用したタスクの動作プログラム開発が PC 上で行えることから，プログラム開発中に，ロボットの暴走や環境との衝突といったリスクを低減させることができた．また環境の設定が自由に行えることから，複雑な環境下でタスクを行うような場合の動作プログラム作成にも有効であると考えられる．また，移動ロボットのオドメトリに生じる誤差をシミュレーションすることで，定量的にゴールまでの生存確率や経路中の危険な部分を探すことができるようになった．

今後は，これらの機能をさらに発展，および必要とされる機能の追加を行っていくことで，動作プログラム環境の更なる改善を行っていく予定である．

参考文献

1) 鈴木昭二，木元克美，油田信一：移動ロボットの自律行動のためのプログラム開発環境の構築，日本ロボット学会誌 Vol.12 No.3, pp.497 ～ 506, (1994)
2) 木元克美，油田信一：移動ロボットのための超音波センサシミュレータ，日本ロボット学会誌 Vol.13 No.2, pp.297 ～ 304, (1995)
3) 木塚貴登，永谷圭司，五福明夫，田中豊：複数台自律移動ロボット研究のための動作プログラムシミュレータの開発，ロボティクス・メカトロニクス講演会，(2002)
4) 平山智信，永谷圭司，五福明夫：可操作度を考慮した移動マニピュレータの動作計画，第 19 回日本ロボット学会学術講演会予稿集，(2001)
5) Masahiro Irie, Keiji Nagatani, Akio Gofuku: Path Evaluation for a Mobile Robot Based on a Risk of Collision, IEEE International Symposium on Computational Intelligence in Robotics and Automation, (2003)
6) Dieter Fox, Wolfram Burgard, Frank Dellaert, Sebastian Thrun: Monte Carlo Localization: Efficient Position Estimation for Mobile Robots, in Proc. of the Sixteenth National Conference on Artificial Intelligence (AAAI-99), Orlando, Florida, (1999)