

自律移動ロボットにおける処理時間と推定精度を考慮した モンテカルロ位置推定法

Monte Carlo Localization for Autonomous Mobile Robots Considering Processing Time and Estimation Accuracy

○石原 悠, 鈴木 崇文 (慶應義塾大学大学院)

高橋 正樹 (慶應義塾大学)

* Yu ISHIHARA, Takafumi SUZUKI (Keio Univ.),

Masaki TAKAHASHI (Keio Univ.)

ishihara@yt.sd.keio.ac.jp, suzuki@yt.sd.keio.ac.jp, takahashi@sd.keio.ac.jp

Abstract—Monte Carlo Localization (MCL) is typically used for self localization of autonomous mobile robots approximating its probabilistic distribution by particles. Processing time of MCL is mainly influenced by the number of particles but it depends on the size of the space particles are distributed. Therefore adaptive adjustment of particles is effective to control the processing time. In this paper, adaptive adjustment of particles for localization is designed. In addition, to reduce the processing time, the effective resolution to scan the white lines is studied. To verify the practical effectiveness of the proposed methods, simulation and experiment were carried out applying to the RoboCup Soccer Middle-size League.

1. 緒言

Monte Carlo Localization (以下 MCL)はパーティクルと呼ばれる粒子を用い確率分布を近似することで位置推定を行う手法である。本手法は非線形性を有するロボットの移動を扱え、初期位置を予め与える必要がなく、導入も容易なため、近年ロボットの位置推定に広く用いられている[1][2][4]-[8][10][11]。MCLにおいて位置推定に必要な計算量は主に前述のパーティクルの数によって左右され数が多いほど時間がかかる。また、パーティクルは推定している地点の尤度によって分布が拡大、縮小するが、位置推定に必要なパーティクル数はパーティクルの分布する空間の大きさで決まってくる事が知られている[3][11]。したがって、パーティクル数を適応的に調整することが可能であれば計算資源を節約し、ロボットの行動に必要な時間の中で画像処理等、計算を多く必要とする処理に時間を割り当てることが可能になると考えられる。本研究が実証対象とするRoboCup サッカー中型リーグにおいても、白色ゴールへの変更やボールに対する特定色の廃止等、近年物体に対し特定の色を割り当てないようにルールが改正されてきており、画像処理による計算時間の増加が見込まれ、計算資源の節約はロボットが実時間で行動するために重要と考えられる。

また、パーティクルフィルタはその性質から、主に大域的自己位置推定において推定に用いるパーティクル数が少ない場合、パーティクルの収束は早い

ものの真値周辺からパーティクルが消失してしまうことがしばしばある。特に通路や部屋、そしてRoboCup サッカー中型リーグで用いられる対称なフィールドのように、ロボットの位置と似通ったマップ形状の地点が存在する環境においてこれは顕著であり、このことは、不要なリセット[9]を増やし初期位置推定や誘拐状態からの復帰を遅らせる要因となる。したがって、大域的自己位置推定ではパーティクル数を増加させて推定を行い、高い確率で正しい位置を算出することが望ましい。しかし、その場合にはパーティクル数の増加に伴う初期位置算出までの時間の増加との両立を考える必要がある。

さらに、RoboCup サッカー中型リーグではこれまで、主にフィールドに引かれている白線までの距離を放射状に探索して自己位置推定を行うもの[4]-[8][10]が多かったが、これらは白線探索時の角度分解能について細かく言及していない。しかし、この分解能が細かいほど MCL の処理時間に影響を与えるため、計算資源の節約には適当な角度分解能を知る必要がある。また、このような知見は RoboCup サッカー中型リーグで用いられる全方位カメラに限らず、レーザーレンジファインダ等の放射状に探索するセンサを用いて行う自己位置推定に広く利用できることが期待される。

以上より、自己位置推定の精度を維持しながら、計算時間を短縮できれば、ロボットをより効率的に動作させられると考えられる。この際、白線探索の角度分解能は、より多くの情報を得るために細かくとることが望ましいと考えられる。

そこで本研究ではRoboCup サッカー中型リーグを題材とし、MCLにおける局所的自己位置推定においてパーティクル数の適応的調整による自己位置推定に必要な計算時間の減少および精度の維持、MCLにおける処理時間低減のための白線探索時の角度分解能についての検討を行う。また、実機実験により提案手法の有効性を確認する。

2. パーティクル数の適応的調整法

パーティクル数を適応的に調整する手法は、これまでに複数提案されている。Heinemannら[10]はパー

ティクル数を位置推定の精度に応じて変化させる方法を提案し、計算時間を減少させることに成功している。Kwok ら[11]はカルバックライブラー情報量に基づいてパーティクル数を決定する Kullback-Leibler divergence(KLD)サンプリング[3]による自己位置推定を提案している。KLD サンプリングでは真の確率分布に対するパーティクルによる近似誤差が ε 以下になる確率が $1-\delta$ になるように以下の式を用いて各ループでのパーティクル数 n が決定される[12].

$$n = \frac{k-1}{2\varepsilon} \left\{ 1 - \frac{2}{9(k-1)} + \sqrt{\frac{2}{9(k-1)} z_{1-\delta}} \right\} \quad (1)$$

ここで、 k はロボットの存在する空間を複数に分割した(分割した空間1つ1つをビンと呼ぶ)ときにパーティクルが1つ以上入っているビンの数であり、 $z_{1-\delta}$ は正規標準偏差の上側 $1-\delta$ 分位数である。したがって、KLD サンプリングはパーティクルが分布する空間の広さに応じてパーティクル数を調整するアルゴリズムである。Kwok らは直接計算時間について言及していないが、パーティクル数の減少とともに計算時間も減少すると考えられる。

本研究では、局所的自己位置推定時において発生するパーティクルの分布拡大、縮小に対しパーティクル数を調整する必要があると考え、KLD サンプリングによりパーティクル数を決定する。

3. 自己位置推定手法

KLD サンプリングによってパーティクル数の調整を行う白線を用いた局所的自己位置推定法について述べる。

3.1 環境モデル

本研究では RoboCup サッカー中型リーグを対象とする。Fig. 1にフィールドを示す。フィールドは緑の地面の上に白線が引かれている環境である。

観測情報としては一般的に用いられるこの白線を利用する。フィールドを $0.1\text{m} \times 0.1\text{m}$ のグリッドに分割し、各グリッドから白線までの各方向における相対距離を事前に環境モデルとしてロボットに与えておく。このときの角度の刻みは白線探索の角度分解能 α deg とし、環境モデルは $m_i(x, y, \theta)$ 、($i = 0, \alpha, 2\alpha, \dots, 360$) とする。

全方位カメラによる画像の中心から外側に向けて放射状に白線を探索し、角度 α ごとのロボットから見た最近傍の白点を抽出する。そして実際に観測された各方向までの相対距離を観測情報 d_i とする。

3.2 移動モデル

パーティクルをサンプリング周期毎に遷移させるための移動モデルは、デッドレコニング情報として得られる移動量 $(dx_t, dy_t, d\theta_t)$ に不確かさ $(\gamma_x, \gamma_y, \gamma_\theta)$ を加えたものとして表される。

$$\begin{aligned} x_t^j &= x_{t-1}^j + dx_t + \gamma_x \\ y_t^j &= y_{t-1}^j + dy_t + \gamma_y \\ \theta_t^j &= \theta_{t-1}^j + d\theta_t + \gamma_\theta \end{aligned} \quad (2)$$

ここで、不確かさ γ_x および γ_y は 0m から $\pm 0.1\text{m}$ の範囲、 γ_θ は $\pm 2.5\text{deg}$ の範囲でランダムに与えた。また、 $(x_t^j, y_t^j, \theta_t^j)$ 、($j = 0, 1, \dots, n$) は時刻 t におけるパーティクルの位置と向きである。

さらに、角度 θ_t^j は方位センサを用いて取りうる値の範囲を計測方位から $\pm 45\text{deg}$ の範囲に限定する。

3.3 パーティクルの重み

時刻 t における j 番目のパーティクル p_t^j の重み w_t^j は環境モデルと観測情報より計算する。

まず以下の式を用いてモデルと観測値の誤差 e を計算する。

$$e(x_t^j, y_t^j, \theta_t^j) = \sum_i |m_i(x_t^j, y_t^j, \theta_t^j) - d_i| \quad (3)$$

この誤差 e を用いてパーティクル s_t^j の尤度 l_t^j を次のように計算する。

$$l_t^j = \frac{1}{1 + e^4} \quad (4)$$

最後にパーティクルの重み w_t^j は尤度 l_t^j より次式を用いて計算する。

$$w_t^j = \frac{l_t^j}{\sum_{k=1}^n l_t^k} \quad (5)$$

3.4 リサンプリング

リサンプリングは前節で求めたパーティクルの重みに応じて確率的にパーティクルを選択することにより行う。この際選択されたパーティクルが新たなビンに入るたび、前述の式(1)より、パーティクル数の下限 n を計算する。 $k=1$ の場合は下限を考慮せず選択する。今回ビンの大きさ Δ は $0.1\text{m} \times 0.1\text{m} \times 5\text{deg}$ とし、 $\varepsilon=0.7$ 、 $\delta=0.99$ とした。パーティクルの選択はこの下限を満たすか、または予め設定した上限 M に達するまで行う。本研究では $M=200$ とした。

3.5 自己位置の決定

各ロボットはループ毎に全パーティクルの位置および向きの重み付き平均をそれぞれ計算し、その計算結果を自己位置として利用する。

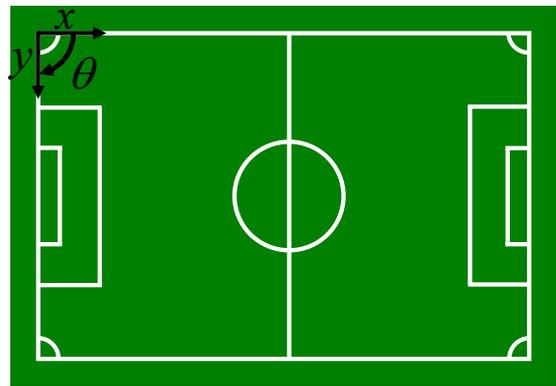


Fig. 1 Field of RoboCup Middle Size League

4. 実験

4.1 実験条件

提案手法の有効性を検証するために、以下の実験を行った。実験フィールドサイズは $18\text{m}\times 12\text{m}$ のフィールドの中で、Fig. 2に示す斜線部の領域であるが、ロボットに与える環境モデルは通常の $18\text{m}\times 12\text{m}$ のフィールドである。また、ロボットは全方位移動機構により移動し、以下のいずれの実験も特に断りのない限りFig. 2に示した地点に最初配置する。全方位カメラのフレームレートは 30fps とする。

また、各制御周期の x 方向、 y 方向の移動量は全方位移動機構に取り付けられた4つのホイールの回転量をエンコーダを用いて測定することで計算し、回転角は角速度センサの測定値を積分することにより算出する。ロボットに搭載した方位センサは8方位を識別可能である。

さらに今回自己位置推定に要求する精度は使用するロボットサイズ $0.5\text{m}\times 0.5\text{m}$ (高さは含まない) に対し、推定値の平均値が x 方向、 y 方向に対しそれぞれ $\pm 0.5\text{m}$ 以内、姿勢角度 θ 方向に対し $\pm 20\text{deg}$ 以内とする。また、要求する計算時間はRoboCupサッカー中型リーグにおいて、これまで EIGEN Keio Univ.[13] のロボットの制御周期が 30ms から 40ms であるため、センシングから自己位置推定までで 40ms 以下とする。今回計算にはCPUとしてPentium M 1.1GHzを搭載したLaptop型PC、Panasonic製Let's note R3を用いた。

4.2 白線探索の角度分解能が推定精度に与える影響の検証

白線探索の角度分解能が推定精度に与える影響を検証するために、Fig. 2で示した地点にロボットを配置し、初期値を与えた場合の自己位置推定結果を記録した。ここで角度分解能は 2deg , 3deg , 4deg , 6deg , 12deg , 36deg の6パターンを用いた。また、パーティクル数の適応的調整は行わず $N=200$ で固定し、全パーティクルに初期値として $x=1.7\text{m}$, $y=3.3\text{m}$, $\theta=90\text{deg}$, $w_0=1/200$ を与えた。測定結果は初期位置を与えた影響を低減するため、測定開始100ループ経過後500ループの推定値の平均値として算出した。結果をTable 1に示す。

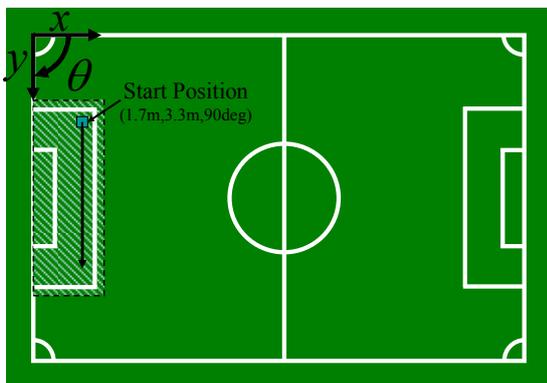


Fig. 2 Experiment environment

Table 1より角度分解能が 36deg の場合においてのみ、 x 軸方向に要求精度を逸脱する推定誤差が見られる。一方で、他の角度分解能における x 軸、 y 軸方向の推定誤差は要求精度を満たすことが判った。また、 θ 方向についてはいずれの分解能でも要求精度を満たす推定結果であった。したがって、今回の検証の範囲では、 12deg より細かい角度分解能で白線探索を行えば要求精度を満たした自己位置推定結果が得られると言える。

4.3 パーティクル数の適応的調整が推定精度に与える影響の検証

パーティクル数の適応的調整が推定精度に与える影響を検証するため、前節と同様の方法でFig. 2に示した初期位置におけるロボットの自己位置推定結果を記録した。適応的調整を行うにあたり、初期パーティクル数は $N=200$ とした。推定結果をTable 2に示す。

Table 2より、パーティクル数を固定した場合と同様に角度分解能が 36deg の場合において、 x 軸方向に要求精度を逸脱する推定誤差が見られるものの、他の角度分解能では、 x 軸、 y 軸方向および θ 方向の推定誤差は要求精度を満たしている。

したがって、角度分解能を 12deg より細かくしパーティクル数の適応的調整を行った場合の自己位置推定はパーティクル数固定の場合と同様に要求精度を満たすことが判った。また以上の結果から、パーティクル数の適応的調整は推定精度に大きな影響を与えないと言える。

4.4 白線探索の角度分解能およびパーティクル数の適応的調整による計算時間の検証

4.2節、4.3節の実験において、白線探索の角度分解能およびパーティクル数の適応的調整に伴って減少する計算時間を検証するため、それぞれの実験においてセンシングから自己位置推定までの計算時間を計測した。Fig. 3にパーティクル数を200で固定した場合およびパーティクル数を適応的に調整した場合における各角度分解能に対する計算時間を示す。

Fig. 3からパーティクル数を固定した場合、適応的に調整した場合の双方において、分解能の減少と共に計算時間も減少しているのがわかる。したがって、前節までの結果をふまえると角度分解能が 2deg から 12deg の範囲における自己位置推定は要求精度を満たすため、今回は考慮していないオクルージョン等による白線の隠れを無視すれば、局所的自己位置推定においては白線探索の角度分解能を必要に応じて粗くし計算時間を減少させても問題ないと考えられる。また、パーティクル数固定の場合は 12deg 、適応的調整を行った場合は 6deg より粗い角度分解能において、計算時間が 33ms と一定になることがわかった。これは今回用いたプログラムが毎周期カメラの画像を必ず取得するようになっていたため、自己位置推定に必要な計算時間が今回用いた全方位カメラのフレームレートである 30fps 以下となり、カメラの画像取得スピードを上回ったためと考えられる。

4.5 ロボット移動時の推定精度の検証

ロボット移動時においても提案手法が正しく自己位置推定を行えるか確認するため、ロボットをFig. 2に示す $y=1.7\text{m}$ の直線上を図の矢印の方向に約 0.7m/s で移動させ、パーティクル数固定(従来手法)の場合と適応的に調整した場合とで実験を行った。ロボットの移動は無線を用いて人間による外部からの遠隔操作により行う。本検証においてロボット操作時に角度を維持するのが難しいという観点から角度に関しては検証を行わない。また、白線探索の角度分解能は、 12deg では前節で得られた画像取得スピードと計算時間の関係からパーティクル数を適応的に調整した場合と固定した場合とで計算時間の比較が難しいため 6deg を用いた。

Fig. 4とFig. 5にパーティクル数を固定した場合および適応的に調整した場合のロボットの軌跡と推定結果、Fig. 6に処理時間、Fig. 7にパーティクル数の時刻歴をそれぞれ示す。ここで、Fig. 6、Fig. 7において移動開始時を0番目のループとしている。

Fig. 4とFig. 5から従来手法、提案手法共に推定誤差は要求精度内におさまっている。このときのセンシングから位置推定までの計算時間の平均は従来手法が 45ms 、提案手法が 39ms であった。したがって、提案手法は求める制御時間内に計算時間を短縮しながら要求精度を満たした推定が行えたといえる。ここで、提案手法ではロボット静止時より計算時間が増加しているが、これは移動に伴う画像の乱れによって白線の観測が不十分になったことや全方位移動機構のすべりの影響等によりパーティクルの分布が拡大しパーティクル数が増加しやすかったためと考えられる。

さらに、Fig. 6からパーティクル数を固定した場合の推定では頻繁に計算時間が 50ms 以上に増加しているが、適応的調整を施した場合においてはほとんどの場合において計算時間は 50ms 以下に収まっている。したがって、各ループにおける計算時間という観点からも提案手法は有効に動作すると言える。

4.6 少ないパーティクル数による推定とパーティクル数の適応的調整による推定の精度の検証

計算時間を考慮する場合、パーティクル数を適応的に調整する代わりに、少ないパーティクル数で固定して推定を行うことも一つの手段として考えられる。そこで、本節ではパーティクル数を減らした場合における影響を検証するため、Fig. 8に示す地点・方向にロボットを配置し、パーティクル数を75で固定した場合と適応的に調整した場合とで4.2節と同様の方法で自己位置推定を行った。ここでパーティクル数75は今回用いた計算機において計算速度が 33ms となったパーティクル数である。角度分解能は 6deg を用いる。

Table 3, Table 4, Table 5に各測定地点における推定結果の平均値および最大値と最小値を示す。

Table 3とTable 4から測定点1や測定点2において推定精度は要求精度を満たしており、パーティクル

数の違いによって精度に差はほとんど見られない。これは、この2つの地点は白線の角のように特徴的な点が近くに存在したため、パーティクルの分布が広がりにくく、位置を推定するのに十分なパーティクル数が確保されていたためと考えられる。しかしながらTable 5をみると、平均値は共に要求精度を満たしているが、最小値はパーティクル数を固定した場合の方が適応的に調整した場合に比べ 0.3m 程度大きい瞬間的に要求精度を逸脱した誤差が y 軸方向に見られる。ここでFig. 9に示す x 軸、 y 軸両方向の推定誤差の和の履歴より、パーティクルを固定した場合は一時的に要求精度を逸脱する 0.5m 以上の推定誤差が断続的に存在するのがわかる。一方、適応的にパーティクル数を調整している場合は、推定精度は推定全体を通して維持されている。したがって、推定精度と計算時間を同時に考慮するためにはパーティクル数を適応的に調整したほうが良いと考えられる。

Table 1 Estimation result with fixed number of particles

Angular resolution [deg]	Average (x axis)[m]	Average (y axis)[m]	Average (θ)[deg]
2	1.78	3.16	89.7
3	1.72	3.15	89.3
4	1.77	3.16	91.1
6	1.79	3.15	91.3
12	1.70	3.15	88.5
36	0.72	3.09	84.3

Table 2 Estimation result with adaptively changed number of particles

Angular resolution [deg]	Average (x axis)[m]	Average (y axis)[m]	Average (θ)[deg]
2	1.76	3.18	87.8
3	1.72	3.18	88.2
4	1.74	3.18	88.6
6	1.67	3.15	83.8
12	1.66	3.15	87.6
36	0.70	3.12	81.4

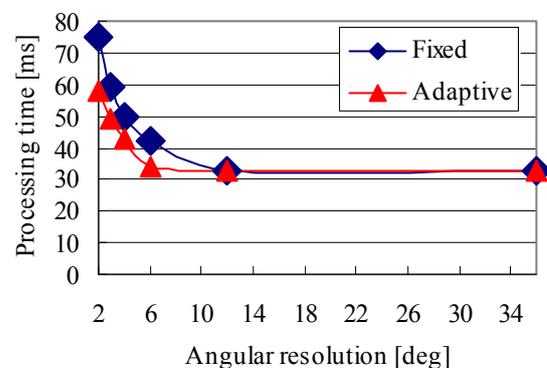


Fig. 3 Processing time of each angular resolution

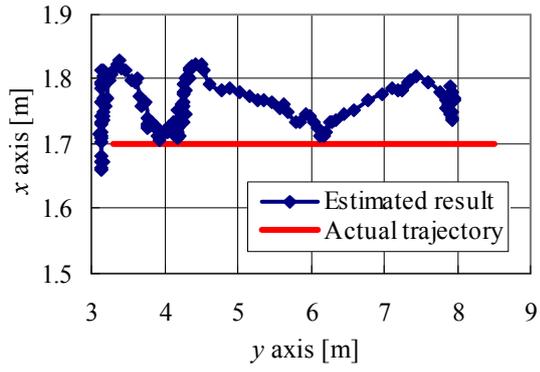


Fig. 4 Estimation result on straight trajectory with fixed number of particles

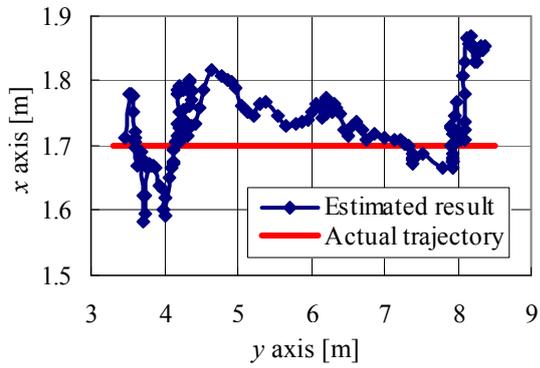


Fig. 5 Estimation result on straight trajectory with adaptively changed number of particles

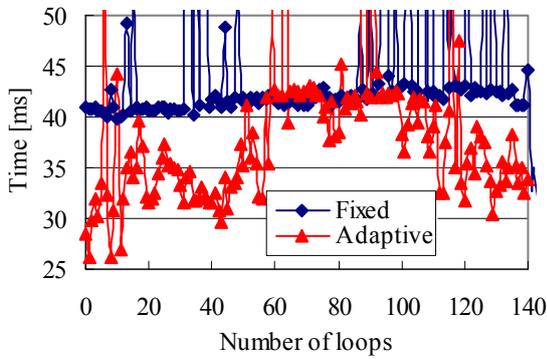


Fig. 6 Time history of processing time

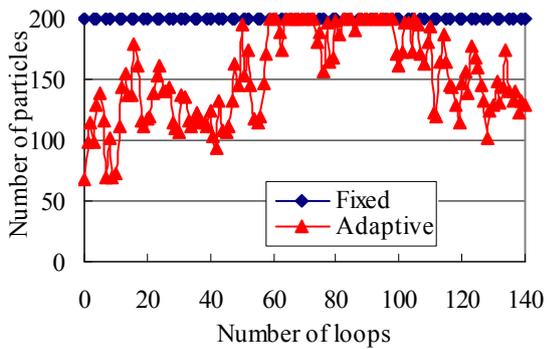


Fig. 7 Time history of number of particles

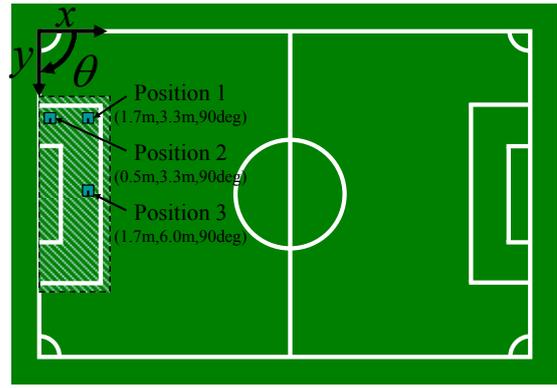


Fig. 8 Robots position

Table 3 Estimation result (Position 1)

		x axis	y axis	θ [deg]
Fixed ($N=75$)	Maximum	1.71	3.19	88.6
	Average	1.64	3.15	87.2
	Minimum	1.57	3.12	86.1
Adaptive	Maximum	1.73	3.20	87.7
	Average	1.67	3.15	83.8
	Minimum	1.59	3.10	77.5

Table 4 Estimation result (Position 2)

		x axis	y axis	θ [deg]
Fixed ($N=75$)	Maximum	0.51	3.24	92.8
	Average	0.47	3.18	88.3
	Minimum	0.44	3.13	84.0
Adaptive	Maximum	0.51	3.29	92.4
	Average	0.47	3.19	88.2
	Minimum	0.44	3.12	84.1

Table 5 Estimation result (Position 3)

		x axis	y axis	θ [deg]
Fixed ($N=75$)	Maximum	1.75	6.15	95.0
	Average	1.67	5.72	90.5
	Minimum	1.60	5.21	86.3
Adaptive	Maximum	1.73	6.13	93.5
	Average	1.67	5.79	91.0
	Minimum	1.62	5.50	88.8

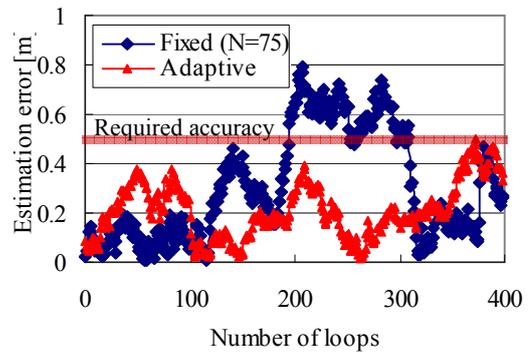


Fig. 9 Time history of estimation error at position 3

5. 結言

本研究では RoboCup サッカー中型リーグを題材とし、ロボットをより効率的に動作させるため、Monte Carlo Localization (MCL)における局所的自己位置推定においてパーティクル数を適応的に調整する手法を検討し、実機実験によって提案手法が可能な限り計算時間を短縮しながら推定精度を維持することができることを示した。また、計算時間を減少させるため、白線探索時の角度分解能についても検討し、自己位置推定に利用可能な角度分解能の1つの指針を実験により示した。その中で、推定精度を維持しながら計算時間を減少させる場合、細かい角度分解能で推定を行うためにはパーティクル数を適応的に調整すると有効であると判った。

今後の課題としては、今回は行わなかった大域的自己位置推定における計算時間の短縮や誘拐問題への対応、ロボットの場所や周囲の状況によって異なると考えられる必要な角度分解能の検討などが挙げられる。

謝辞

本研究の一部は文部科学省グローバル COE プログラム「環境共生・安全システムデザインの先導拠点」によるものであることを記し、謝意を表す。

参考文献

- [1] Frank Dellaert, Dieter Fox, Wolfram Burgard and Sebastian Thrun, Monte Carlo Localization for Mobile Robots, In Proc. of IEEE ICRA, pp.1322-1328, 1999
- [2] Sebastian Thrun, Dieter Fox, Wolfram Burgard and Frank Dellaert, Robust Monte Carlo localization for mobile robots, Artificial Intelligence 128(2001)99-141
- [3] Dieter Fox, KLD-sampling: Adaptive particle filters. In Advances in Neural Information Processing Systems 14, 2002
- [4] K.Morisaki, M.Takahashi and K.Yoshida, Line Based Self-Localization for Autonomous Omni-directional Soccer Robot, 26th Annual Conference of the Robotics Society of Japan, 2008.
- [5] K.Demura and Yu Nakagawa, Monte Carlo Localization using White Lines of Omni-Directional Images, 26th Annual Conference of the Robotics Society of Japan, 2008.
- [6] Y.Sato, Y.Takemura, A.Sanada, T.Kitazumi, Y.Ogawa, S.Yamaguchi and H.Miyamoto, A Study of Line Based Localization for RoboCup Middle Size League, Proc. of the 2008 JSME Conf. on Robotics and Mechatronics, 2008.
- [7] E.Menegatti, A.Pretto and E.Pagello, Testing omnidirectional vision-based Monte-Carlo Localization under occlusion, Proc. of Int. Conf. on Intelligent Robots and Systems (IROS04), pp.2487-2494, 2004
- [8] A. Merke, S. Welker and M.Riedmiller, Line Based Robot Localization under Natural Light Conditions, Proc. of ECAI 2004 Workshop on Agents in Dynamic and Real-Time Environments, 2004.
- [9] R.Ueda, T.Arai, K.Asanuma, K.Umeda and H.Osumi, Recovery Methods for Fatal Estimation Errors on Monte Carlo Localization, Journal of the Robotics Society of Japan, 23, 4, pp.466-473, 2005.
- [10] P.Heinemann, J.Haase and A.Zell, A Combined Monte-Carlo Localization and Tracking Algorithm for RoboCup, Proc. of Int. Conf. on Intelligent Robots and Systems (IROS06), pp.1535-1540, 2006.
- [11] C.Kwok, D.Fox, M.Meila, Adaptive Real-time Particle Filters for Robot Localization, Proc. ICRA '03. IEEE Int. Conference on Robotics and Automation, pp.2836-2841, 2003.
- [12] Sebastian Thrun, Wolfram Burgard and Dieter Fox, 上田隆一訳, 確率ロボティクス, 毎日コミュニケーションズ, 2007
- [13] T.Shimizu, K.Morisaki, T.Suzuki, N.Tomoyasu, M.Takahashi and K.Yoshida, EIGEN Keio Univ. Team Description, In RoboCup International Symposium 2008, CD-ROM, 2008