

パーティクルフィルタを用いたロボット制御への取り組み

齋藤 碧[†]小林一郎^{††}
[†] お茶の水女子大学理学部情報科学科 ^{††} お茶の水女子大学大学院人間文化創成科学研究科理学専攻

1 はじめに

近年、システムやプラントの無人化に伴い、自律した行動をするロボットの研究が進められている。自律性能を高めるにはロボットの制御は外乱に強く柔軟であることが望ましい。制御対象の状態を確率分布で近似して表現するため、外乱からの影響にも柔軟に対応できるパーティクルフィルタが注目を集めている [2][3]。

本研究では、パーティクルフィルタをシミュレータ上で動作するロボットのコントローラに組込んだ制御を行い、その有効性を示す。

2 パーティクルフィルタ

パーティクルフィルタは、観測できない状態ベクトル $\mathbf{x}_t$ を観測可能な観測ベクトル $\mathbf{y}_t$ から推定する手法である。状態 $\mathbf{x}_t$ と観測値 $\mathbf{y}_t$ は、それぞれ以下に示すシステムモデル (1)、観測モデル (2) によって得られる。

$$\mathbf{x}_t = F_t \mathbf{x}_{t-1} + G_t \mathbf{v}_t \quad (1)$$

$$\mathbf{y}_t = H_t \mathbf{x}_t + \mathbf{w}_t \quad (2)$$

$\mathbf{v}_t$ は l 次元のシステムノイズ、 $\mathbf{w}_t$ は m 次元の観測ノイズを表す。 F_t , G_t , H_t はそれぞれ変数に対応する係数を表す行列である。また、 K 個の重み付けされた粒子の集合 $X_t = \{(\mathbf{x}_t^{(k)}, \pi_t^{(k)})\}_{k=1}^K$ で状態 $\mathbf{x}_t$ を表す。ここで $\pi_t^{(k)}$ は個々のパーティクルの重みである。パーティクルフィルタのアルゴリズムを以下に示す。

1. 初期設定：ランダムな K 個の $\mathbf{x}_{t-1}$ を生成する
2. 予測：式 (1) に従い $F_t \mathbf{x}_{t-1} + G_t \mathbf{v}_t \rightarrow \mathbf{x}_t$ として $\mathbf{x}_t$ を予測値とする
3. 尤度計算： $\mathbf{x}_t$ と $\mathbf{y}_t$ を比較し重み π_t を計算する
4. リサンプリング： π_t に比例した確率で $\mathbf{x}_t$ を K 個抽出する
5. 時間更新： $t \rightarrow t+1$ として 2~4 を繰り返す

パーティクルフィルタは、予測、尤度計算、リサンプリングを繰り返して時間更新を行う。制御対象の用途に合わせて対象の追跡や予測の変更が可能である。

An Approach to Controlling a Robot with Particle Filter

[†] Midori SAITO(g0920515@is.ocha.ac.jp)

^{††} Ichiro KOBAYASHI(koba@is.ocha.ac.jp)

Department of Informaitoin Science, Faculty of Sciences, Ochanomizu University ([†])

Advanced Sciences, Graduated School of Humanities and Sciences, Ochanomizu University (^{††})

2-1-1 Otsuka, Bunkyo-ku, Tokyo 112-8610, Japan

3 実験

パーティクルフィルタの状態推定機能を用いた2つのロボット制御実験を以下に示す。

3.1 実験1：ロボットの自己位置推定

実験1では、3D ロボットシミュレータ Simbad[4] を用いてロボットの位置座標をパーティクルフィルタで追跡し、自己位置推定を検証する。対象ロボットには、80 カウンタごとに回転速度をランダムに与え、それ以外は直進するように設定した。また、壁に衝突した際は、初期位置に戻るようにした。パーティクル数は100とし、システムノイズは $\mathbf{v}_t \sim N(0, 1)$ 、観測ノイズは考慮しない。推定結果はリサンプリング後に出力することにより追跡とみなす。運動方程式は用いず、シミュレータが取得したロボットの x , z 座標のみの推定を行った。

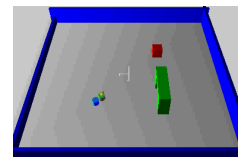


図 1: ロボット (緑) をロボット (青) が追跡

3.1.1 実験結果および考察

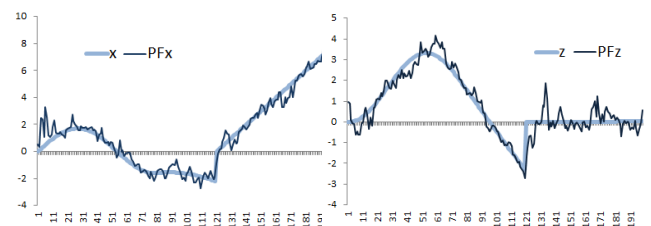


図 2: x 座標推定結果

図 3: z 座標推定結果

図 2, 図 3 はそれぞれ x , z 座標の推定結果である。薄い太線が実際のロボットの観測結果であり、濃い細線がパーティクルフィルタの自己位置推定結果を表す。図が示すように特に z 座標の観測値が 0 付近で漸近している際に安定せず全体的にブレが生じているが、 x , z 共に追従ができていることがわかる。

3.2 実験2：倒立振子の安定化制御

本実験では文献 [2] を参考に、倒立振子の外観 (図 4) や運動方程式・状態方程式を設定した。長さ $l = 0.5(\text{m})$ 、質量 $m = 0.3(\text{kg})$ の回転軸を振り子とみなし、台車の質量 $m_c = 3.0(\text{kg})$ とする。振り子の角度を $\phi(\text{rad})$ とし、

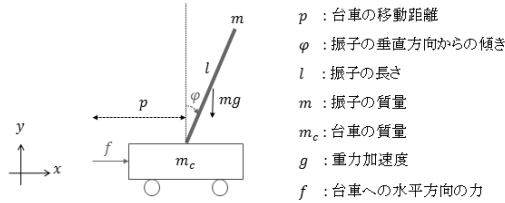


図 4: 倒立振子外観

その角速度を $\dot{\phi}$ (rad/s), 角加速度を $\ddot{\phi}$ (rad/s²) とする. また, 台車の移動距離を p (m), その速度を $\dot{p}$ (m/s), 加速度を $\ddot{p}$ (m/s²) とする. 重力加速度 g は 9.8(m/s²) とした. パーティクル数は 500, $\mathbf{v}_t \sim N(0, 0.01)$, 倒立振子の運動方程式を満たすため使用シミュレータに物理演算エンジン PhysX[5] を用いた. 反発係数を 0.0, 動摩擦係数・静止摩擦係数を 0.5 とした.

3.2.1 倒立振子の状態方程式

倒立振子の運動方程式は以下の式で定義される [2].

$$\ddot{\phi} = \frac{g \sin \phi + \cos \phi \frac{-f - m l \phi^2 \sin \phi}{m_c + m}}{l(\frac{4}{3} - \frac{m \cos^2 \phi}{m_c + m})} \quad (3)$$

$$\ddot{p} = \frac{f + m l (\phi^2 \sin \phi - \ddot{\phi} \cos \phi)}{m_c + m} \quad (4)$$

パーティクルフィルタでは時間更新を行うため, これらの運動方程式をノイズを加えた状態方程式に変換し離散時刻で表現する.

$$dX(t) = h(X(t)) + H(X(t))dW(t) \quad (5)$$

ただし,

$$X(t) = \begin{bmatrix} \phi_1(t) \\ \phi_2(t) \\ p_1(t) \\ p_2(t) \end{bmatrix} = \begin{bmatrix} \phi(t) \\ \dot{\phi}(t) \\ p(t) \\ \dot{p}(t) \end{bmatrix},$$

$$h(X(t)) = \begin{bmatrix} \phi_2(t) \\ \frac{g \sin \phi_1(t) + \cos \phi_1(t) \frac{-f(t) - m l \phi_2^2 \sin \phi_1(t)}{m_c + m}}{l(\frac{4}{3} - \frac{m \cos^2 \phi_1(t)}{m_c + m})} \\ p_2(t) \\ \frac{f(t) + m l (\phi_2(t)^2 \sin \phi_1(t) - \phi_2(t) \cos \phi_1(t))}{m_c + m} \end{bmatrix},$$

$$H(X(t)) = \begin{bmatrix} 0 & 0 \\ \eta_{\phi_2} & 0 \\ 0 & 0 \\ 0 & \eta_{p_2} \end{bmatrix}.$$

また, $W(t)$ は 2 次元ウィナー過程とする. ここで振り子が倒立状態 ($\phi_1 = 0, 2\pi, \dot{\phi}_1 = 0$) 付近の状態方程式を近似する.

$$f_1(\phi_1) = -0.806314\phi_1^2 + 5.06622\phi_1 \quad (6)$$

f_1 の運動方程式は, 式 (6) の様に ϕ_1 の 2 次方程式で近似される [2]. よって, 制御対象を ϕ_1 とし, パーティクルフィルタで予測した出力結果 $\hat{\phi}_1$ を f に変換することで台車に加える力とした. また, $\phi_1 = 0$ 付近において式 (6) を振り子の位置と角加速度により, 以下に示す 4 つに場合分けをして制御した.

$$f(\hat{\phi}_1) = \begin{cases} -f_1(\hat{\phi}_1) & \text{if } (p > 0, \phi_2 > 0) \\ f_1(-\hat{\phi}_1) & \text{if } (p > 0, \phi_2 < 0) \\ f_1(\hat{\phi}_1) & \text{if } (p \leq 0, \phi_2 > 0) \\ -f_1(-\hat{\phi}_1) & \text{if } (p \leq 0, \phi_2 < 0) \end{cases}$$

3.2.2 実験結果および考察

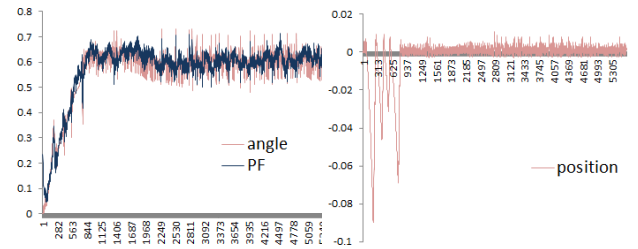
図 5: 角度 ϕ_1

図 6: 振り子の位置

図 5 は薄い線が倒立振子の振り子の角度 ϕ_1 , 濃い線は ϕ_1 のパーティクルフィルタでの予測値を表した. また, 図 6 は振り子の支点からの距離を示した. 初期において振り子の変動が見られるが, 収束値に漸近しており安定化が図れた. 実験 2 では実験 1 と比べ, パーティクル数を増加させたためパーティクルフィルタの推定精度が上がることを確認された.

4 まとめ

シミュレータ上でパーティクルフィルタのロボット制御への性能を評価した. 実際に線形な運動方程式では表現されないものを, 運動方程式の近似とパーティクルフィルタでの柔軟な制御が可能であることを確認した. 今後の課題は, 外乱を加えた制御への適応, 初期位置設定やノイズの加え方によるパーティクルフィルタの改善が挙げられる.

参考文献

- [1] 樋口知之, 粒子フィルタ, 電子情報通信学会誌, Vol.88, No.12, pp.989-994, 2005.
- [2] D.Stahl,J.Hauth,“PF-MPC:Particle Filter-Model Predictive Control”,Berichte des Fraunhofer ITWM,Nr.201(2011).
- [3] 西田健, パーティクルフィルタとロボット制御, 第 3 回横幹連合総合シンポジウム 2010.
- [4] <http://simbad.sourceforge.net/>
- [5] <http://www.nvidia.com/>