

# 尤度を共有する複数のパーティクルフィルタを用いたシステム制御

齋藤 碧 (指導教員：小林一郎)

## 1 はじめに

システムの制御は、運動方程式を用いた様々な手法においてその有効性が示されている。しかし、環境には外乱が含まれており、非線形の状態を線形近似した運動方程式では制御できない場合もある。パーティクルフィルタは制御対象の状態を確率分布で近似して表現するため、近年、システム制御などにも応用され、注目を集めている [2][3]。

本研究では、振り子の倒立安定化問題において、パーティクルフィルタを用い、外乱に対して頑健かつ運動方程式を用いない制御手法を提案する。

## 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。

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

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

$\mathbf{v}_t$  はシステムノイズ、 $\mathbf{w}_t$  は観測ノイズを表す。 $F_t$ ,  $G_t$ ,  $H_t$  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。一般的なパーティクルフィルタのアルゴリズムを以下に示す。

- step1 初期設定：ランダムな  $K$  個の  $\mathbf{x}_{t-1}$  を生成する
- step2 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
- step3 尤度計算： $\mathbf{x}_t$  と  $\mathbf{y}_t$  を比較し重み  $\pi_t$  を計算する
- step4 リサンプリング： $\pi_t$  に比例した確率で  $\mathbf{x}_t$  を  $K$  個抽出する
- step5 時間更新： $t \rightarrow t + 1$  として step2～step4 を繰り返す

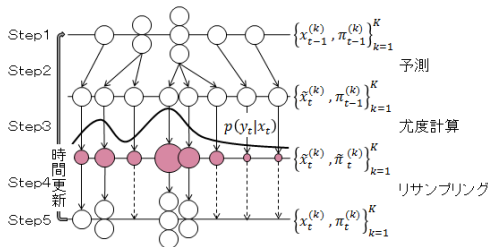


図 1: パーティクルフィルタ

パーティクルフィルタは、図 1 に示すように、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。

## 3 尤度共有による状態推定

パーティクルフィルタは、観測可能な値に対する追従を行う。しかし、実時間内に複雑な制御対象の複数の情報を取得、近似計算をするには制御速度の問題がある。本研究では、適用するパーティクルフィルタの粒子の集合を  $X_t = \{\hat{\mathbf{x}}_t^{(k)}, \hat{\mathbf{z}}_t^{(k)}, \pi_t^{(k)}\}_{k=1}^K$  として  $\mathbf{x}$  と  $\mathbf{z}$  が依存関係にある場合、 $\mathbf{x}$  のみの観測から  $\mathbf{z}$  の値を推定する手法を提案する。提案手法のアルゴリズムを示す。

- step1 初期設定：ランダムな  $K$  個の  $\mathbf{x}_{t-1}, \mathbf{z}_{t-1}$  を生成する
- step2 予測：式 (1) に従い、それぞれノイズを乗せる
- step3 ソート： $\hat{\mathbf{x}}_t, \hat{\mathbf{z}}_t$  をそれぞれ昇順にソートする
- step4 観測値取得：観測可能な  $\mathbf{x}$  を取得、 $\mathbf{z}$  は観測値なし
- step5 尤度計算：重み  $p(x_t | \hat{\mathbf{x}}_{t-1}^{(k)}) (1 \leq k \leq K)$  を計算
- step6 尤度共有：粒子のインデックスに重みを依存した形で  $\hat{\mathbf{z}}$  にも  $\hat{\mathbf{x}}$  と同じ尤度を与える
- step7 リサンプリング： $\pi_t$  に比例した確率で  $\mathbf{x}_t, \mathbf{z}_t$  を  $K$  個抽出する
- step8 時間更新： $t \rightarrow t + 1$  として step2～step7 を繰り返す

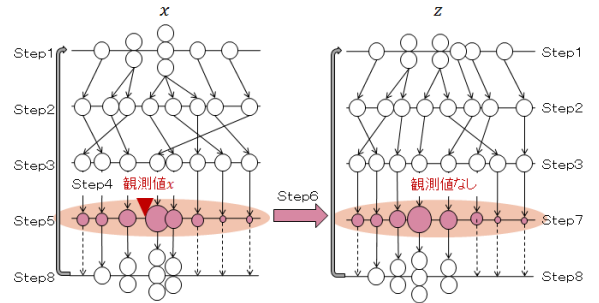


図 2: 尤度を共有する 2 連のパーティクルフィルタ

## 4 実験：倒立振子の安定化制御

本研究では、倒立振子を制御対象とし、地面に障害物が存在する場合に、運動方程式による制御と提案手法である運動方程式を用いずに振り子の傾きと台車にかかる力をパーティクルフィルタで推定する制御とを比較する。実験では文献 [2] を参考に、倒立振子の外観 (図 3) や運動方程式を設定した。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、その角速度を  $\dot{\phi}(\text{rad/s})$ 、角加速度を  $\ddot{\phi}(\text{rad/s}^2)$  とする。また、台車の移動距離を  $p(\text{m})$ 、その速度を  $\dot{p}(\text{m/s})$ 、加速度を  $\ddot{p}(\text{m/s}^2)$  とする。重力加速度  $g$  は  $9.8(\text{m/s}^2)$  とした。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[4] を用いた。外乱を表現するために倒立振子の初期位置付近に一辺  $5.0(\text{mm})$  の立方体を 22 個設置し、反発係数を 0.0、動摩擦係数・静止摩擦係数を 0.5 とした。

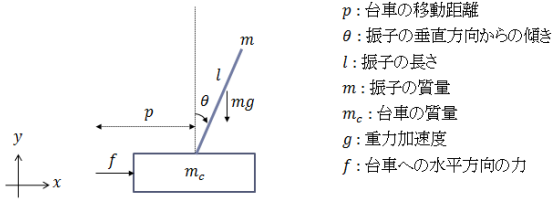


図 3: 倒立振り子外観

#### 4.1 運動方程式を用いた制御

倒立振り子の運動方程式は以下の式で定義される [2].

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

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

ここで振り子が倒立状態 ( $\phi = 0, 2\pi, \dot{\phi} = 0$ ) 付近の方程式を近似する.

$$f_1(\phi) = -0.806314\phi^2 + 5.06622\phi \quad (5)$$

$f_1$  の運動方程式は, 式 (5) の様に  $\phi$  の 2 次方程式で近似される [2]. よって, 制御対象を  $\phi$  とし, 式 (5) より  $\phi$  を  $f$  に変換することで台車に加える力とした. また,  $\phi = 0$  付近において式 (5) を振り子の位置と角速度により, 以下に示す 4 つに場合分けをして制御した.

$$f(\phi) = \begin{cases} -f_1(\phi) & \text{if } (p > 0, \dot{\phi} > 0) \\ f_1(-\phi) & \text{if } (p > 0, \dot{\phi} < 0) \\ f_1(\phi) & \text{if } (p \leq 0, \dot{\phi} > 0) \\ -f_1(-\phi) & \text{if } (p \leq 0, \dot{\phi} < 0) \end{cases} \quad (6)$$

#### 4.2 尤度共有による状態推定を用いた制御

倒立振り子が外乱に影響される場合, 制御量  $f$  は, 振り子の直立状態からの角度  $\phi$  に依存する. 本研究では角度  $\phi$  と台車に加える力  $f$  を推定するパーティクルをそれぞれ 500 個用意する. パーティクルの集合を  $X_t = \{\hat{\phi}_t^{(k)}, \hat{f}_t^{(k)}, \pi_t^{(k)}\}_{k=1}^K$  と設定し, 観測可能な  $\phi$  から  $\hat{f}$  を導く. 推定するにあたり,  $\hat{\phi}$  と,  $\hat{f}$  の初期値の範囲を経験的に以下の様に設定する.

$$0.0 \leq \{\hat{\phi}_0^{(k)}\}_{k=1}^K \leq 0.5 \quad (7)$$

$$0.0 \leq \{\hat{f}_0^{(k)}\}_{k=1}^K \leq 10.0 \quad (8)$$

このように得られた推定値  $\hat{f}$  を  $f \propto \hat{f}$  として出力する. 振り子が傾く方向により 2 つに場合分けをする.

$$f = \begin{cases} r\hat{f} & \text{if } (p < 0.0) \\ -r\hat{f} & \text{if } (p \geq 0.0) \end{cases} \quad (9)$$

上記,  $\hat{\phi}$ ,  $\hat{f}$  の初期値と  $r$  は経験的に求められた値である.

### 4.3 実験結果および考察

#### 運動方程式を用いた制御

図 4 と図 5 は, それぞれ, 運動方程式を用いた際の倒立振り子の振り子の角度  $\phi$ , 振り子の重心の支点からの距離を示した. 運動方程式を用いた制御では, 初期においては制御できていることが観察されるが, 時間が経過するにつれて外乱による誤差が蓄積されてしまい, 最終的に振り子が倒れてしまっている.

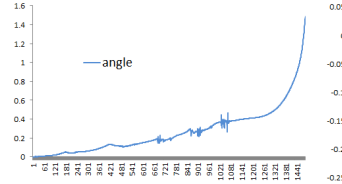


図 4: 角度  $\phi$

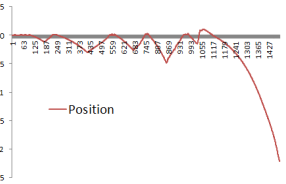


図 5: 振り子位置

#### 尤度共有による状態推定を用いた制御

式 (9) において  $r = 5.0$  として実験と行った. 図 6 は, 濃い太線が実際に観測された  $\phi$  の値, 薄い細線はパーティクルフィルタによる  $\phi$  の推定値を表した. 振り子の位置を示す図 7 により外乱が発生している場合でも左右に振り子が揺れていて制御可能であることが伺える. 図 8 は,  $\hat{\phi}$  と尤度を共有して得られた  $\hat{f}$  の値である. これにより, 実際に観測されない値も提案手法を用いることにより推定できることが示された.

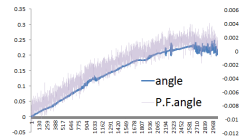


図 6: 角度  $\phi$

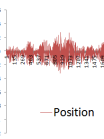


図 7: 振り子位置

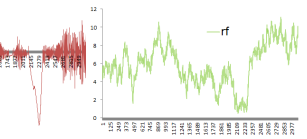


図 8: 推定値  $\hat{f}$

### 5 まとめと今後の課題

シミュレータ上でパーティクルフィルタの倒立振り子安定化制御への性能を評価した. 解析的に近似困難な環境外乱に対し, パーティクルフィルタの状態推定が有効であり, 提案手法では, 尤度を共有したパーティクルフィルタにおいて, 直接観測値が得られない場合でも, 観測可能な値の重みを用いることにより推定が可能となることを示した. 今後の課題は, 安定性の検討, 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推定精度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 回幹幹連合総合シンポジウム 2010.
- [4] <http://www.nvidia.com/>