Online Estimation of Geometric and Inertia Parameters for Multirotor Aerial Vehicles

권건우

Introduction

- Motivation
  - Precise and robust control requires accurate knowledge of geometric and inertia parameters.
  - Exogenous factors cause the change of its physical parameters.
  - Ex) Payload transportation or manipulation
  - In this sense, it is essential to estimate the parameters in real time.

- Previous works related “on-board sensor system”
  - Focus on autonomy even if it utilizes on-board sensors
  - Are on the basis of kinematics
  - Do not cover dynamic models, thus cannot get these parameters
- This paper depends on motor speed measurements to estimate these parameters

Introduction

- Previous works related to “parameter estimation”
  - Offline maximum likelihood approach
  - This approach considers different sensor uncertainties well.
  - But, this method runs offline. (Takes too much time to estimate on flight)

  - Other paper achieves online estimation through recursive least square method
  - However, the method relies on motion capture sensor(Measure Acceleration)
  - Not perfectly on-board sensor system

- Contribution
  1. Real time, on board framework to estimate the dynamic properties
  2. When subject to changes during operation, quickly re-estimate the properties
  3. Offer observability analysis

Preliminaries (v_tools.cpp에서 구현)

$$
\begin{tikzcd}
\frac{\phi}{2}\vec{\boldsymbol{u}} \arrow[r, shift left, "\text{exp map}"] & \boldsymbol{q} \arrow[l, shift left, "\text{log map}"] \arrow[d, "\text{Log map}"] \\
\boldsymbol{Exp}(\theta\boldsymbol{u}) = \boldsymbol{q} & \boldsymbol{Log}(\boldsymbol{q}) = \theta\boldsymbol{u} \arrow[l, "\text{Exp map}"']
\end{tikzcd}
$$

$$\boldsymbol{q} = q_w + \vec{\boldsymbol{q}} = cos\theta + \vec{\boldsymbol{u}}sin\theta$$

$$\boldsymbol{where}\ \theta = \frac{\phi}{2}$$

$$\rightarrow \vec{\boldsymbol{u}} = \frac{\vec{q}}{||\vec{q}||_2}, \theta = atan2(||\vec{\boldsymbol{q}}||_2, q_w)$$

$$\boldsymbol{q} = cos\theta + iu_x sin\theta + iu_y sin\theta + iu_z sin\theta$$

Preliminaries (**geom_inertia_estimator.cpp** 에서 구현)

자세 적분 (qboxplus)

$$ {}^{I}\boldsymbol{q}_{B[t+1]} = {}^{I}\boldsymbol{q}_{B[t]}\,{}^{B[t]}\boldsymbol{q}_{B[t+1]} = {}^{I}\boldsymbol{q}_{B[t]} \boxplus \boldsymbol{\Omega}\Delta\mathrm{t} = {}^{I}\boldsymbol{q}_{B[t]}\boldsymbol{Exp_{q}}(\boldsymbol{\Omega}\Delta\mathrm{t}) $$

카메라에서 측정한 자세와 Temporary belief 자세 사이의 innovation
→ 자세 오차 계산 필요 (qboxminus)

$$ \boldsymbol{q_{cam}} \boxminus \boldsymbol{q_{temp}} = \boldsymbol{Log}\left({}^{\boldsymbol{temp}}\boldsymbol{q_{cam}}\right) = \boldsymbol{Log}\left({}^{\boldsymbol{temp}}\boldsymbol{q_{glob}}\,{}^{\boldsymbol{glob}}\boldsymbol{q_{cam}}\right) = \boldsymbol{Log}\left(\boldsymbol{q^{*}_{temp}}\boldsymbol{q_{cam}}\right) $$

Method

Unlike other papers, this paper takes dynamic model into account to estimate the model parameters.

A. Model

- **Objectives**

1. **Accurately track inertia parameters even when mid-flight changes take place**

2. **Applicable to any kind of UAV**

3. **Allow the integration of measurements from different types of sensors**

Method

Relative position from Center of body to each rotor - Known

Center of Mass – Unknown

A time dependent state variables $\boldsymbol{x_t}(t) = \begin{bmatrix} {}_{W}\boldsymbol{r}_{WM}(t) \\ {}_{W}\boldsymbol{v}_{WM}(t) \\ \boldsymbol{q}_{WM}(t) \\ \boldsymbol{\Omega}(t) \end{bmatrix}$

State variables $\boldsymbol{x}(\boldsymbol{t}) = \begin{bmatrix} \boldsymbol{x_t}(t) \\ \boldsymbol{x_c} \end{bmatrix}$

A time independent state variables $\boldsymbol{x_c} = \begin{bmatrix} m \\ \boldsymbol{I_{3\times1}} \\ {}_{B}\boldsymbol{r_{BM}} \\ {}_{B}\boldsymbol{r_{BP}} \\ {}_{B}\boldsymbol{r_{BI}} \\ \boldsymbol{b}_a \\ \boldsymbol{b}_\Omega \end{bmatrix}$

이 논문을 선정하게 된 결정적인 이유

※ Recursive State Estimation – Probabilistic Robotics

At the core of probabilistic robotics is the idea of estimating state from sensor data.

State estimation addresses the problem of estimating quantities from sensor data that are indirectly observerable, but that can be inferred.

Kalman Filter, Extended Kalman Filter, Unscented Kalman Filter
→This is a type of recursive state estimation method.
→Also, this method belongs to Bayesian filter.

EKF algorithm Predict Update

$EKF(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$ Predict Update

$\bar{\mu_t} = g(u_t, \mu_{t-1})$: Process Model → 역할- 물리 법칙 인지

$$G_t = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_m}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial g_m}{\partial x_1} & \cdots & \frac{\partial g_m}{\partial x_n} \end{bmatrix}$$

$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_T$

요점: Process Model은 물리 법칙을 인지하게끔 해준다.

EKF algorithm Measurement Update

$EKF(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$ Measurement Update

$h_{sensor}(\bar{\mu}_t, u_t)$: 센서 모델이라 부르지만, Temporary belief로 센서 측정값과 동일하게 맞춰준다

$$H_t = \begin{bmatrix} \frac{\partial h_{s1}}{\partial x_1} & \cdots & \frac{\partial h_{s1}}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial h_{sp}}{\partial x_1} & \cdots & \frac{\partial h_{sp}}{\partial x_n} \end{bmatrix} (p \times n \text{ matrix})$$

z_t는 굳이 state variable을 직접적으로 측정하는 값이 아니어도 된다.

이유: Recursive State Estimation은 간접적으로 모든 상태값을 보정하는 것이 가능하기 때문이다.

$K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$ $(n \times p\ matrix)$ → p 개의 상태 관측으로 n개 상태 변수 모두 보정 가능

$$\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t, u_t))$$

$$\Sigma_t = (I - K_t H_t)\bar{\Sigma}_t$$

반환

질량 추정 예제

$F = ma$로 문제 설정하고, p, v, m을 상태 변수로 설정

이를 Extended Kalman Filter로 적용

일정한 힘 인가	PD 제어