SI2003 2003.12.21

飛行船ロボットの画像情報に基づく制御

深尾隆則　藤谷和司　金出武雄

京都大学
カーネギーメロン大学

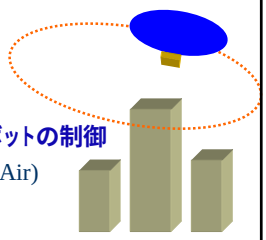
飛行船システムに関する研究

- リモートモニタリング
- 地雷探査
- 携帯電話基地局
- 監視システム

単カメラを備えた飛行船ロボットの制御

UAV, LTA (Lighter Than Air)

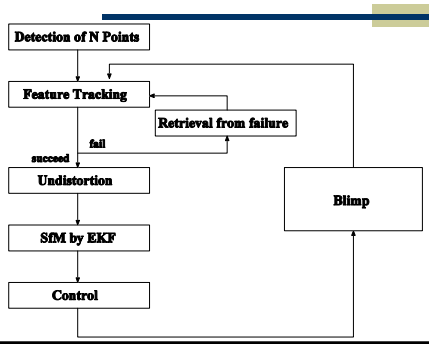
⇒ レスキュー活動など
特定ターゲット周りの
3次元再構成画像の提示



制御システム

- センサとしてカメラだけを用いる
  ⇐ 地面に対する固定点や接地点がない
- SfM (Structure from Motion) アルゴリズムで
  回転，速度，高度を推定して用いるが，
  高度以外の位置は用いない
- 画像平面上の点を直接制御
  画像に基づく制御

System Chart



Point Feature Tracker

Lucas-Kanade Tracker の発展
二段階アルゴリズム，特徴点発見アルゴリズム
特に，回転とスケーリングを伴う場合の正確な追従

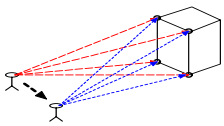
Structure from Motion (SfM)

- 拡張カルマンフィルタを用いる方法 (A. Azarbayejani et.al.)

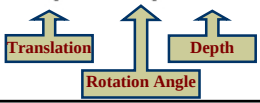
オンライン性を重視
簡潔化されたSfM手法

- 特徴点が同一平面上
- 焦点が既知

推定パラメータ



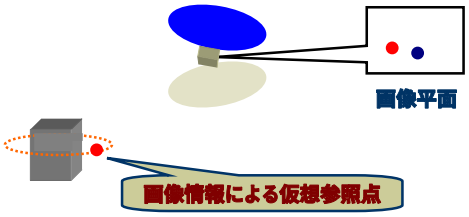
$$[t_x, t_y, t_z, \omega_x, \omega_y, \omega_z, \alpha].$$



飛行船ロボット

- 長さ: 3.5m
- 直径: 1.5m
- モータ: 3

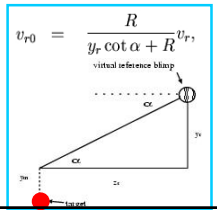
画像に基づくトラッキング制御



モデル (1)

$$\frac{d}{dt}\begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} = V_r - V - \Omega \times \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix},$$

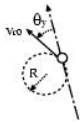
$$v_{r0} = \frac{R}{y_r \cot\alpha + R} v_r,$$

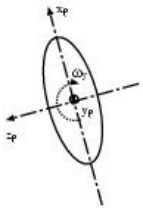


モデル (2)

$$\frac{d}{dt}\begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} = \begin{bmatrix} v_{r0}\cos\theta_y - v \\ u_r - u \\ v_{r0}\sin\theta_y - w \end{bmatrix} - \begin{bmatrix} \omega_y z_p - \omega_z y_p \\ \omega_z x_p - \omega_x z_p \\ \omega_x y_p - \omega_y x_p \end{bmatrix},$$

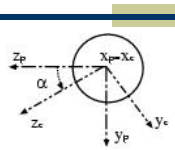
$$\frac{d}{dt}\begin{bmatrix} \theta_x \\ \theta_y \\ \theta_z \end{bmatrix} = \begin{bmatrix} \omega_x \\ \omega_{yr} + \omega_y \\ \omega_z \end{bmatrix},$$





モデル (3)

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix}.$$



仮定:

- ローリングとピッチング運動が無視できる
- 重心や浮力中心が y_c–z_c 平面上

$$\frac{d}{dt}\begin{bmatrix} x_c \\ y_c \\ z_c \\ \theta_y \end{bmatrix} = \begin{bmatrix} -1 & \sin\alpha y_c - \cos\alpha z_c & 0 \\ 0 & -\sin\alpha x_c & -\cos\alpha \\ 0 & \cos\alpha x_c & -\sin\alpha \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega_y \\ u \end{bmatrix} + \begin{bmatrix} v_{r0}\cos\theta_y \\ -v_{r0}\sin\theta_y \sin\alpha \\ v_{r0}\sin\theta_y \cos\alpha \\ \omega_{yr} \end{bmatrix}$$

モデル (4)

$$\begin{bmatrix} X \\ Y \end{bmatrix} = f\begin{bmatrix} \frac{x_c}{z_c} \\ \frac{y_c}{z_c} \end{bmatrix}$$

画像平面

SfM

$$\frac{d}{dt}\begin{bmatrix} X \\ Y \\ \theta_y \end{bmatrix} = \begin{bmatrix} -\frac{Y}{y_c} & Y\sin\alpha - f\cos\alpha - \frac{X^2}{f}\cos\alpha \\ 0 & -X\sin\alpha - \frac{XY}{f}\cos\alpha \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega_y \end{bmatrix}$$

SfM

$$+ \begin{bmatrix} \frac{Y}{y_c} v_{r0}\cos\theta_y - \frac{XY}{fy_c} v_{r0}\sin\theta_y \cos\alpha \\ -\frac{Y}{y_c} v_{r0}\sin\theta_y \sin\alpha - \frac{Y^2}{fy_c} v_{r0}\sin\theta_y \cos\alpha \\ \omega_{yr} \end{bmatrix}$$

$$+ \begin{bmatrix} \frac{XY}{fy_c}\sin\alpha u \\ \left(-\frac{Y}{y_c}\cos\alpha + \frac{Y^2}{fy_c}\sin\alpha\right) u \\ 0 \end{bmatrix},$$

無視

$$K_u(y_p - y_r) + \dot{y}_r$$

$$\frac{d}{dt} y_p = -u, \qquad y_c = \frac{Y}{f\sin(\alpha-\theta_x) + Y\cos(\alpha-\theta_x)} y_p$$

モデル (5)

ビジュアルフィードバックのための変換 (F. Conticelli et al.)

$$\zeta_1 = \frac{X}{Y}, \qquad \zeta_2 = \frac{1}{Y}.$$

$$\frac{d}{dt}\begin{bmatrix}\zeta_1\\ \zeta_2\\ \theta_y\end{bmatrix} = \begin{bmatrix} -\frac{1}{y_c} & \sin\alpha - f\cos\alpha\zeta_2 + \sin\alpha\zeta_1^2 \\ 0 & \zeta_1\zeta_2\sin\alpha + \frac{1}{f}\cos\alpha\zeta_1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} v \\ \omega_y \end{bmatrix} + \begin{bmatrix} \frac{1}{y_c}v_{r0}\cos\theta_y + \frac{1}{y_c}v_{r0}\sin\theta_y\sin\alpha\zeta_1 \\ \frac{1}{fy_c}v_{r0}\sin\theta_y\cos\alpha + \frac{1}{y_c}v_{r0}\sin\theta_y\sin\alpha\zeta_2 \\ \omega_{yr} \end{bmatrix}.$$

モデル (6)

トラッキング制御のための変換

$$\eta_1 = \zeta_1 \quad \eta_2 = \zeta_2 - \zeta_2^d \quad \eta_3 = \theta_y$$

$$\frac{d}{dt}\begin{bmatrix}\eta_1\\ \eta_2\\ \eta_3\end{bmatrix} = \begin{bmatrix} -\frac{1}{y_c} & \sin\alpha - f\cos\alpha(\eta_2+\zeta_2^d) + \sin\alpha\eta_1^2 \\ 0 & \eta_1(\eta_2+\zeta_2^d)\sin\alpha + \frac{1}{f}\cos\alpha\eta_1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} v \\ \omega_y \end{bmatrix} + \begin{bmatrix} \frac{1}{y_c}v_{r0}\cos\eta_3 + \frac{1}{y_c}v_{r0}\sin\eta_3\sin\alpha\eta_1 \\ \frac{1}{fy_c}v_{r0}\sin\eta_3\cos\alpha + \frac{1}{y_c}v_{r0}\sin\eta_3\sin\alpha(\eta_2+\zeta_2^d) \\ \omega_{yr} \end{bmatrix}$$

$$\equiv \begin{bmatrix} h_{11} & h_{12} \\ 0 & \eta_1 h_{22} \\ 0 & 1 \end{bmatrix}\begin{bmatrix} v \\ \omega_y \end{bmatrix} + \begin{bmatrix} h_{31} \\ v_{r0}\sin\eta_3 h_{32} \\ \omega_r \end{bmatrix},$$

コントローラ (1)

バックステッピングの考えから (Z. P. Jiang and H. Nijmeijer)

もし $\eta_1 = 0$ **ならば**

$$\dot{\eta}_2 = \sin\eta_3 v_{r0}\frac{1}{y_c}\left\{\sin\alpha(\eta_2+\zeta_2^d) + \frac{1}{f}\cos\alpha\right\},$$

ここで, $\eta_3 = -\varphi(\eta_2 v_{r0}) \qquad \varphi(s) = \frac{s}{1+s^2}.$ **と選ぶと**

$$\dot{\eta}_2 = -v_{r0}\sin\varphi(\eta_2 v_{r0}) + (\text{positive term})$$

この観点から以下のようにおく

$$\bar{\eta}_3 = \eta_3 + \varphi(\eta_2 v_{r0}),$$

$$\Rightarrow \quad \eta_1 = 0, \quad \bar{\eta}_3 = 0$$

コントローラ (2)

Kinematic controller

$$\omega_y = -\frac{1}{K_3}(C_2\bar{\eta}_3 + K_3 h_5 + K_2\eta_2 v_{r0} h_{32} h_6),$$

$$v = \frac{y_c}{K_1}\{C_1\eta_1 + (K_1 h_{12} + K_2\eta_2 h_{22})\omega_y$$

$$V = \frac{K_1}{2}\eta_1^2 + \frac{K_2}{2}\eta_2^2 + \frac{K_3}{2}\bar{\eta}_3^2,$$

$$\dot{V} = -C_1\eta_1^2 - K_2 h_{32} v_{r0}\eta_2\sin\varphi(v_{r0}\eta_2) - C_2\bar{\eta}_3^2 \leq 0,$$

収束性

仮定として $v_{r0} \neq 0$

$$\lim_{t\to\infty}\eta_2 = 0$$

$$\lim_{t\to\infty}\eta_3 = \lim_{t\to\infty}(\bar{\eta}_3 - \varphi(v_{r0}\eta_2)) = 0$$

$\Rightarrow$ $X \to 0 \quad Y \to Y^d \quad \theta \to 0$

$x_p \quad z_p$ **は望ましい値に収束する!**

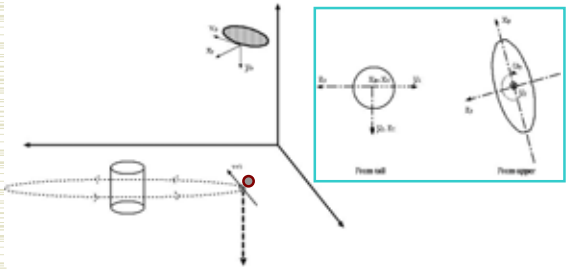
Dynamic Controller

$$\tau_v = -P_1(v - \bar{v}),$$

$$\tau_\omega = -P_2(\omega_y - \bar{\omega}_y),$$

$$\tau_u = -P_3(u - \bar{u})$$

位置に基づく飛行船の制御のためのモデル



位置に基づく飛行船の制御

$$\frac{d}{dt}\begin{bmatrix} x_c \\ y_c \\ z_c \\ \theta_y \end{bmatrix} = \begin{bmatrix} -1 & y_c & 0 \\ 0 & -x_c & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega_y \\ u \end{bmatrix} + \begin{bmatrix} v_{r0}\cos\theta_y \\ -v_{r0}\sin\theta_y \\ u_r \\ \omega_{yr} \end{bmatrix}.$$

Kinematic Controller

$$\begin{aligned} v_f &= K_1 x_c + v_{r0}\cos\theta_y, \\ \omega_{yf} &= -K_3 \sin\theta_y + K_2 y_c v_{r0} - \omega_{yr}, \\ u_f &= K_4 z_c, \end{aligned}$$

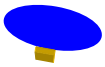
Dynamic Controller

$$\begin{aligned} \tau_v &= -P_1(v - \bar{v}), \\ \tau_\omega &= -P_2(\omega_y - \bar{\omega}_y), \\ \tau_u &= -P_3(u - \bar{u}) \end{aligned}$$

実験環境

制御手順

- テーブルの3角が自動で得られる
- 簡単な安定化コントローラが端に平行になるように適用される
  EKF が良好な推定のために進む
- 仮想的な望ましい軌道がある角の回りに生成される
- 主コントローラが劣駆動飛行船ロボットに適用される



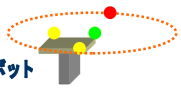
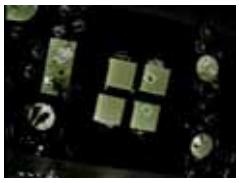
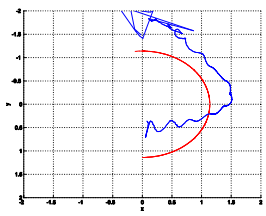


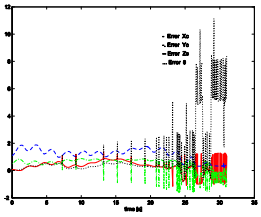
Image-based Control の結果



30 frames/s

Image-based Control の結果

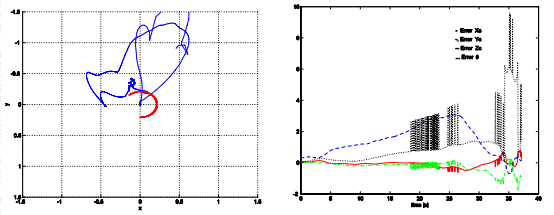




Tracking Errors
Red: Xc Green: Yc
Blue: Zc Black: θ

Position-based Control の結果

Position-based Control の結果



Tracking Errors
Red: Xc Green: Yc
Blue: Zc Black: θ

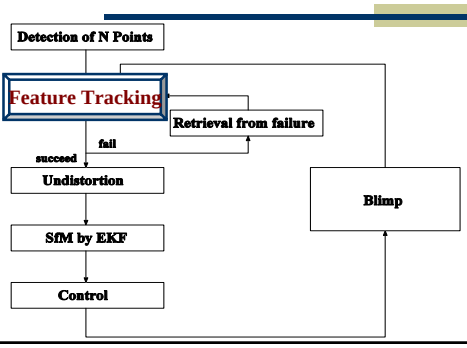
おわりに

- 画像に基づくトラッキング制御は
  位置に基づく制御よりも良好に機能した

⇩

画像平面上の点を直接制御することにより
SfMアルゴリズムから得られた推定値の
利用を減らした結果

System Chart



Point Feature Tracker

Extension of Lucas-Kanade Tracker:
Two Step Algorithm, Detection Algorithm

$$\epsilon = \int_W [J(A\mathbf{x}+\mathbf{d}) - I(\mathbf{x})]^2 w(\mathbf{x}) d\mathbf{x}$$

$$\mathbf{d} = [d_x,\ d_y]^T$$

$$A = \begin{bmatrix} s_1 & 0 \\ 0 & s_1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \approx \mathbf{1} + \begin{bmatrix} s_0 & -\theta \\ \theta & s_0 \end{bmatrix}$$

Very accurate tracking
of a point feature
with rotation & scaling

Two Step Algorithm

Translation

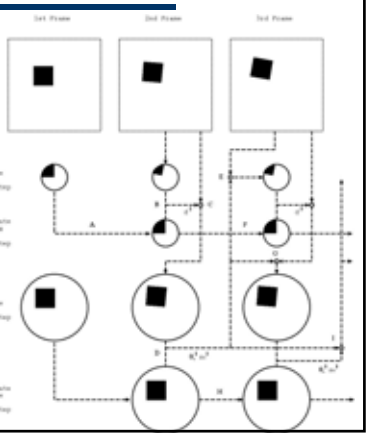
$$\int_{W_1} G_c w(\mathbf{x}) d\mathbf{x} \begin{bmatrix} d_x \\ d_y \end{bmatrix} = \int_{W_1} [I(\mathbf{x}) - J(\mathbf{x})] \begin{bmatrix} g_x \\ g_y \end{bmatrix} w(\mathbf{x}) d\mathbf{x},$$

⇩ slide

Rotation & Scaling

$$\int_{W_2} G_a w(\mathbf{x}) d\mathbf{x} \begin{bmatrix} \theta \\ s_0 \end{bmatrix} = \int_{W_2} [I(\mathbf{x}) - J(\mathbf{x})] \begin{bmatrix} -g_x y + g_y x \\ g_x x + g_y y \end{bmatrix} w(\mathbf{x}) d\mathbf{x}.$$

Inverse algorithm



Ideal Affine Case

Automatic detection
of good point features
for tracking

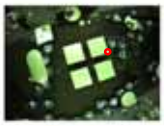
Two step algorithm
to track accurately

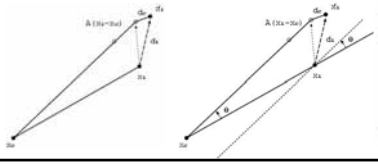
First frame image & warped image by estimation

Two Step Algorithm

Retrieval from Failure

- Failure of feature tracking due to large movement of blimp
- Recovery by geometric restriction among correctly tracked points





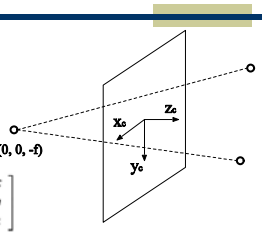
Model for SfM

Central Projection

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \frac{1}{1+z_c\beta}\begin{bmatrix} x_c \\ y_c \end{bmatrix}$$

Object and Camera coordinates

$$\begin{bmatrix} x_c \\ y_c \\ z_c\beta \end{bmatrix} = \begin{bmatrix} t_x \\ t_y \\ t_z\beta \end{bmatrix} + \begin{bmatrix} 1 & & \\ & 1 & \\ & & \beta \end{bmatrix} R \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$



$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} X \\ Y \\ 0 \end{bmatrix} + \alpha \begin{bmatrix} X\beta \\ Y\beta \\ 1 \end{bmatrix}$$ ←(X,Y) Initial value on image plane

Estimated Parameters

$$R = \begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix}$$

$$q = [q_0, q_1, q_2, q_3]^T$$

$$\delta q = [\sqrt{1-\epsilon}, \omega_x/2, \omega_y/2, \omega_z/2]^T, \quad \varepsilon = (\omega_x^2+\omega_y^2+\omega_z^2)/4.$$

Simplified SfM algorithm

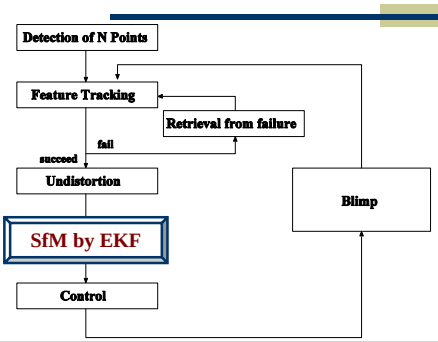
- Features on the same plane
- Known focal length

Estimated Parameters

$$[t_x, t_y, t_z, \omega_x, \omega_y, \omega_z, \alpha].$$

Translation | Depth | Rotation Angle

System Chart



SfM by EKF

$$x_t = [t_x, t_y, t_z, \omega_x, \omega_y, \omega_z, \alpha].$$

$$x_{t+1} = \Phi x_t + \bar{w}_t,$$
$$y_t = H(x_t)x_t + \bar{v}_t$$

→ Extended Kalman Filter

Φ: Identity Matrix — assume slow movement

➢ Arbitrary scaling — resolve by setting of desired path