Passivity-based Pose Synchronization: Experimentation using Quadrotor

Hayato Aoki
FL12-19-3
25th, Dec., 2012

Summary up to Middle Report

Background

Mobile sensor network achieve specified task efficiently

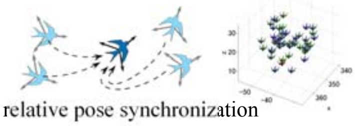
- measurement
- surveillance
- searching & rescue

Advantage

- robustness to agents failures
- adaptation to environmental changes

Idea of control method

⇒ Flocking Algorithm



Middle Report

- Proposing control law for relative pose synchronization
- Simulation in *SE*(3)

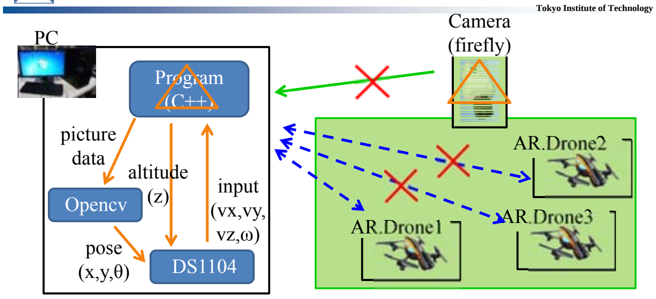
Objective

Proposing flocking algorithm and Implementation on quadrotor

Outline

- Summary up to Middle Report
- Experimentation setting using Quadrotor
- Flocking Algorithm

Previous of Experiment Settings

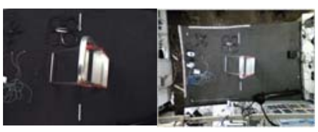


Problem

- old C++ program is too complicated and heavy.
- only one PC can connect one AR.Drone.
- Cvdrone cannot recognize firefly.
- Firefly's angle of view is too small, etc…

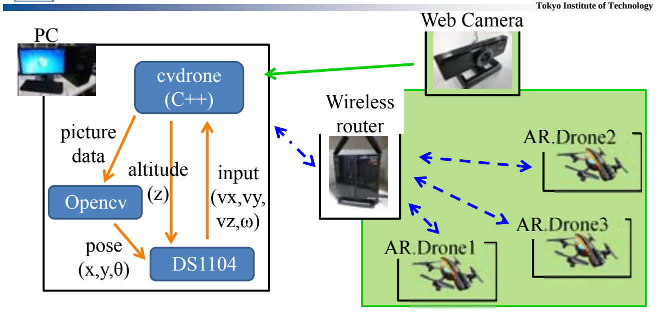
Solutions

- C++ program is too complicated and heavy.
  ⇒ Use new C++ program, cvdrone.
- only one PC can connect One AR.Drone.
  ⇒ Connect through a network access point (wireless router) and modify the drone's firmware and cvdrone.
- Cvdrone cannot recognize firefly.
- Firefly's angle of view is too small.
  ⇒ Introduce a new camera.



FireFly new

Plan of Experiment Settings



I can control multiple AR.Drone using One PC.

Remaining Task

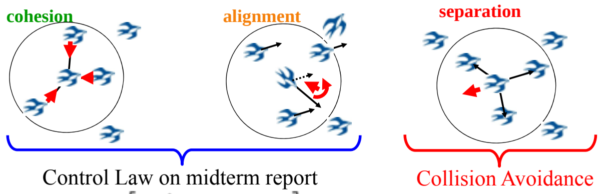
- Camera calibration
- Make simulink block
- Experiment

Outline

- Summary up to Middle Report
- Experimentation setting using Quadrotor
- Flocking Algorithm

Flocking Algorithm

Flocking consists of three rules, which are cohesion, alignment[1]



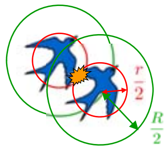
$$V^b_{wi} = K_i \sum_{j\in\mathcal{N}_i} \begin{bmatrix} e^{-\hat{\xi}\theta_{wi}}(p_{wj}-p_{wi}) \\ \mathbf{sk}(e^{-\hat{\xi}\theta_{wi}}e^{\hat{\xi}\theta_{wj}})^\vee \end{bmatrix} + \begin{bmatrix} v_d \\ \omega_d \end{bmatrix}$$

cohesion alignment

Collision: $\|p_{wi} - p_{wj}\| \leq r$

r: Collision Distance

R: Distance for Collision Avoidance



Collision Avoidance[2]

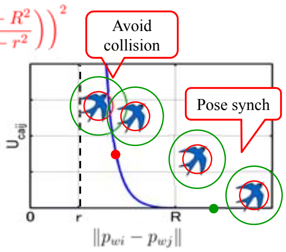
Potential Function for Collision Avoidance

$$U_{ij}(\|p_{wi}-p_{wj}\|) = \left(min\left(0, \frac{\|p_{wi}-p_{wj}\|^2 - R^2}{\|p_{wi}-p_{wj}\|^2 - r^2}\right)\right)^2$$

$$i \neq j,\ R > r > 0$$

If $\|p_{wi} - p_{wj}\| = r$
then $U_{ij}(\|p_{wi}-p_{wj}\|) = \infty$

If $\|p_{wi} - p_{wj}\| \geq R$
then $U_{ij}(\|p_{wi}-p_{wj}\|) = 0$



Control Low(Flocking Algorithm)

$$V^b_{wi} = K_i \sum_{j\in\mathcal{N}_{Oi}} \begin{bmatrix} e^{-\hat{\xi}\theta_{wi}}(p_{wj}-p_{wi}) \\ \mathbf{sk}(-e^{\hat{\xi}\theta_{wi}}e^{\hat{\xi}\theta_{wj}})^\vee \end{bmatrix} + \begin{bmatrix} v_d \\ \omega_d \end{bmatrix} - \sum_{j\in\mathcal{N}_{Ci}} \begin{bmatrix} e^{-\hat{\xi}\theta_{wi}}\frac{\partial U_{ij}}{\partial p} \\ 0 \end{bmatrix}$$

cohesion alignment separation

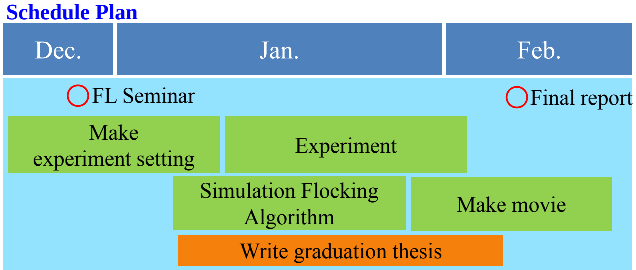
$\mathcal{N}_{Oi}$: Neighbor for pose synch $\mathcal{N}_{Ci}$: Neighbor for Collision Avoidance

Conclusion Schedule

Conclusion

- Progress of making experiment setting
  - Solving some problems
- Proposing Flocking Algorithm
  - Adding collision avoidance to middle conclusion

Schedule Plan



References

[1] C. W. Reynolds, “Flocks, Herds and Schools: A Distributed Behavioral Model,” *Computer Graphics*, Vol. 21, No. 4, pp. 25–34, 1987.

[2] T. Hatanaka, Y. Igarashi, M. Fujita and M. W. Spong, "Passivity-based Pose Synchronization in Three Dimensions," *IEEE Transactions on Automatic Control*, Vol. 57, No. 2, pp. 360–375, 2012.

[3] T. Ibuki, T. Hatanaka and M. Fujita, “Further Results on Passivity-based 3D Pose Synchronization,” *IEEE Transactions on Automatic Control*, 2013

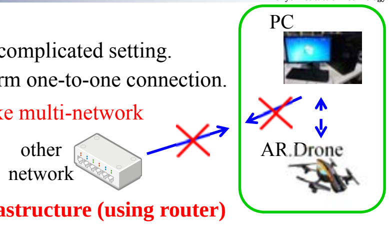
Appendix

Connection method of AR.Drone

Usual method ⇨ **Ad-hoc**

- It's unnecessary to do complicated setting.
- It's only possible to form one-to-one connection.

⇨ Impossible to make multi-network



This time method ⇨ **Infrastructure (using router)**

- It's necessary to prepare a wireless router and do complicated setting.
- It's possible to make multi-network.