Final Project: Distributed Leader-Follower Affine Formation Maneuver Control for High-Order Multiagent Systems

Soowhan Yi
Mechanical Engineering
University of Washington
syi94@uw.edu

I. Introduction

Due to emerging demand for its potential applications in robotics, aerospace, and automobile industry, multiagent formation control and manuever has been widely studied. In applications such as unmanned ground vehicles, with satillites or with drones, multiagent formation control and manuever algorithm can be designed to maneuver through a unknown map with different designed formation control methods. To design the algorithm behind those controlled formation, it traditionally used agents'relative displacements, distances, or bearing. Although there were different previous works utilizing these traditional methods to solve formation maneuver control problem, they were not able to realize translational, rotational and scaling manuever at the same time.

To solve this formation maneuver control problem, the paperutilizes stress matrices among other solutions that realizes those manuevers at the same time, such as berycentric coordinates-based approach and complex Laplacians-based approach, because the berycentric coordinate-based approach requires relative rotation matrix and complex Laplacians-based approach is only applicable in some special dimension, and therefore stress matrix-based approach is more flexible and realizable. In "S.zhao, Affine Formation", the formation maneuver control problem is solved with stress matrix-based approach and achieves translational, rotational, scaling, and shearing maneuver concurrently. However, it does not consider the case where the leaders of the formation have time varying acceleration. The followers following leaders with time varying acceleration would have coupled inputs as those followers have to use global information of acceleration of the leaders to calculate control input.

Therefore the paper suggests two layered leader-follower control strategy with stress matrix to fully distribute the control strategy for followers in affine formation maneuver. Also, for the leaders, it achieves autonomous maneuver control algorithm with information of their desired trajectory and finite time derivatives.

II. Mathematical formulation

In order to properly formulate this multiagent formation manuever control problem, the paper utilizes two layered leader follower strategy and stress matrix in undirected graph. First, in graph theory, this stress vector $\omega = [\omega_1, \omega_2, \cdots, \omega_m] \in R^m$ is said to be in equilibrium, when it satisfies,

$$\sum_{j \in N_i} \omega_{ij}(p_j - p_i) = 0,$$

where this p_i and p_j are the position of i and j agents, and this vector $\omega_{ij}(p_j - p_i)$ represents the tension force between the two agents. In a matrix form, above equation can be expressed as

$$(\Omega \otimes I_d)p = 0$$

where $p = [p_1^T, p_2^T, \cdots, p_n^T]^T \in R^{dn}$, and $\Omega \in R^{n \times n}$ satisfying

$$[\Omega]_{ij} = \begin{cases} 0, & i \neq j, (i,j) \notin \varepsilon \\ -\omega_{ij}, & i \neq j, (i,j) \in \varepsilon \\ \sum_{j \in N_i} \omega_{ik}, & i = j. \end{cases}$$

By letting these tension force in each agents with its neighbors to sumed to be 0, then their formations would be structurally rigid. This structural rigidity of the formation is proved to be universally rigid if and only if there exists a stress matrix Ω such that Ω is positive semidefinite and $rank(\Omega) = n - d - 1$.

Also in "S.zhao", it shows the desired trajectories of each leaders can be expressed as

$$p_i^*(t) = A(t)r_i + b(t)$$

where r_i is the nominal configuration of each agents. With this time varying desired trajectory information of each agents, trajectory tracking control algorithm is designed fo the leaders' formation. It also introduces the leader-follower strategy that utilizes above stress matrix and leaders positions to calculate the desired trajectories of followers. It shows that the number of leaders selected in a dimension d in order to be affinely localizable should be d+1. Also it shows that $\lim_{t\to\infty} p_f = -\bar{\Omega}_{ff}^{-1}\bar{\Omega}_{fl}p_l$ where

$$\Omega = \begin{bmatrix} \Omega_{ll} & \Omega_{lf} \\ \Omega_{fl} & \Omega_{ff} \end{bmatrix}$$

, where Ω_{ff} has to be non-singular. Therefore it suggests the number of leaders in dimension d and contraint on stress matrix in order to find convergable route for followers. Then it utilizes this desired positions of the leaders and followers to build the control objectives for leaders and followers, respectively.

$$\lim_{t\to\infty} p_l - p_l^* = 0$$
$$\lim_{t\to\infty} p_f - p_f^* = p_f - \bar{\Omega}_{ff}^{-1}\bar{\Omega}_{fl}p_l = 0$$

The formulation of this problem heavily builds upon the "S.zhao" paper. However, this paper "Distributed leader-follower" is novel in a way that it builds upon the original paper with higher order so that it can render fully distributed control strategy for the followers and achieve autonomous manuever for group of leaders.

III. Analysis

Tracking Control Algorithm for First Leader

Assuming that the trajectory and its finite time derivative information is already given, the paper designes tracking control algorithm for leaders. First it uses a finite time backstepping approach to achieve $lim_{t\to\infty} p_1 - p_1^* = 0$ for the first leader. Using the existing control objectives as auxiliary variable and a virtual variable that is to be determined later, auxiliary variable can be designed as such,

$$Z_{11} = p_1 - p_1^*$$
$$Z_{1i} = p_1^{i-1} - \alpha_{1(i-1)}(t)$$

. Then, if we can determine $\alpha_{1(i-1)}(t)$ and know all the time derivative information of itself, then we would be able to design the tracking control algorithm for the first leader. By using above equations, the paper achieves the relation between auxiliary variables in higher order.

$$\dot{Z}_{11} = \dot{p}_1 - \dot{p}_1^* = Z_{12} - \alpha_{11} - \dot{p}_1^*$$
$$\alpha_{11}(t) = \dot{p}_1^* - k_{11}sig^{\beta}(Z_{11}) - k_12Z_{11}$$

, where

$$sig^r(x) = [sig^r(x_1), sig^r(x_2), \cdots sig^r(x_n)]$$
$$sig^r(x_i) = sgn(x_i)|x_i|^r$$

and $sgn(x_i) = \begin{cases} -1 & x_i < 0 \\ 0 & x_i = 0 \\ 1 & x_i > 0 \end{cases}$. Also $k_{11}$and k_{12} are positive control gains, and $0 \leq \beta \leq 1$. Therefore, substituting above α_{11} equation to the $\dot{Z}_{11}$ equation, yields

$$\dot{Z}_{11} = Z_{12} - k_{11}sig^{\beta}(Z_{11}) - k_{12}Z_{11}$$

Repeating the same procedure for Z_{12} and continuing on Z_{1i},

$$\begin{aligned}
&Z_{12} = p_2^{(1)} - \alpha_{11}(t)\\
&\dot{Z}_{12} = p_2^{(2)} - \dot{\alpha}_{11} = Z_{13} + \alpha_{12} - \dot{\alpha}_{11}\\
&\alpha_{12} = \dot{\alpha}_{11}\dot{Z}_{11} + \ddot{p_1^*} - k_{11}sig^{\beta}(Z_{12}) - k_{12}Z_{12} - Z_{11}\\
&Z_{1i} = p_1^{i-1} - \alpha_{1(i-1)}(t)\\
&\dot{Z}_{1i} = Z_{1(i+1)} - k_{11}sig^{\beta}(Z_{1i}) - k_{12}Z_{1i} - Z_{1(i-1)}\\
&\alpha_{1i} = \dot{\alpha}_{1i}\dot{Z}_{1i} - k_{11}sig^{\beta}(Z_{1i}) - k_{12}Z_{1i} - Z_{1(i-1)}\\
&\dot{Z}_{1n} = -k_{11}sig^{\beta}(Z_{1n}) - k_{12}Z_{1n} - Z_{1(n-1)}u_i = \alpha_{in}
\end{aligned}$$

. Finally, it utilizes the nth order intergrator α_{1n} for the control inputs and its stability is verified with sum of quadratic form of Lyapunov functions.

$$V_{1n} = \frac{1}{2}(Z_{11}^T Z_{11} + Z_{12}^T Z_{12} + \cdots + Z_{1n}^T Z_{1n})$$
$$\dot{V}_{1n} = -k_{11}Z_{11}^T sig^{\beta}(Z_{11}) - \cdots - k_{1n}Z_{1n}^T sig^{\beta}(Z_{1n}) - 2k_{12}V_{1n}$$

Then,

$$\begin{aligned}
Z_{1i}^T sig^{\beta}(Z_{1i}) &= sgn(Z_{1i1})|Z_{1i1}|sgn(Z_{1i1})|Z_{1i1}|^{\beta} + \cdots\\
&\quad + sgn(Z_{1in})|Z_{1in}|sgn(Z_{1in})|Z_{1in}|^{\beta}\\
&= \sum_{k\in dim} |Z_{1ik}|^{1+\beta}
\end{aligned}$$

$$||Z_{1i}||^{\beta+1} = (\sum_{k\in dim} Z_{1ik}^2)^{\frac{\beta+1}{2}} \leq \sum_{k\in dim}(Z_{1ik}^2)^{\frac{\beta+1}{2}} (\because 0 \leq \beta \leq 1)$$
$$\therefore Z_{1i}^T sig^{\beta}(Z_{1i}) = \sum_{k\in dim} |Z_{1ik}|^{1+\beta} \geq ||Z_{1i}||^{\beta+1}$$

$$\begin{aligned}
||Z_{1i}||^{\beta+1} &= ||2(V_{1i} - V_{1(i-1)})||^{\beta+1}\\
&= 2^{\frac{\beta+1}{2}}(V_{1i} - V_{1(i-1)})^{\frac{\beta+1}{2}}
\end{aligned}$$

Therefore

$$\begin{aligned}
\dot{V}_{1n} &\leq -2^{\frac{\beta+1}{2}}k_{11}(V_{11} + \sum_{i=2}^{n}(V_{1i} - V_{1(i-1)})^{\frac{\beta+1}{2}}) - 2k_{12}V_{1n}\\
&\leq -2^{\frac{\beta+1}{2}}k_{11}V_{1n}^{\frac{\beta+1}{2}}
\end{aligned}$$

. Here we know that $0 \leq \beta \leq 1$, and k_{11} and V_{1n} are positive. Therefore $\dot{V}_{1n}$ is negative definite and $\lim_{t\to\infty} p_1 - p_1^* = 0$

Tracking Control Algorithm for Second Leader Group

Before designing control algorithm for second leader group, it needs to have desired trajectory and time derivative information of those second leader group. To estimate those information in a distributed way, the second leader group utilizes the desired trajectory and its nth order derivatives of the first leader to generate desired trajectory and the time derivatives of second leader group. Since we know the desired trajectory,$b(t)$, and time derivatives of the first leader,$b^{(j)}(t)$, where $j = 0, 1, \cdots, n$, desired trajectory and time derivatives of second leader group can be estimated with

$$\dot{\hat{b}}_i^{(j)} = -\rho_j sgn[\sum_{k=1}^{M} sgn(\omega_{ik})(\hat{b}_i^{(j)} - \hat{b}_k^{(j)})]$$

, where $i \in 2, \cdots, M$ and ρ_j is positive constant. Also the time derivative information of A(t) can be estimated in a distributed way.

$$\dot{\hat{a}}_i^{(j)} = -\rho_j sgn[\sum_{k=1}^{M} sgn(\omega_{ik})(\hat{a}_i^{(j)} - \hat{a}_k^{(j)})]$$

, where $\hat{a}_i^{(j)} \in R^{d^2}$ is agent i's estimation of a(t), and a(t) is column vector form of A(t).

Then combining those two estimates generates desired trajectory and time derivative information of each agents.

$$p_i^*(t) = A(t)r_i + b(t)$$

. With this generated desired trajectory of each agents, again, finite time backstepping appoarch is used to generate the control inputs of each agents.

$$\begin{aligned}
Z_{i1} &= p_j - p_j^* = Z_{i2} - \alpha_{i1} - \dot{p}_j^* \\
\alpha_{i1}(t) &= \dot{p}_j^* - k_{i1} sig^{\beta}(Z_{i1}) - k_j 2 Z_{i1} \\
Z_{ik} &= p_1^{k-1} - \alpha_{1(k-1)}(t) \\
\alpha_{ik} &= \dot{\alpha}_{ik}\dot{Z}_{ik} - k_{11} sig^{\beta}(Z_{ik}) - k_{12} Z_{ik} - Z_{1(i-1)} \\
u_i &= \alpha_{in}
\end{aligned}$$

PDm Control Algorithm for Followers with Constant Gains

The author was inspired from "Containment Control of Multiagent Systems With Dynamic Leaders Based on a PIn-Type Approach" and proposed PDm control algorithm for followers. Using all the desired trajectory information and their time derivative information of neighbors of itself, the agent is able to generate adaptive gains using offline computation. The author proposes control algorithm for followers

$$\begin{aligned}
\dot{u}_i = -\eta_i \sum_{j \in V_l \bigcup V_f} \omega_{ij}[k_0(p_i - p_j) + k_1(\dot{p}_i - \dot{p}_j) \\
+ \cdots + k_m(p_i^{(m)} - p_j^{(m)})]
\end{aligned}$$

. Using the dynamics of leaders and followers and their compact form,

$$\begin{aligned}
\dot{x}_i &= (B \otimes I_d)x_i, i \in V_l \\
\dot{x}_i &= (B \otimes I_d)x_i + (C \otimes I_d)u_i, i \in V_f \\
\dot{x}_l &= (I_M \otimes B \otimes I_d)x_l \\
\dot{x}_f &= (I_{n_f} \otimes B \otimes I_d)x_f + (\Lambda\Omega_{ff} \otimes C \otimes I_d)x_f \\
&\quad - (\Lambda\Omega_{fl} \otimes CK \otimes I_d)x_l
\end{aligned}$$

where $x_i = [p_i^T, p_i^{(1)T}, m\cdots, p_i^{(m)T}]^T \in R^{d(m+1)\times 1}$, and

$$\begin{aligned}
B &= \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ 0 & 0 & \cdots & 0 \end{bmatrix} \\
C &= \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix}^T \\
\Lambda &= diag[\eta_{M+1}, \cdots, \eta_N] \in R^{n_f \times n_f}
\end{aligned}$$

the tracking error can be defined as

$$X_f(t) = x_f(t) - x_f^*(t) = x_f + [\Omega_{ff}^{-1}\Omega_{fl} \otimes I_{(m+1)d}]x_l$$

. From this tracking error equation, η_i will be designed to let this tracking error to converge to zero. By taking the derivative of it,

$$\begin{aligned}
\dot{X}_f(t) &= \dot{x}_f + [\Omega_{ff}^{-1}\Omega_{fl} \otimes I_{(m+1)d}]\dot{x}_l \\
&= (I_{n_f} \otimes B \otimes I_d)x_f + (\Lambda\Omega_{ff} \otimes C \otimes I_d)x_f \\
&\quad - (\Lambda\Omega_{fl} \otimes CK \otimes I_d)x_l \\
&\quad + [\Omega_{ff}^{-1}\Omega_{fl} \otimes I_{(m+1)d}](I_M \otimes B \otimes I_d)x_l \\
&= (I_{n_f} \otimes B \otimes I_d - \Lambda\Omega_{ff} \otimes CK \otimes I_d)X_f(t)
\end{aligned}$$

With this time derivative information, construct the Lyapunov function and take the derivative.

$$\begin{aligned}
V &= X_f^T(\Omega_{ff} \otimes P \otimes I_d)X_f \\
\dot{V} &= \dot{X}_f^T(\Omega_{ff} \otimes P \otimes I_d)X_f + X_f^T(\Omega_{ff} \otimes P \otimes I_d)\dot{X}_f
\end{aligned}$$

Using the above $\dot{X}_f(t)$ equation and Algebraic Riccati Equation,

$$\begin{aligned}
B^T P + PB - PCC^T P + I_{m+1} = 0 \\
\dot{X}_f(t) = (I_{n_f} \otimes B \otimes I_d - \Lambda\Omega_{ff} \otimes CK \otimes I_d)X_f(t) \\
\dot{V} = X_f^T(\Omega_{ff} \otimes (PB + B^T P) \otimes I_d)X_f \\
- 2X_f^T(\Omega_{ff}\Lambda\Omega_{ff} \otimes (PCC^T P) \otimes I_d)X_f \\
= -X_f^T[\Omega_{ff} \otimes I_{m+1} \otimes I_d]X_f + X_f^T[(\Omega_{ff} \\
- 2\Omega_{ff}\Lambda\Omega_{ff}) \otimes PCC^T P \otimes I_d]X_f
\end{aligned}$$

To prove stability of this Lyapunov function, $(\Omega_{ff} - 2\Omega_{ff}\Lambda\Omega_{ff}) \leq 0$, since $\Omega_{ff} \otimes I_{m+1} \otimes I_d \geq 0$ and $PCC^T P \geq 0$. If we choose η_i as

$$\min_{i \in V_f} \eta_i \geq \frac{1}{2\lambda_{min}(\Omega_{ff})}$$

, then

$$y^T(\Omega_{ff} - 2\Omega_{ff}\Lambda\Omega_{ff})y = -(\Omega_{ff}y)^T(2\Lambda - \Omega_{ff}^{-1})\Omega_{ff}y \leq 0,$$

for any $y \in R^{n_f}$.

PDm Control Algorithm for Followers with Adaptive Gains

With constant control gain larger than $\frac{1}{2\lambda(\Omega_{ff})}$, the trajectory would be depending on the formation of the graph. So the author suggests using adaptive gains with Algebraic Riccati Equation.

$$\begin{aligned}
\dot{u}_i &= -\eta_i \sum_{j \in V_l \bigcup V_f} \omega_{ij}[k_0(e_{ij}) + k_1(e_{ij}^{(1)}) + \cdots + k_m(e_{ij}^{(m)})] \\
\dot{\hat{\eta}}_i &= \gamma_i s_i^T(PCC^T P \otimes I_d)s_i \\
s_i &= \sum_{j \in V_l \bigcup V_f} \omega_{ij}(x_i - x_j)
\end{aligned}$$

.

Considering the previous Lyapunov function and building upon it,

$$
\begin{aligned}
V_1 &= X_f^T(\Omega_{ff} \otimes P \otimes I_d)X_f \\
V_2 &= V_1 + \sum_{i=M+1}^{N} (\hat{\eta}_i(t) - \bar{\eta})^2/\gamma_i \\
\dot{V}_1 &= -X_f^T[\Omega_{ff} \otimes I_{m+1} \otimes I_d]X_f + X_f^T[(\Omega_{ff} \\
&\quad - 2\Omega_{ff}\Lambda\Omega_{ff}) \otimes PCC^TP \otimes I_d]X_f \\
&= -X_f^T[\Omega_{ff} \otimes I_{m+1} \otimes I_d]X_f \\
&\quad + X_f^T[\Omega_{ff} \otimes PCC^TP \otimes I_d]X_f \\
&\quad - 2[\Omega_{ff} \otimes I_{(m+1)d}X_f]^T[\hat{\Lambda}(t) \otimes PCC^TP \otimes I_d] \\
&\quad \times [(\Omega_{ff} \otimes I_{(m+1)d})X_f] \\
\dot{V}_2 &= \dot{V}_1 + \sum_{i=M+1}^{N} 2(\hat{\eta}_i(t) - \bar{\eta})[s_i^T(PCC^TP \otimes I_d)s_i]
\end{aligned}
$$

Substituting $\dot{V}_1$ into $\dot{V}_2$ equation achieves,

$$
\begin{aligned}
\therefore \dot{V}_2 &= -X_f^T[\Omega_{ff} \otimes I_{m+1} \otimes I_d]X_f \\
&\quad + X_f^T[\Omega_{ff} \otimes PCC^TP \otimes I_d]X_f \\
&\quad - 2\bar{\eta} \sum_{i=M+1}^{N} s_i^T(PCC^TP \otimes I_d)s_i \\
&= -X_f^T[\Omega_{ff} \otimes I_{m+1} \otimes I_d]X_f \\
&\quad + X_f^T[\Omega_{ff} \otimes PCC^TP \otimes I_d]X_f \\
&\quad - 2\bar{\eta}X_f^T[\Omega_{ff}\Omega_{ff}(PCC^TP \otimes I_d)]X_f.
\end{aligned}
$$

Applying the same logic that was applied in previous section, to obtain the stability$\dot{V}_2 \leq 0$,

$$
\begin{aligned}
2\bar{\eta}\Omega_{ff}\Omega_{ff} - \Omega_{ff} &\geq 0 \\
\therefore \bar{\eta} &\geq \frac{1}{2\lambda_{min}(\Omega_{ff})}
\end{aligned}
$$

IV. Simulation

Manevuer Design

Before simulating, the desired trajectory has to be generated according to the given desired position and velocity value at time t_k and t_{k+1}. With the previous assumption that desired trajectory,$p_i = A(t)r_i + b(t)$, is known and n^{th} order differentiable, the paper assumes 3^{rd} order time differentiable equation,

$$
b(t) = c_3(t - t_k)^3 + c_2(t - t_k)^2 + c_1(t - t_k) + c_0,
$$

where $c_i \in R^2$ are constant vectors at $t \in [t_k, t_{k+1}]$ for all the leader positions. With known desired position and velocity profile at two different given time, then the desired trajectory can be found with

$$
\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & \Delta T & \Delta T^2 & \Delta T^3 \\ 0 & 1 & \Delta T & 3\Delta T^2 \end{bmatrix}^{-1} \otimes I_2 \begin{bmatrix} S_{t_k} \\ v_{t_k} \\ S_{t_{k+1}} \\ v_{t_{k+1}} \end{bmatrix}
$$

where $S_{t_k}, v_{t_k}, S_{t_{k+1}}, v_{t_{k+1}}$ are the desired position and velocity profile at t_k and t_{k+1}. Also the scaling maneuver is designed with,

$$
\begin{aligned}
A(t) &= \phi(t)I_2 \\
\phi(t) &= \varsigma_3(t - t_k)^3 + \varsigma_2(t - t_k)^2 + \varsigma_1(t - t_k) + \varsigma_0,
\end{aligned}
$$

where $\varsigma_3, \varsigma_2, \varsigma_1, \varsigma_0$ are known desired scaling factor at t_k and t_{k+1}. With the same matrix equation used for translational maneuver, scaling maneuver constants,$\varsigma_i \in R, i = 0, 1, \cdots 3$, can be also be inferred. Also the rotational and shearing design is described in a same manner, where

$$
\begin{aligned}
A_{rot}(t) = R(\theta(t)) &= \begin{bmatrix} cos\theta(t) & -sin\theta(t) \\ sin\theta(t) & cos\theta(t) \end{bmatrix} \\
A_{she}(t) &= \begin{bmatrix} \psi_1(t) & 0 \\ 0 & \psi_2(t) \end{bmatrix}
\end{aligned}
$$

are rotational and shearing matrix for each design, and $\theta(t) \in [0, 2\pi)$ and $\psi(t) \in (0, 1)$ are rotation angle with respect to the nominal configuration and shearing factors respectively.

V. Conclusion