Fully Distributed Flocking with a Moving Leader for Lagrange Networks with Parametric Uncertainties

Paul F. Roysdon

July 2, 2015

1 Introduction & Motivation

This project is motivated by nonlinear problems which arise from leader-follower flocking, of which we investigate three cases: a leader with constant velocity, a leader with variable velocity, and a leader with variable velocity with fully distributed flocking. This multi-agent network applies knowledge of each agents' kinematics and neighboring agent kinematics (one-hop and two-hop), to maintain the formation and prevent collisions.

2 Problem description

Implementation is based on [1] and [2] (applying our *apriori* knowledge of [3]) which outline the distributed continuous adaptive control algorithm and estimator. We applied [1] to an initial network containing six agents, with two additional agents joining the network at the mid-point of the trajectory. This multi-agent network was demonstrated for each of the following algorithms:

1. Leader with constant velocity, [1] eqns. (4-7)
2. Leader with variable velocity, [1] eqns. (17-20)
3. Leader with variable velocity with fully distributed flocking, [1] eqns. (24-28)

3 Experimental Setup, Results & Analysis

Simulation was performed in Matlab & Simulink v8.3 R2014a, requiring only the basic install of Matlab toolboxes and Simulink blocksets. While initialization was performed in Matlab, modeling of the potential function and flock was performed in Embedded Matlab Function blocks within Simulink. We used a variable step size, ODE45 solver, with a tolerance of 1e-3, and saved the "yout" and "tout" outputs to the Matlab workspace and a ".mat" file for later analysis and animation. A "README.txt" file is provided with the simulation source code, which describes the computer system requirements and basic operation of the simulation.

The initial states of the followers are chosen as $q_1(0) = [2, 6]^T$, $q_2(0) = [4, 6]^T$, $q_3(0) = [6, 6]^T$, $q_4(0) = [6, 4]^T$, $q_5(0) = [6, 2]^T$, $\dot{q_1} = [0, 0.5]^T$, $\dot{q_2} = [0.5, 0]^T$, $\dot{q_3} = [0, -0.5]^T$, $\dot{q_4} = [-0.5, 0]^T$, $\dot{q_5} = [-0.5, 0.5]^T$, $i = 1, ..., 5$. and the leaders trajectory is chosen as $q_0(0) = [8, 8]^T$ and $\dot{q_0} = [1, 0.5]^T$. The initial values for the estimates of the leaders velocity are $v_i(0) =$

$[0.5, 0.5]^T$, $i = 1, ..., 5$. The control parameter is chosen as $\Gamma_i = 1 \times I_2$, $i = 1, ..., 5$, where $R = 8$ and $d = 1$.

For continuity of this report, all figures are located in Section 5, and movie animations, both 2D and 3D are supplied as attachments with this report.

3.1 Algorithm I

In the first part, we simulate the case where the leader has a constant velocity under the control algorithm (4)-(7) in [1].

Fig. 5.1(a) shows the trajectories of the leader (denoted as a red diamond) and the followers (denoted as green circles). Clearly, all followers move cohesively with the leader without colliding with each other. Fig. 5.1(b) shows the velocity of the followers and the leader. It can be seen that the velocities of the followers converge to that of the leader and all agents move with the same velocity.

Figures 5.2(a)&(b) show the trajectories and velocities of the leader and followers. In this case, we see that flock Agent 1 (denoted as a blue square) is outside the defined radius of the other flock Agents, and clearly departs, un-influenced by the flock. Flock Agent 1 is instead influenced by its own initial conditions, achieving a stead-state velocity independent of the flock.

3.2 Algorithm II

In the second part, we simulate the case where the leader has a varying velocity under the control algorithm (17)-(20) in [1]. The initial states of the leader and followers are chosen as above. The control parameters are chosen as $\alpha = 1$, $\gamma = 50$, $\beta = 0.1$, and $\Gamma_i = 1 \times I_2$, $i = 1, ..., 5$.

Fig. 5.3(a) shows the trajectories of the followers and the leader. The agents maintain the initial connectivity while avoiding collisions with a sinusoidal velocity. Fig. 5.3(b) shows that each follower moves with the same velocity as the leader.

Fig. 5.4(a) shows the trajectories of the followers and the leader with a single departing agent (again Agent 1). This case clearly shows that Agent 1 re-joins the flock once it is within the defined radius ($R = 8$). Fig. 5.4(b) shows that each follower moves with the same velocity as the leader, and Agent 1 quickly matches the flock velocity once it is within range at time=33 seconds.

Fig. 5.5(a) demonstrates that the initial velocity convergence for the flock occurs in less than 0.01 seconds. While, Fig. 5.5(b) clearly shows that once Agent 1 is within range of the flock, the entire flock adjusts their respective velocities to include Agent 1, eventually converging again with the leader velocity over a period of 0.2 seconds. This is a very important result as it demonstrates the connectivity and velocity matching of algorithm II.

3.3 Algorithm III

In the third part, we simulate the case where the leader has a varying velocity under the fully distributed control algorithm (24)-(28) in [1]. Here the initial states and the leaders trajectory are chosen as in the second part. The control parameter is chosen as $\Gamma_i = 1 \times I_2$, $i = 1, ..., 5$.

Fig. 5.6(a) shows the trajectories while Fig. 5.6(b) shows the velocities of the followers and the leader. The leader-following flocking is achieved in this case.

Fig. 5.7(a)&(b) show the trajectories velocities for the single agent departure case (again Agent 1). Fig. 5.7(b) shows that each follower moves with the same velocity as the leader, and Agent 1 momentarily re-joins the flock, quickly matching the flock velocity once it is within range at time=33 seconds.

Fig. 5.8(a) demonstrates that the initial velocity convergence for the flock occurs in less than 1.2 seconds. While, Fig. 5.8(b) clearly shows that once Agent 1 is within range of the flock, the entire flock adjusts their respective velocities to include Agent 1, eventually converging again with the leader velocity. However, it is important to note that because Agent 1 did not converge quickly enough, it then leaves the flock, because it is momentarily outside of the defined radius, and thus maintains its most recent adjusted velocity (due to flock velocity matching) as a constant. This is a very important result as it demonstrates the connectivity and velocity matching of algorithm III.

4 Conclusion

In this short evaluation of the distributed leader-follower flocking problem, it was shown that the algorithms achieve connectivity maintenance, collision avoidance, and velocity matching, where only the neighbors information was used in the control design. Animation movies of these results, are attached with this report. In the 3D animation we used an AH-64 helicopter instead of other available models (jet, quadcopter, etc.) because it better conveys the orientation change over the trajectory, as the helicopter blades create a nice local x-y axis.

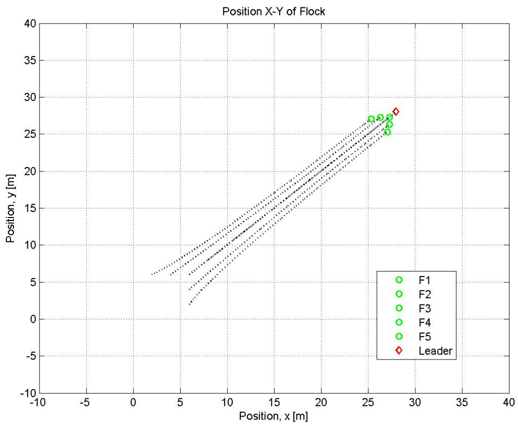
References

[1] S.Ghapani, J.Mei, W.Ren, *Fully Distributed Flocking with a Moving Leader for Lagrange Networks with Parametric Uncertainties*, Automatica, pre-publish. 2015.

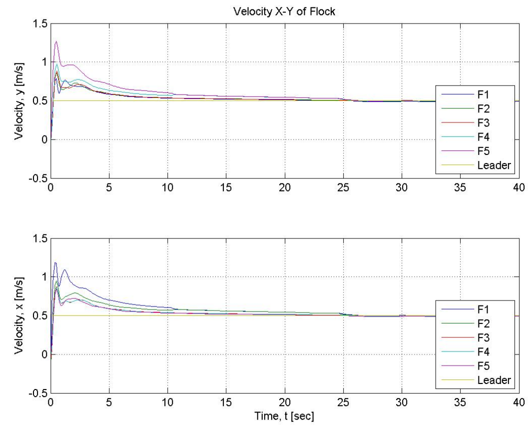
[2] Y.Cao, W.Ren, *Distributed Coordinated Tracking With Reduced Interaction via a Variable Structure Approach*, IEEE Transactions on Automatic Control, Vol. 57, No. 1, January 2012.

[3] H.J.Marquez, *Nonlinear Control Systems*, Wiley Publication, 2003.

5 Appendix

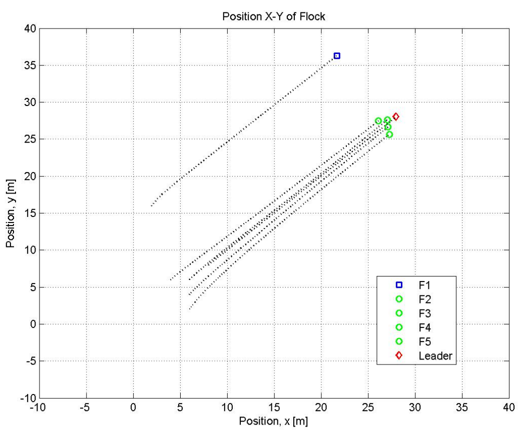


(a)

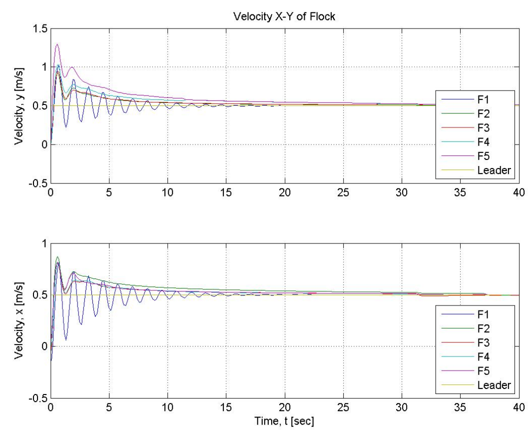


(b)

Figure 5.1: Algorithm I: trajectories (a), velocities (b)

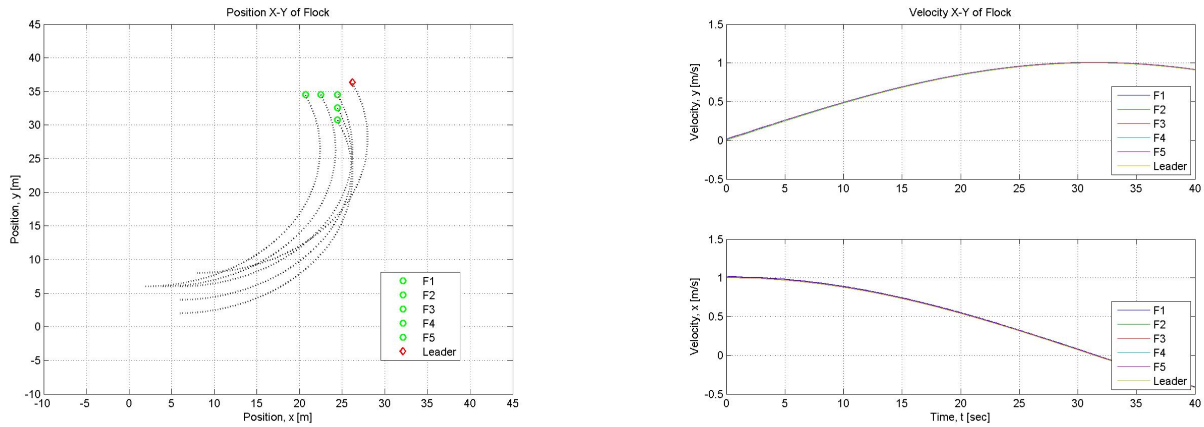


(a)



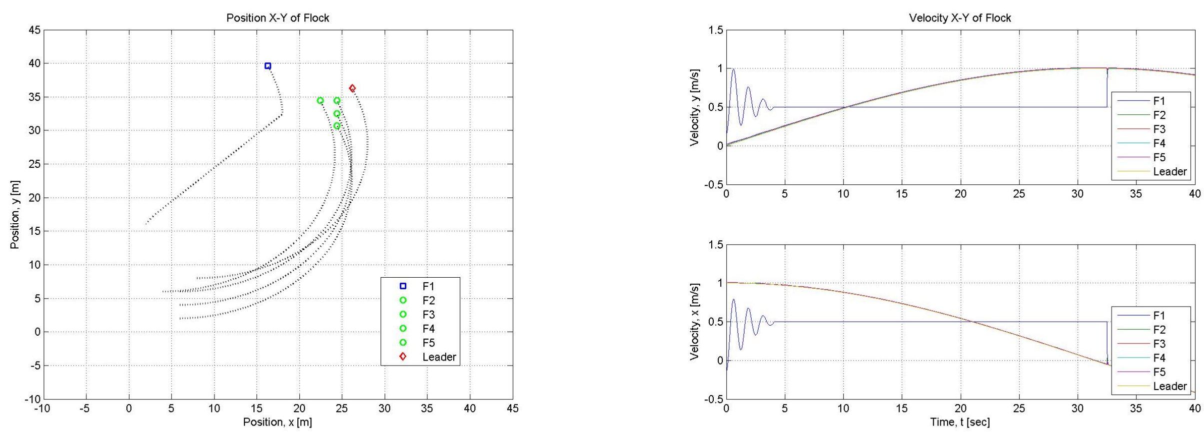
(b)

Figure 5.2: Algorithm I *departure*: trajectories with 1 departing agent (a), velocities (b)



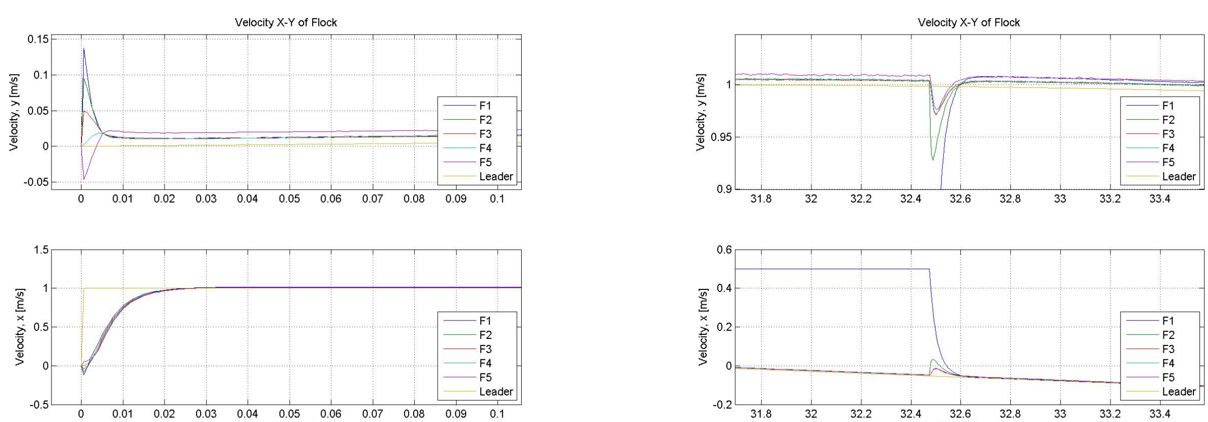
(a) (b)

Figure 5.3: Algorithm II: trajectories (a), velocities (b)



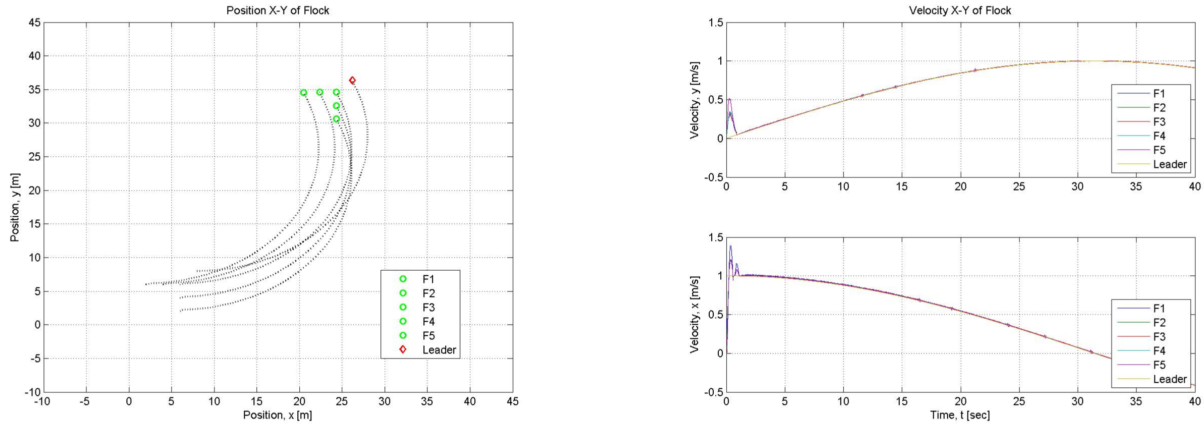
(a) (b)

Figure 5.4: Algorithm II *departure*: trajectories with 1 departing agent (a), velocities (b)



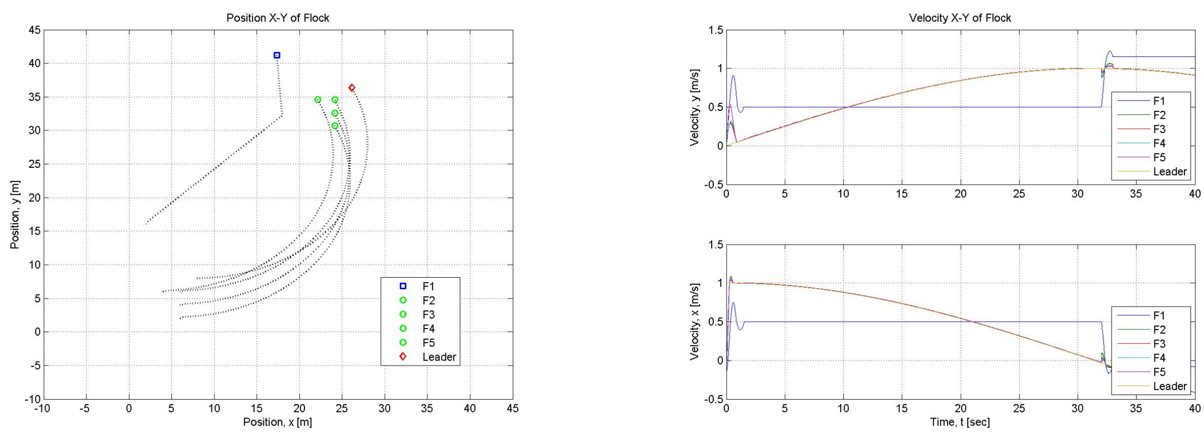
(a) (b)

Figure 5.5: Algorithm II *convergence*: initial convergence (a), Agent 1 joining the flock (b)



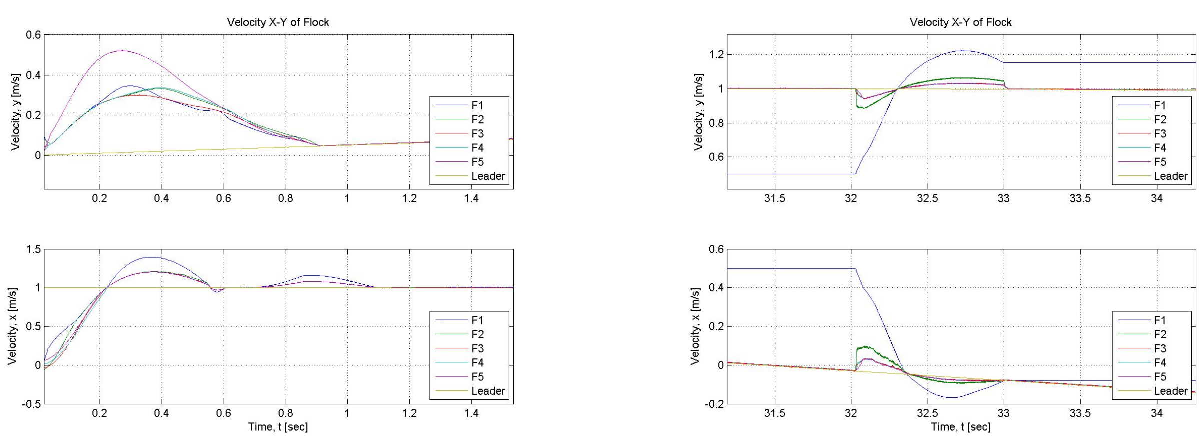
(a) (b)

Figure 5.6: Algorithm III: trajectories (a), velocities (b)



(a) (b)

Figure 5.7: Algorithm III *departure*: trajectories with 1 departing agent (a), velocities (b)



(a) (b)

Figure 5.8: Algorithm III *convergence*: initial convergence (a), Agent 1 joining flock (b)