Bio-Inspired Position Control of Satellite Constellations

Michael Rubenstein[1] and Zachary Manchester[2]

[1] Northwestern University, Evanston, IL 60201, USA
rubenstein@northwestern.edu
[2] Stanford University, Stanford, CA 94305, USA
zacmanchester@stanford.edu

Abstract. We present a biologically inspired controller for creating formations in satellite swarms. This controller can place satellites in formation where there is equal spacing between individuals, or place them all in the same location. The controller is fully decentralized without any human control. It only relies on simple satellite capabilities such as light sensing, neighbor communication, and attitude control, enabling it to run on the simplest of satellites. We present the controller and demonstrate its operation in a realistic simulation environment.

1 Introduction

Progress in consumer electronics, particularly the development of smartphones, has had a dramatic impact on the aerospace industry. Many of the components needed onboard spacecraft are now available in very small packages at low cost. This trend is behind the success of kilogram-scale CubeSats [10], as well as the emergence of even smaller spacecraft like 100 gram PocketQubes and gram-scale ChipSats like the Sprite [7] (Fig. 1).

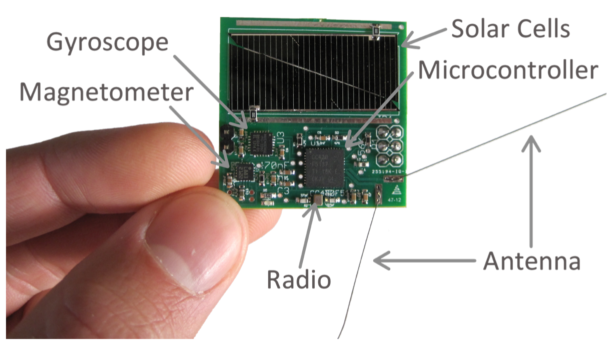

Fig. 1. The Sprite spacecraft.

Over the past decade, these small satellites have gone from university research projects to the foundation of a new industry promising to deliver imagery [11], synthetic aperture radar [1], and data backhaul for tiny "Internet of Things" devices [5] with near-continuous coverage of the entire Earth. To achieve this coverage, several companies are either planning or currently launching constellations of hundreds [11] to thousands [8, 6] of satellites.

Deploying and managing a large constellation poses serious challenges. Typically, many spacecraft are deployed into the same orbit from a single launch vehicle and must then be "phased" into desired relative positions along that orbit. Satellite operators currently rely heavily on tracking and centralized planning and control performed on the ground to accomplish these tasks [3]. Reducing reliance on ground-based infrastructure and increasing spacecraft autonomy will be key to keeping hardware and operating costs low as commercial spacecraft constellations continue to expand.

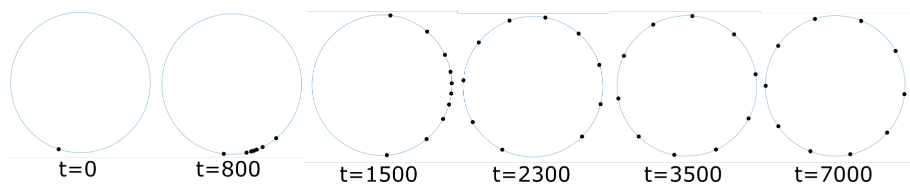

Fig. 2. Overview of desired satellite behavior in simulation. Satellites start in same position and orbit and use drag-based control to move to equal spacing. Time labeled in hours.

Inspired by distributed robotics and sensor networks, we propose a decentralized method to manage the position of a large constellation of satellites, easily scalable to large numbers. This method will take a swarm of satellites and place them in an equally spaced constellation (Fig. 2), one of the most common satellite formations. It is also capable of the opposite: taking dispersed satellites and bringing them to the same point in space. This method is fully autonomous and does not require any complex sensing, such as GPS or ground-based tracking, so it is possible to implement on swarms of thousands of gram-scale ChipSats like the Sprite [7].

The paper proceeds as follows: Section 2 reviews the basics of satellite dynamics, formation control. Sections 3 and 4 introduce the novel distributed control methodology. Section 5 then presents the results of numerical simulations demonstrating the proposed control law. Finally, section 6 summarizes our conclusions and directions for future research.

2 Background

2.1 Orbital Dynamics

Orbital dynamics in low-Earth orbit are dominated by two forces: gravity and atmospheric drag. To a good first approximation, the Earth's gravitational field is uniformly spherical and obeys the inverse-square law,

$$F_G = \frac{-\mu m}{\|\boldsymbol{r}\|^3}\boldsymbol{r}, \tag{1}$$

where $\mu \approx 3.986 \times 10^{14}$ m^3/s^2 is the "standard gravitation parameter" for the Earth, m is the mass of the spacecraft, and r is the spacecraft's position vector measured from the center of the Earth.

Atmospheric drag obeys the equation,

$$F_D = -\frac{1}{2}\rho C_D A \|\boldsymbol{v}\| \boldsymbol{v}, \tag{2}$$

where ρ is the atmospheric density, C_d is the drag coefficient, typically taken to be 2.2 for spacecraft, A is the projected area normal to the velocity vector, and $\boldsymbol{v}$ is the spacecraft's velocity vector relative to the atmosphere. We use a simple isothermal exponential model for atmospheric density as a function of altitude,

$$\rho = \rho_0 e^{h/H}, \tag{3}$$

where ρ_0 is the surface density, h is the altitude measured from the Earth's surface, and H is a constant known as the atmosphere's "scale height."

2.2 Spacecraft Attitude Control

Most spacecraft have some means of controlling their attitude (orientation). This can range from passive solutions like spin stabilization to active closed-loop control with thrusters. Most small satellites use a combination of magnetic torque coils, which are essentially electromagnets that can be turned on and off to exert torques against the Earth's magnetic field, and reaction wheels, which are internal flywheels that can be rotated in one direction to cause the spacecraft body to rotate in the opposite direction due to conservation of angular momentum. Throughout this paper, We will assume that full three-axis attitude control is possible on timescales much shorter than the orbital period of the spacecraft.

2.3 Drag-Based Formation Control

As is well known, a spacecraft in an inverse-square-law gravitational field moves along an elliptical orbit in the absence of other perturbing forces. Atmospheric drag tends to gradually reduce the the size of an orbit and increase the spacecraft's velocity. Since the projected area, A, in equation (2) depends on the spacecraft's orientation relative to the oncoming flow, it is possible to modulate the drag force by controlling the spacecraft's attitude. This is an extremely useful source of orbit actuation since most small spacecraft do not carry propulsion systems for cost and safety reasons.

Drag modulation has been used for initial phasing and long-term formation-keeping of the CubeSat constellation deployed by Planet Labs [4]. Their approach to formation control is based on a linearization of the orbital dynamics known as the Clohessy-Wiltshire equations [12]:

$$\frac{d}{dt}\begin{bmatrix} r \\ \theta \\ \dot{r} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 3\omega_0^2 & 0 & 0 & 2\omega_0 r_0 \\ 0 & 0 & -2\omega_0/r_0 & 0 \end{bmatrix} \begin{bmatrix} r \\ \theta \\ \dot{r} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1/(r_0 m) \end{bmatrix} F_D. \tag{4}$$

These equations, expressed in the orbital plane using polar coordinates, describe motion relative to a nominal circular orbit of radius r_0 and angular frequency ω_0.

Treating $u = F_D$ as a control input and assuming it can be varied between u_{min} and u_{max} by altering the spacecraft's attitude, the time-history of control commands for each spacecraft can be found by solving a convex quadratic program [4]. This approach suffers from two main drawbacks: First, very accurate knowledge of each spacecraft's position is needed to disambiguate individual members of the constellation [3]. And second, control commands for the entire constellation are computed centrally on the ground.

3 Distributed Control Law

For the control laws developed in this paper, we assume that each satellite in the constellation has the following capabilities:

1. The ability to modulate its atmospheric drag
2. The ability to sense its passage into eclipse (Earth's shadow)
3. Communication with its nearest neighbors (leader and follower)

We use these assumed abilities to create two behaviors useful for formation control: drag-based position control and relative position sensing established using Earth-shadow crossing times.

3.1 Drag-Based Satellite Control

In one method for relative position control, the satellites modify their drag to adjust their speed and position relative to their neighbors. For example, the Sprite pictured in Fig. 1 can orient itself so that it is facing the direction of motion edge-on for low drag, or face-on for high drag. The satellites have a continuous range of drag between u_{min} and u_{max}, which is achieved by allowing a continuous range of orientations between edge-on and face-on. Due to the somewhat counter-intuitive nature of orbital dynamics, when a satellite increases its drag, the interaction with the atmosphere will lower its orbit, and increase its speed. For two satellites in the same orbit, a leader and a follower, to move closer to each other, the follower increases its drag, which reduces its altitude and increases its speed relative to the leader. Similarly, to increase the distance between the two, the leader increases its drag, increasing its speed and moving it farther from the follwer.

Since the satellites we are considering have drag-only control, they can never increase their altitude or slow down, which makes formation keeping somewhat complicated. To allow for motion toward the rear neighbor, we introduce a neutral drag u_0, which is between u_{min} and u_{max}. Satellites keep track of their total accumulated drag impulse J, defined as,

$$J = \int_0^T u(t)\ dt, \tag{5}$$

and are required to keep their own J within some constant value ΔJ of J_0, defined as,

$$J_0 = \int_0^T u_0(t)\ dt, \tag{6}$$

drag (i.e. $-\Delta J \leq (J - J_0) \leq \Delta J$). This then allows a satellite to move closer to its follower by setting its drag lower than u_0.

A demonstration of this drag-based control concept is illustrated in figure 3, where a leader and follower satellite move ahead and behind, respectively, from a satellite that maintains $u = u_0$. Once these satellites have moved to a desired distance from the center satellite, they adjust their drag so their total drag impulse returns to J_0, causing them to match the speed of the center satellite.

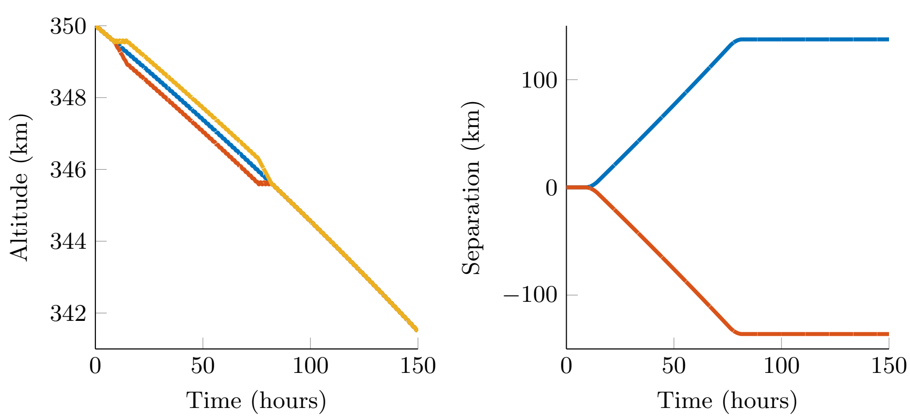

Fig. 3. Simulated drag based maneuvering demonstration for three satellites. (Left) The altitude of the satellites over time, where the neutral satellite is shown in blue, the leader is shown in red, and the follower is shown in yellow. (Right) the distance from the front and rear satellites to the center satellite.

3.2 Shadow-Crossing Position Sensing

To estimate the relative positions (orbital phases) of neighboring satellites, we make use of the time between each satellite's crossing into Earth's shadow. This event is easy to detect with a very simple low-cost sensor, such as a photodiode.

Since the orbital period of our satellites is small compared to the Earth's one year orbital period around the sun, we approximate the shadow crossing as a fixed point along the orbit.

The relative phase angle between satellites a and b along their orbit is given approximately by,

$$\theta_a - \theta_b \approx \frac{2\pi(\tau_b - \tau_a)}{P}, \tag{7}$$

where θ is phase angle, τ is crossing time, and P is the average orbital period. To compare shadow crossing times, satellites broadcast messages when they detect their own crossing, and compare that with the times they receive from other satellites. A satellites leader will have the nearest earlier crossing time, while its follower will have the nearest later crossing time.

4 Control Law

We now present a method to control a constellation of simple satellites that, when deployed from a single launch vehicle, can either 1) form and maintain a ring of equally spaced satellites along an entire orbit, or 2) form and maintain a cluster (Fig. 2).

The method of control is inspired by the work on synchronization of pulse-coupled biological oscillators by Strogatz et. al [9] and the work on desynchronization of wireless sensor networks by Nagpal et. al [2]. These works describe systems where distributed agents periodically "fire" a signal, such as a firefly flashing its light or a wireless sensor node transmitting a radio message, and use this signal to either synchronize [9] or desynchronize [2] these events across the group. This behavior can be visualized in a phase-space diagram as a collection of points, one for each agent, moving along a circle. An agent's position on the circle represents the current phase of its oscillator (Fig. 4). A "firing" location exists on the circle, and when the point crosses this location, the agent transmits a signal and adjusts its phase by jumping ahead or behind in phase based on signals it receives from other agents. The work presented in [9] and [2] demonstrate control methods that are guaranteed to move all agents to the same position on the circle (synchronize) or to positions equally spaced apart on the circle (desynchronize).

Key to the satellite control algorithm described in this paper is the analogy between the phase space of coupled oscillators and the physical position of satellites along their orbit (Fig. 4). Just as synch or desynch can create global outcomes where all agents are grouped together or equally spaced apart in a phase-space diagram, our proposed controller can create global outcomes where all satellites are grouped together or equally spaced apart along a physical orbit. In the satellite system, the "firing" signal occurs when the satellite detects it has crossed into Earth's shadow (its light sensor transitions from high to low). When a satellite detects this crossing, it "fires" by broadcasting a radio signal. We make the assumption that this signal reaches a satellite's leader and follower instantaneously. Based on the crossing times of itself and its nearest neighbors,

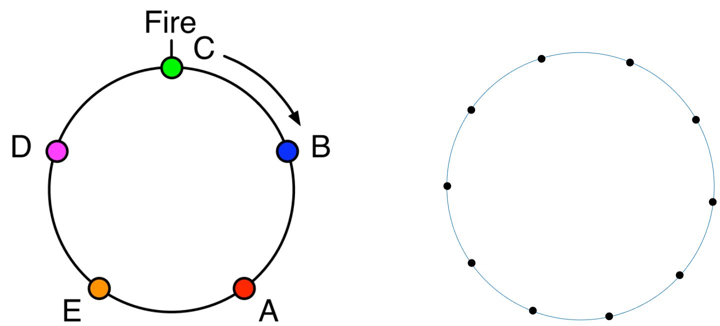

Fig. 4. Analogy of desynchronization of wireless sensor networks and satellite formation control. (Left) a modified figure taken from [2] showing phase diagram of desynchronized network. (Right) Position of 10 satellites in orbit using controller presented here.

the satellite adjusts its drag, analogous to how pulse-coupled oscillators adjust their phase to move towards or away from their neighbors in phase space.

4.1 Equal Spacing Control

To create an swarm of equally spaced satellites as shown in Fig. 2, the high-level idea is to have satellites move to a position that is equidistant between its leader and follower. This is the same controller behavior in [2] except instead of controlling the phase of oscillators, we are controlling the position of satellites. When this algorithm is run on all spacecraft, the end result is a ring of equally spaced satellites.

Assuming a satellite itself crosses at time τ, and detects the crossing times of its leader, τ_L, and follower, τ_F, then its desired crossing time is $(\tau_L + \tau_F)/2$. The error between a satellite's current orbital phase and its desired orbital phase is then:

$$e = \pi \left(\frac{2\tau - (\tau_L + \tau_F)}{P} \right) \tag{8}$$

To move to the desired phase, each satellite sets its drag using a PD control law:

$$u_{PD} = K_P e + K_D \dot{e}. \tag{9}$$

Additionally, a saturation operation is applied to u_{PD} to ensure than each satellite's value of J stays within ΔJ of J_0:

$$u = \begin{cases} u_0, & J > J_0 + \Delta J \\ u_0, & J < J_0 - \Delta J \\ u_{PD}, & \text{otherwise} \end{cases} \tag{10}$$

4.2 Clustering

To create a cluster of satellites, as shown in Fig. 2, a very similar controller can be used. Here, the high-level idea is to have satellites move to a position that is halfway between two points. The first point is its closest neighbor, i.e. $\tau_{min} = min(\tau_L, \tau_F)$ and the second point is the midpoint between leader and follower$(\tau_L + \tau_F)/2$. Therefore its desired crossing time is:

$$\tau_{desired} = \begin{cases} \tau_{min}/2 + (\tau_L + \tau_F)/4, & \text{if } \tau_{min} = \tau_L \\ -\tau_{min}/2 + (\tau_L + \tau_F)/4, & \text{if } \tau_{min} = \tau_F \end{cases} \tag{11}$$

The error between a satellite's current orbital phase and its desired orbital phase is then:

$$e = \frac{2\pi(\tau - \tau_{desired})}{P} \tag{12}$$

To move to the desired phase, each satellite sets its drag using a PD control law:

$$u_{PD} = K_P e + K_D \dot{e}. \tag{13}$$

Additionally, a saturation operation is applied to u_{PD} to ensure than each satellite's value of J stays within ΔJ of J_0:

$$u = \begin{cases} u_0, & J > J_0 + \Delta J \\ u_0, & J < J_0 - \Delta J \\ u_{PD}, & \text{otherwise} \end{cases} \tag{14}$$

5 Numerical Simulations

To demonstrate the proposed control laws, numerical simulations of the full orbital dynamics of a satellite constellation in low-Earth orbit were performed using the dynamics presented in Section 2 and the control laws presented in Section 3. MATLAB's ODE45 solver was used to propagate the nonlinear equations of motion for each satellite. Physical parameters (mass, area, etc.) for the Sprite ChipSat were used in all simulations.

These simulations were used to test a variety of number of satellites for both the equal-spacing and clustering controllers. Fig. 2 show an example of how the simulated satellites behave with the equal-spacing controller. We tested both the equal-spacing and clustering controllers for a range of swarm sizes from 4 to 50 satellites. Figure 5 shows the end result of the equal-spacing controllers for a range of swarm sizes. Qualitatively the controllers generated the desired spacing and resulted in a stable final formation. To quantitatively measure performance for each experiment, we measured the distance between each satellite and its nearest ahead and behind neighbor. Figure 6 shows how these distances evolve over time for a equal-spacing and a clustering experiment. As expected, all satellites move to positions that have approximately the same distance between itself and its nearest neighbors for the equal-spacing controller, and all distances are close to zero for the clustering experiment.

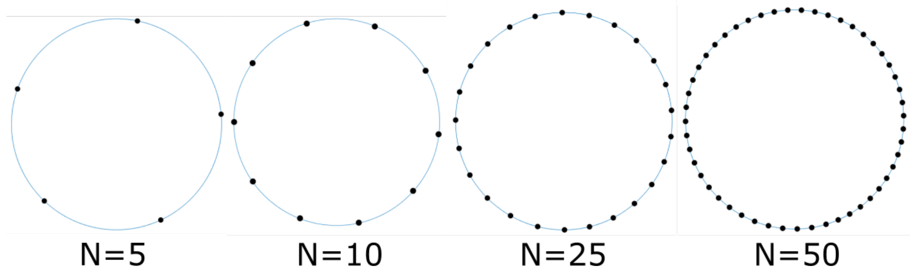

Fig. 5. Convergence of satellite position using an identical equal-spacing controller for 5, 10, 25, and 50 satellites.

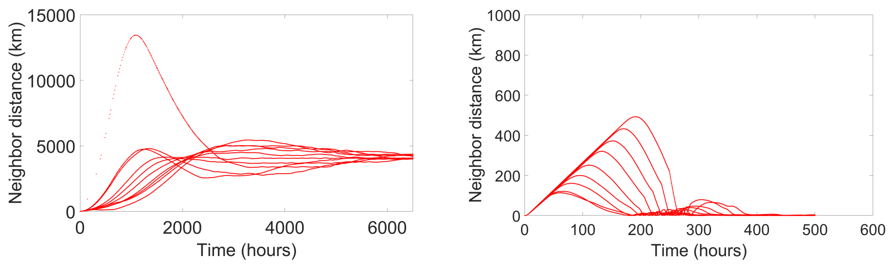

Fig. 6. Spacing between satellites and their nearest neighbors. (Left) Ten satellites use the equal-spacing controller to move apart from each other. (Right) Ten satellites first spread out in orbit and then use the clustering controller to move to the same point in space.

6 Conclusions

Here we have shown a fully distributed controller that enables swarms of satellites to either spread equally out in orbit, or cluster together in orbit. This controller could be used to reduce the human control needed to create and maintain current satellite constellations, or could enable much larger numbers in the future. The satellite requirements are simple enough to implement this on gram-scale ChipSats like the Sprite [7]. In the future we hope to be able to demonstrate this controller in a real system

References

1. Banazadeh, P.: Space Infrastructure 2.0! Meet Capella Space (May 2017)
2. Degesys, J., Rose, I., Patel, A., Nagpal, R.: Desync: self-organizing desynchronization and tdma on wireless sensor networks. In: Proceedings of the 6th international conference on Information processing in sensor networks. pp. 11–20. ACM (2007)
3. Foster, C., Hallam, H., Mason, J.: Orbit Determination and Differential-drag Control of Planet Labs Cubesat Constellations. arXiv:1509.03270 [astro-ph, physics:physics] (Sep 2015)

4. Foster, C., Mason, J., Vittaldev, V., Leung, L., Beukelaers, V., Stepan, L., Zimmerman, R.: Constellation Phasing with Differential Drag on Planet Labs Satellites. Journal of Spacecraft and Rockets 55(2), 473–483 (Mar 2018)
5. Henry, C.: Iridium teams up with LEO Internet of Things startup Magnitude Space. http://spacenews.com/iridium-teams-up-with-internet-of-things-startup-magnitude-space/ (Sep 2017)
6. Henry, C.: OneWeb asks FCC to authorize 1,200 more satellites. http://spacenews.com/oneweb-asks-fcc-to-authorize-1200-more-satellites/ (Mar 2018)
7. Manchester, Z., Peck, M., Filo, A.: Kicksat: A crowd-funded mission to demonstrate the world's smallest spacecraft. In: AIAA/USU Conference on Small Satellites. Logan, UT (2013), pubs-conference
8. Meyer, D.: Here's What You Need to Know About SpaceX's Satellite Broadband Plans. http://fortune.com/2018/02/22/spacex-starlink-satellite-broadband/ (Feb 2018)
9. Mirollo, R.E., Strogatz, S.H.: Synchronization of pulse-coupled biological oscillators. SIAM Journal on Applied Mathematics 50(6), 1645–1662 (1990)
10. Swartwout, M.: The First One Hundred CubeSats: A Statistical Look. Journal of Small Satellites 2(2), 213–233 (2013)
11. Vance, A.: The Tiny Satellites Ushering in the New Space Revolution. Bloomberg.com (Jun 2017)
12. W. H. Clohessy: Terminal Guidance System for Satellite Rendezvous. Journal of the Aerospace Sciences 27(9), 653–658 (Sep 1960)