2017-08-07

An Integrated Architecture for Distributed Estimation, Navigation and Control of Aerospace Systems

Thanh Vu
University of Miami, t.vu3@umiami.edu

UNIVERSITY OF MIAMI

AN INTEGRATED ARCHITECTURE FOR DISTRIBUTED ESTIMATION, NAVIGATION AND CONTROL OF AEROSPACE SYSTEMS

By

Thanh Vu

A DISSERTATION

Submitted to the Faculty
of the University of Miami
in partial fulfillment of the requirements for
the degree of Doctor of Philosophy

Coral Gables, Florida

August 2017

UNIVERSITY OF MIAMI

A dissertation submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

AN INTEGRATED ARCHITECTURE FOR DISTRIBUTED ESTIMATION, NAVIGATION AND CONTROL OF AEROSPACE SYSTEMS

Thanh Vu

Approved:

Ryan Karkkainen, Ph.D.
Professor of Mechanical and Aerospace Engineering

Amir Rahmani, Ph.D.
Robotics System Engineer
Jet Propulsion Laboratory
California Institute of Technology

Neil Johnson, Ph.D.
Professor of Physics

Jae Chung, Ph.D.
Professor of Mechanical Engineering
Stevens Institute of Technology

Guillermo J. Prado, Ph.D.
Dean of the Graduate School

VU, THANH (Ph.D., Mechanical and Aerospace Engineering)

An Integrated Architecture
for Distributed Estimation, Navigation and Control
of Aerospace Systems (August 2017)
Abstract of a dissertation at the University of Miami.

Dissertation supervised by Professor Ryan Karkkainen.
No. of pages in text. (71)

Distributed autonomous systems have demonstrated many advantageous features over their centralized counterparts. In exchange for their scalability and robustness, there are additional complexities in obtaining a global behavior from local interactions. These complexities stem from a coordination problem while being provided a limited information set. Despite this information constraint, it is still desirable for agents to construct an estimate of the entire system for coordination purposes.

This work proposes an architecture for the coordination and control of a multi-agent system while being limited in external communication. This architecture will examine how to internally estimate states, assign goals, and finally execute a suitable control. These techniques are first applied to a simple linear model and then extended to non-linear domains. Specific aerospace applications are explored where there exists a need for accuracy and precision.

to my friends and family

no matter where they are

Acknowledgements

I would like to thank my advisor Dr. Amir Rahmani and my co-advisor Dr. Jae Chung, who supported me in the past few years through the research and completion of my degree. I believe their personality and technical capability was an indispensable factor for me to finish this endeavor.

Thanh Vu

University of Miami

August 2017

Table of Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Distributed Systems

With ongoing advances in multi-agent robotics, there has been a growing need for decentralized algorithms to ensure robustness, scalability, and computational efficiency compared to their centralized counterparts. These distributed algorithms, despite their additional complexity, are seen as the primary of implementing networks of large sizes as seen by Hadaegh et al. [1]. Distributed algorithms cover many heavily-researched topics of multi-agent systems such as localization, sensing, formation control, and task assignment. Individually, the distributed algorithm in each of these areas seeks to replicate capabilities traditionally seen in monolithic architectures while maintaining the scalability seen in distributed networks. Formation flying spacecraft is a particular application where all of these algorithms become necessary for large scale implementation. For example, Roberts [2] shows that scientific inferometry missions rely more and more on keeping a tight formation while efficient formation assignment would be able to address fuel considerations.

1.1.1 Formation Flying Spacecraft

Formation flying satellites are becoming a major new development in space operations. The commercial, scientific, and military sectors all wish to expand the mission capabilities of their satellite fleets such as increased communication volume for information transfers, increased field of surveillance, and improved navigational accuracy for military and civilian aircraft [3]. While current distributed missions, such as GPS, are conducted in satellite constellations, these constellations do not have a coupled control law that takes into account the states of other satellites [4]. Therefore, formation flying would be able to perform missions with tighter formation requirements. With limited funds, implementing formation flying into existing satellite technology is a cost-effective way to extract more utility. Formation flying capabilities increases not only the scope of satellite missions but also the reliability. In addition to performing synchronous measurements, formations have redundancies in operation, which means failure of one spacecraft would not endanger the integrity of the mission [5]. Through the comprehensive survey on guidance and control techniques for formation flying spacecraft by Scharf et al., one can see that a central-control framework lacks robustness to changes within the formations [4, 6]. Distributed systems also allow for additional autonomy in conducting missions as they can reduce the reliance on receiving instructions from a ground station [5]. However, the fundamental challenge to the implementation of distributed systems is achieving a desired global outcome from isolated, local interactions.

A notable but novel use of formation flying can be seen in the distributed aperture telescope system [2]. In light of current restrictions on the cost and logistics of sending large telescopes for scientific observations, formation flying spacecraft could be used

to circumvent this problem. Each of the satellites within the formation would act as a section of a larger reflecting telescope, and the formation, as a whole, would become a "virtual telescope" with an aperture several times larger than their conventional reflecting counterparts [7]. This would give astronomers access to better clarity and resolution compared to individual telescopes. Lastly, the robustness of architecture would avert a incident similar to the Hubble telescope, which had a manufacturing defect in its lens and had to be repaired in space. Smaller mirrors would be more cost-efficient to manufacture, and satellites with defective instruments can be replaced and substituted fairly easily. This application is notable for its tight requirement on the shape of the formation in order to achieve satisfactory resolution. Thus, accurate estimation of the global formation structure is paramount to the success of such a mission. With a deliberate need for accuracy and precision, swarm based approaches, which emphasize low-complexity to reduce computational and power requirements, would not be entirely suitable for such an application [8].

1.1.2 Heterogeneous Systems

Most of the time, actuators and physical capabilities of agents differ dramatically. In the context of search and rescue vehicles, air-based vehicles and ground based vehicles each offer advantages and shortcomings that could be mitigated if their capabilities were combined. Aerial vehicles have extended range, increased field of vision, and are less susceptible to obstacles. Ground-based vehicles can maneuver into tight space, require minimal power consumption on standby, and are more durable to environmental hazards. Combining these capabilities effectively would greatly increase the efficiency of these autonomous units. In addition to differing capabilities, hetero-

geneous systems display more resiliency to failure. For example, a major storm or sudden gust could wipe a whole swarm of UAVs, however ground vehicles would be less susceptible to weather; whereas a earthquake could devastate ground units, but aerial ones would be left intact.

1.2 Areas of Interest for Multi-Agent Systems

1.2.1 Estimation

State estimation and localization has always been an integral part of a robot's perception component. A pertinent topic in localization is overdetermined navigation, which arises in situations where GPS is not readily available. Localization in a global frame of reference is often very difficult without such external communication. While a naive solution could simply be obtained by integration of on-board sensors, because of accumulated numerical errors and noise, this odometry method would start to drift away from the true value. Therefore, fusion of position data from odometry and externally measured correction is necessary to provide an accurate estimation of positioning [9, 10]. In the context of distributed systems, with each agent operates a local estimation that does not necessarily include all the states of the entire system; however, through communication, information can be passed on through the system so eventually unobservable states could be estimated. Rao et al. [11] first propose a fully decentralized Kalman filter which effectively partitions the centralized Kalman filter and then fuses the communication via a assimilation of the all information collected. Olfati-Saber [12, 13] proposed a distributed network of Kalman filters that share information about estimates and uncertainty and then combines them via a

consensus algorithm. Thus, a framework for fusing multiple estimation schemes is elegantly established. Martinez [14] extended the Kalman filter concept to mapping applications via a more general spatial-based framework when agents estimate fields quantities within their vicinity and perform consensus to construct the global map.

An alternate framework for combining consensus with Bayesian filter is proposed by [15]. Its major advantages over a standard Kalman Filter approach is its general applicability to nonlinear systems with the ability to fully propagate dynamics and uncertainties. However, the Bayesian filtering is more computational expensive and its combination with consensus requires more for bandwidth to transmit the necessary probability density functions.

1.2.2 Control

Control of multi-agent systems is a topic that is extremely rich in size and scope. There are many different differing scenarios that researchers have used to tackle this problem. The first approach has been the centralized approach where there is a single agent or observer that takes all the current information on the state of the system and computes every single trajectory and control that all the agents would follow. The primary advantage is that you can employ inexpensive, "dumb" agents because there is not a significant need for on-board processing. However, it is not a robust way of organizing a system. Since all effective computational power rests with the central unit, then it can not react as quickly to disturbances and failures. It also cannot accommodate changes in the network size as the central unit has to re-updated for the new changes. However, distributed architectures provides a solution to many of these issues, such as resiliency to single-point failures, adaptability to

system changes, and scalability to large number of agents. Since they rely heavily on communication, however, bandwidth and communication issues are large concerns compared to the centralized case. Many different methodology of distributed control have been considered. The first is a simple leader-follower system [16–18] which reduces the problem to orienting the swarm to a single agent, which eliminates the need for a global coordinate system. Gradient descent approaches have also been used because of ease of implementation [19, 20]. Decentralized controllers based on convex optimization [21], have generally been used to ensure constraints satisfaction and optimality of the solution relative to simple linear controllers. There have also been probabilistic techniques such as Markov chains [22], bio-inspired control [23], and evolutionary genetic algorithms [24]. These probabilistic approaches utilize a simpler framework at the agent level so it is very effective in organizing large-scale swarm and are extremely robust to single-point failures.

Concerns in estimation naturally leads into investigation into control in the presence of such uncertainty. The premier example for the single agent is the separation principle Tillerson et. al. [25] investigates the effectiveness of an LP controller with regards to different methods of localizing a formation of satellites. Rantzer [26] investigated the use of multiple controllers working as a team and was able to prove a separation principle for output feedback with bounded information propagation. Smith and Hadaegh sketched a formation controller that uses parallel estimators, however the full formation states have to be observable by every estimator [27]. Rahmani et al. [28] was able to extend this idea with distributed systems that estimates not only states but also control of the entire formation via communication across

a connected network. Vu and Rahmani [29] then showed that these results could successfully apply to spacecraft in a low-earth orbit.

1.2.3 Guidance and Assignment

Another major area of interest is task assignment, which is a combinatorial optimization problem that seeks to find a minimum (or maximum) weight matching of a bipartitite graph. With respect to formation control, this is usually seen as assignment of positions within the formation with the weights being the distance to each location. One of the first major algorithms is the Iterated Closest Point Algorithm (ICP), first used by Besl [30] and Chen [31] to match point clouds for 3D shapes. This algorithm seeks to minimize total distance error by selecting an optimum translation and rotation from one set to the other based on some initial data. While this algorithm will always converge, its direct application to the assignment problem leads to two unwanted phenomenons. The first problem is that there is no guarantee of a one to one mapping between the two sets because it relies on the nearest neighbor rule to match up points. McDonald [32] resolved this problem by incorporating the Hungarian algorithm to maintain the bijective nature of the matching. However, the modified algorithm will still converge to a local minimum based on the initial data. Therefore, it is still necessary to obtain a good estimate for the initialization of the rotation and translation vector. Alighanbari and How [33], proposes a robust task assignment algorithm based on the petal algorithm, which achieves the same performance as a centralized controller but requires a much higher bandwidth require and computational power so that the agents can achieve the same situational awareness.

An alternative solution to the assignment problem is an auction algorithm, in which tasks would be assigned to the higher bidder among the swarm. While initial auction algorithms, like [34], needed a centralized depository to keep track of all the bids, distributed-based auctions have since been implemented as seen by Choi et al. [35] who allow the new bids to propagate through the network via a concensus algorithm. Choi also considered the possiblility of of multi-task assignment by bundling these task and performing bundle construction and conflict resolution separatedly. Morgan et al. [36] proposes an architecture that deploys both swarm assignment and trajectory optimization (SATO). This architecture uses an auction-based approach to determine the assignment and sequential convex programming to generate trajectories before combining them into a model predictive control formulation. However, a key difference between the auctions-based approach and the architecture to be proposed is the degree of communication. The SATO algorithm requires on both perfect information states and transmission of nominal trajectories and bids to propagate through the network, whereas our architecture addresses the estimation problem while only relying on the communication of the current neighboring state estimates and covariances.

If one considers task assignment in terms of goal selection, then it can also be thought as a coordination problem in guidance. Cortes et al. [20] uses a descent based approach combined with a partition of the target space to move a formation to cover the space. Saptarshi et al. [37] proposes a method of combining guidance, estimation, and control by combining an inhomogeneous Markov chain with the Bayesian Consensus Filter in order to estimate the probability density functions for the swarm and thus reduce the number of transitions to convergence. This approach is extremely

effective in organizing large number of agents; however, because it is still probabilistic in nature then applications for precise, tight formation are still limited.

1.3 Summary

In most engineering applications, subsystems are designed by different teams and assembled into the finished product. From a systems engineering perspective, this approach is not guaranteed to work, and additional analysis is necessary to ensure adequate performance. By extending this concept to distributed systems, this work proposes to establish a framework for combining these topics of multi-agent systems in an unified architecture. This unified architecture would then be demonstrated through analysis of the subsystems and then verified through simulations.

CHAPTER 2

Review of Essential Concepts

As this work seeks to build framework around control of systems with sensor networks, then relevant topics from control and graph theory will be summarized. These topics will appear again in the context of distributed systems

2.1 Control Theory

Any dynamic system can modeled by a set of differential equations for continuous time or difference equations for discrete time in terms of state variables $x \in \mathbb{R}^p$ and control inputs $u \in \mathbb{R}^q$.

$$\dot{x} = f(x(t), u(t), t) \tag{2.1}$$

$$x[k+1] = g(x[k], u[k], k) \tag{2.2}$$

If the system is linear, the f and g are simply linear combinations of x and u, with $A, \Phi \in \mathbb{R}^{p \times p}$ are the state matrices and $B, \Gamma \in \mathbb{R}^{p \times q}$ are the control matrices:

$$\dot{x} = A(t)x(t) + B(t)u(t) \tag{2.3}$$

$$x[k+1] = \Phi[k]x[k] + \Gamma[k]u[k] \tag{2.4}$$

Typically in the absence of any control input, then a system is asymptotically stable if all the eigenvalues of A are on the open left half complex plane or all the eigenvalues of Φ are inside the unit circle.

2.1.1 Lyapanov Theory

Stability analysis for nonlinear and time-varying systems is relatively difficult to accomplish given the standard definitions of (asymptotic) stability because these systems are so rich in behavior. What Lyapanov noticed was that if one can show that there exists an energy-like function of the state of the system and which obeys certain dissipation requirements, then the system is guaranteed to be stable. Lyapanov's theorem for global asymptotic stability is as follow [38]

Proof: Suppose there exists a scalar function $\mathbf{V}(x(t))$ with the following properties

$$\mathbf{V} > 0 \, \forall x \neq 0$$

$$\mathbf{V} = 0 \, for \, x = 0$$

$$\dot{\mathbf{V}} < 0 \, \forall x \neq 0$$

Now assume that x does not go to 0.

Then there must be an $\epsilon > 0$ such that $\epsilon \leq V(x(t)) \leq V(x(0))$

This is a closed and bounded interval therefore $\dot{V}$, which is assumed continuous, attains its maximum, $-m < 0$, within this interval. Therefore

$$V(x(T)) = V(x(0)) + \int_0^T \dot{V}(x(t))dt$$
$$\leq V(x(0)) - mT$$

Therefore $T > V(x(0))/m$ implies $V(x(T)) < 0$ but this is a contradiction to the assumptions of V

Thus $x(t) \to 0$ ■

The scalar function V is called a Lyapanov function of the system. In cases of physical systems, the function corresponds to the energy of the system. Since this proof places no restrictions on the dynamical system considered, then it can be generally applied to time-varying and nonlinear systems.

2.1.2 Observer-controller Duality

Even if a system is naturally unstable, there are still conditions where it is possible to achieve stability by controlling the system. For a linear time invariant system, its controllability matrix $\mathcal{C} \in \mathbb{R}^{p \times pq}$ needs to be a rank p matrix.

$$\mathcal{C} \quad = \quad [B\, AB\, A^2B \ldots A^{p-1}B] \tag{2.5}$$

If this condition is met then the system is called controllable or (A, B) is called a controllable pair. It ensures the existence of a control sequence that will move the system from one state to any other state in finite time. A basic class of controllers is the state feedback controller where $u(t) = Kx(t)$ and there the complete system dynamics would be as follows:

$$\dot{x} = Ax(t) + Bu(t) = (A + BK)x(t) \tag{2.6}$$

$A + BK$ is the closed loop state matrix and if the system is (A, B) controllable, K could be constructed such that $A + BK$ could have arbitrary eigenvalues. Another closely related concept is observability of the system.

Many times some states are not directly measurable, but the output, z, some function of the the states and the input is measurable.

$$z = h(x(t), u(t)) \tag{2.7}$$

An important subclass of possible functions is the linear time invariant class, where the dynamics are linear and the output is is simply a linear combination of the states as encapsulated by the measurement matrix $H \in \mathbb{R}^{r \times p}$:

$$\dot{x} = Ax(t) + Bu(t) \tag{2.8}$$

$$z = Hx(t) \tag{2.9}$$

An observability matrix $\mathcal{O} \in \mathbb{R}^{rp \times p}$ can be constructed similarly to equation 2.5, and if it has rank p then there would exist a state observer that could construct the states from the outputs, with current estimate $\hat{x} \in \mathbb{R}^p$ and observer gain $L \in \mathbb{R}^{p \times r}$.

$$\mathcal{O} = \begin{bmatrix} H \\ HA \\ HA^2 \\ \vdots \\ HA^{p-1} \end{bmatrix} \tag{2.10}$$

$$\dot{\hat{x}} = A\hat{x} + Bu + L(z - H\hat{x}) \tag{2.11}$$

Let $\eta = x - \hat{x}$ then the dynamics of the error can be written as:

$$\dot{\eta} = A\eta - L(Hx - H\hat{x}) = (A - LH)\eta \tag{2.12}$$

This time if (A, H) is an observable pair then L can constructed such that the $A-LH$ have eigenvalues all on the left half complex plane or negative definite. One of the most powerful theorem in control theory is the separation principle which states the controller gain K and the observer gain L can be constructed separately without affecting the overall stability of the system. A generalization of this statement will be explored for distributed systems in chapter 3. If K was generated by least square controller, then it seeks to minimize the square of the control effort u then it is called a least quadratic regulator. Since observer and controller are mathematical duals of one another, then a least square estimator also exists and is called the Kalman filter, whose observer gain satisfies some form of Riccati equation, as seen in equation 2.15. In fact a system is (A, B) controlable is controllable if and only if (A^T, B^T) is observable.

$$\dot{\hat{x}} = A\hat{x} + Bu + L(z - H\hat{x}) \tag{2.13}$$

$$L(t) = P(t)H^T R^{-1} \tag{2.14}$$

$$\dot{P} = AP(t) + P(t)A^T + Q - L(t)RL^T(t) \tag{2.15}$$

The parameters $Q \in \mathbb{R}_+^{p\times p}$ and $R \in \mathbb{R}_+^{r\times r}$ are positive definite matrices that ensure P is also positive definite. In addition if the measurement and process disturbances are white Gaussian noise, and if Q and R correctly characterize the noise covariance statistics respectively, then the Kalman filter would be the optimal least squares estimator for x. While the Kalman filter propagates the covariance error online in real time, (A, H) observability guarantees that a steady state value exists, P_∞, given as the positive definite solution of the algebraic Riccati eqution. This in turn gives a steady value of of the observer gain, L_∞.

$$\mathbf{0} = AP_\infty + P_\infty A^T + Q - P_\infty H^T R^{-1} H P_\infty \tag{2.16}$$

$$L_\infty = P_\infty H^T R^{-1} \tag{2.17}$$

These steady state values are useful in offline analysis of the filter and gives a idea of how to initialize the Kalman filter. They also come in handy when onboard computational power is severely restricted.

2.2 Graph Theory

Graph theory started an area of mathematics that seeks that quantitatively describe the structure of a network. In general a graph $\mathcal{G}$ is a collection of two sets

$(\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of the nodes or vertices in the network given by some enumeration and $\mathcal{E} \subset \mathcal{N} \times \mathcal{N}$ is the set of edges where each element indicates a connection between its first and second arguments. Consequently, the size of the network is given by $|\mathcal{N}| = n$. A path is an alternating sequence of vertices and edges in which each vertex is unique (except possibly the first and last) and is preceded and followed by an edge to which it is a member of. A connected graph is one where there exists a path between any two vertices in $\mathcal{N}$. For any vertex i, its neighborhood $\mathcal{N}_i \subset \mathcal{N}$ is the set of vertices that shares an edge with i.

One of useful tools for the analysis of graph has been algebraic graph theory, which connects graph theory with linear algebra. For any graph, its matrix representation can be given by the adjacency matrix, A, which is a boolean matrix in which an element $A_{ij} = 1$ if $(i, j) \in \mathcal{E}$ and 0 otherwise. Another matrix of interest is the degree matrix D which is a diagonal matrix given by the size of the neighborhoods of each vertex i.e. $D = diag(|\mathcal{N}_1|, |\mathcal{N}_2|, \ldots, |\mathcal{N}_n|)$. Finally its Laplacian matrix is simply:

$$\mathcal{L} = A - D \tag{2.18}$$

One of the most important features of the Laplacian is its eigenvalues give a measure of the connectivity of the graph. It can be easily verified that a ones vector of correct size is always an element of the null space of the the Laplacian. Any additional zero eigenvalue would indicate the graph is disconnected. In fact, the number of zero eigenvalues gives the number of connected components of the graph. Another important property is that it is positive semidefinite. If the eigenvalues are listed in increasing order, then the second eigenvalue is called the algebraic connectivity.

2.2.1 Consensus

With many different nodes in a network, and if each node was itself an observer of the formation, then surely they would disagree on their states. This is where the consensus protocol, developed by Olfati-Saber and Murray [39], becomes necessary to allow the nodes to agree with each other, as seen in equation 2.19, where x_i be the state of agent i and $x_j \in \mathcal{N}_i$.

$$\dot{x}_i(t) = \sum_{j \in \mathcal{N}_i} (x_j(t) - x_i(t)) \tag{2.19}$$

It can be shown that the consensus dynamics is closely related to the Laplacian of the network. It will converge to an element in the null space of the Laplacian, therefore for a connected network it converge to a multiple of the ones vector $\mathbf{1}$. In its current form the final agreed value will simplify the geometric centroid of the initial states of the network. By adding weights to the protocol it is possible to change the final converged value. The rate of convergence is controlled by the algebraic connectivity of the Laplacian.

CHAPTER 3

Distributed Linear Estimation and Control

3.1 Background

The first problem considered was to determine how would the separation principle hold in the context of distributed systems. First, a framework is needed to consider the states of such a system. For this analysis, a spacecraft flying formation is considered, but the framework would be valid for any linear system. The dynamics of a distributed system can be formed by first considering the dynamics of a single spacecraft and aggregating their respective states to form the state of the entire formation. Again, we will only consider linear dynamics for each spacecraft. While at first glance this might look restrictive, in practice dynamics of most planned formation flying missions can be represented by linearization around an operation point of interest, like the Clohessy-Wiltshire-Hills equations for relative orbital dynamics. Since spacecraft formation operate in relatively close proximity, the use of linearized dynamics is justified.

A sketch of the control scheme, along with its inter-dependencies, is summarized in Figure 3.1. The major concept is that each spacecraft is not only estimating its own states, but also the states, controls, and formation assignment of the other spacecraft.

Using its estimate of the formation control and orientation, it would then implement its own control and select its own goal, respectively. Communication of information enables estimation of states even if no spacecraft can observe the entire formation. With this framework in mind, the following sections will implement the structure with respect to the linear dynamics on the spacecraft formation.

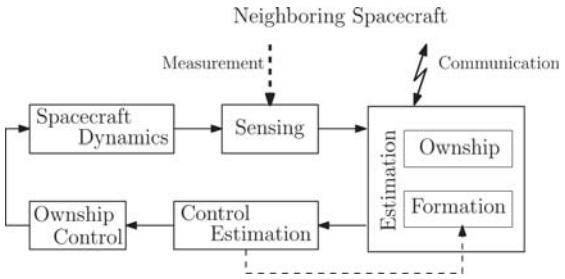


Figure 3.1: Sketch of distributed estimation and control

3.1.1 Dynamics

Consider first, n spacecraft, each under linear dynamics influenced by an exogenous, zero mean white Gaussian noise, v_i.

$$\dot{x}_i \quad = \quad A_i x_i + B_i u_i + v_i \tag{3.1}$$

The states of each spacecraft, x_i, can be concatenated to form an aggregated state vector, $X = [x_1^T \cdots x_n^T]^T$. Similarly, the control input and noise disturbances can also be aggregated as $U = [u_1^T \cdots u_n^T]^T$ and $V = [v_1^T \cdots v_n^T]^T$ respectively. Under this formulation, the state and and control matrix can be written in a block diagonal form: $A = diag(A_1, \cdots, A_n)$ and $B = diag(B_1, \cdots, B_n)$ respectively. These aggregated dynamics can now be written in a form analogous to equation 3.1. Note that while the use of different individual state matrices A_i keeps this analysis general for diverse,

heterogeneous systems, for homogeneous systems within the same environment, the state matrix would be identical.

$$\dot{X} = AX + BU + V \tag{3.2}$$

We assume, each spacecraft i can measure a subset of these aggregated states z_i, usually their own and a few select states of their neighbors, or a linear combination of them via a measurement matrix H_i, with an additive W_i, a zero mean Gaussian measurement noise. Thus, an aggregated system of measurements can be found.

$$z_i = H_i X + W_i \tag{3.3}$$

It should be noted that it has not been required that (A, H_i) be observable, which allows the possibility of the whole system not being observable by one spacecraft. Instead, a weaker condition is assumed where each state could be constructed by at least one spacecraft. In technical terms, this means that (A, H) is observable where $H = [H_1^T \cdots H_n^T]^T$. The linear combination of states for observation is especially applicable to distributed systems since inter-agent distance can simply be represented by 1 and -1 in the corresponding positions in the H_i matrices.

3.1.2 Distributed Estimation

We would like to employ state feedback to ensure global stability of the network, either by LQR or pole placement. However, both methodologies require that K, the state feedback gain, is calculated by assuming full knowledge of the entire formation states.

$$U_{ideal} = KX \tag{3.4}$$

Since complete knowledge of entire system states in a distributed setting is not possible, we assume each spacecraft can measure some of its own and neighboring spacecraft's states represented by $z_i = H_i X$. Provided the spacecraft can locally communicate and share their estimate of the entire formation state $\hat{X}_i$ with their immediate neighbors (sensing and communication neighborhoods do not necessarily need to be the same), each spacecraft can run its local version of a distributed consensus-based Kalman filter to estimate the state of the entire network. Olfati-Saber show that such a filter will converge to the same estimate in the absence of any external control.

However, the problem is that the dynamics of the entire formation depends on knowledge of actual control U implemented by all spacecraft. Since each spacecraft implements its own control locally and does not communicate this value with the entire formation, we use the following distributed estimation and control proposed by [28]. The distributed Kalman filter in presence of locally constructed controls is as follows:

$$\begin{aligned} \dot{\hat{X}}_i &= (A + BK)\hat{X}_i + K_i(z_i - H_i\hat{X}_i) \\ &+ \gamma P_i \sum_{j \in \mathcal{N}_i} (\hat{X}_j - \hat{X}_i), \qquad (3.5) \\ K_i &= P_i H_i^T R_i^{-1}, \qquad (3.6) \\ \dot{P}_i &= AP_i + P_i A^T + Q - K_i R_i K_i^T \qquad (3.7) \end{aligned}$$

where P_i, Q, and R_i are the symmetric covariance matrices of the error, process and measurement noise, respectively. Every term except for the third term in equation

3.5 is the standard local Kalman filter, with the addition of the extra term, which is the consensus algorithm. This term is weighted by both a scalar constant, γ, and the error covariance, P_i. The consensus, γ, is the relative weighting factor for the entire consensus contribution to the estimate, while P_i is weighing each state by its uncertainty. This means each spacecraft would put more "trust" in incoming data if its own estimate is more uncertain. Naturally, for states that it is not observing, then it would need to rely on incoming data to achieve an accurate estimate. Therefore, it is not necessary for any single spacecraft to observe the entire formation, and it is only required that (A, H) be a fully observable pair, i.e. any state is observable by at least one spacecraft.

3.1.3 Distributed Control

Since each spacecraft would have to obtain its own controls from its respective estimate, we can define Π_i as the selection function of its estimate inputs. This selection matrix is a block diagonal matrix whose blocks consist of zero matrices and one identity matrix. These block matrices are square matrices whose length is the size of the control input to each spacecraft, and the identity matrix is situated on the position corresponding to spacecraft i. Now it is possible to write out actual control input to the entire formation.

$$\Pi_i = diag\,(\mathbf{0}, \mathbf{0}, \ldots I \ldots \mathbf{0}) \tag{3.8}$$

$$U(t) = \sum_i \Pi_i \hat{U}_i = \sum_i \Pi_i K \hat{X}_i \tag{3.9}$$

Essentially, each spacecraft i holds a local estimate of what every spacecraft control action should be, i.e. $\hat{U}_i = K\hat{X}_i$. The selection function Π_i only selects the control

pertaining to spacecraft i and sets everything else to zero. This is the actual control implemented by spacecraft i. Hence, the actual control implemented by the entire formation is the concatenation of these locally constructed controls. This is achieved via the summation sign since the selection matrix, Π_i, returns values of spacecraft i's control at its corresponding locations and zero at every other location.

To show the convergence of each spacecraft's estimation to the global state, one can define a spacecraft's error vector. Subsequently the system dynamics can be derived in terms of state variables and the state error by exploiting the fact that $\sum_i \Pi_i = I$.

$$\begin{aligned} \eta_i &= X - \hat{X}_i \qquad (3.10) \\ \dot{X} &= AX + BU \\ \dot{X} &= AX + B\sum_i \Pi_i K \hat{X}_i \\ \dot{X} &= AX + B\sum_i \Pi_i K \left(X - \eta_i\right) \\ \dot{X} &= \left(A + BK\right) X - \sum_i B\Pi_i K \eta_i \qquad (3.11) \end{aligned}$$

Now it is possible to write the error dynamics in terms of the state error and simplify the consensus term since $X_j - X_i = \eta_i - \eta_j$.

$$\dot{\eta}_i = (A + BK - K_i H_i)\eta_i + \gamma P_i \sum_{j \in \mathcal{N}_i} (\eta_j - \eta_i) - \sum_j B\Pi_j K \eta_j \qquad (3.12)$$

where $\mathcal{N}_i$ is the communication neighborhood set of spacecraft i.

3.2 Convergence Conditions

Since we wish to consider the validity of the separation principle to distributed systems, then it is necessary both for the estimates to converge onto the global formation and for the formation itself to converge onto the desired formation simultaneously. With this in mind, we consider a Lyapunov function of the form:

$$V(t) = \sum_i \eta_i^T P_i^{-1} \eta_i + X^T X \tag{3.13}$$

By construction, this Lyapunov function is positive definite and can be zero if and only if the error vector of every spacecraft and the state vector are identically zero.

$$\dot{V} = \sum_i (\dot{\eta}_i^T P_i^{-1} \eta_i - \eta_i P_i^{-1} \dot{P} P_i^{-1} \eta_i + \eta_i^T P_i^{-1} \dot{\eta}_i) + 2X^T \dot{X} \tag{3.14}$$

$$\begin{aligned} \dot{V} &= -\sum_i \eta_i^T (H_i^T R_i^{-1} H_i + P_i^{-1} Q P_i^{-1}) \eta_i \\ &+ 2 \sum_i \left[\eta_i^T P_i^{-1} \left(BK\eta_i - \sum_j B\Pi_j K \eta_j \right) \right] \\ &+ 2\gamma \sum_i \sum_{j \in \mathcal{N}_i} \left[\eta_i^T (\eta_j - \eta_i) \right] \\ &+ 2X^T (A + BK) X - 2X^T \sum_i B\Pi_i K \eta_i \end{aligned} \tag{3.15}$$

Next we define the following relations for concise notation:

$$\begin{aligned} \eta &= \left[\eta_1^T \, \eta_2^T \, \ldots \, \eta_n^T \right]^T \\ \bar{\eta} &= \left[X^T \, \eta^T \right]^T \\ \Lambda_i &= H_i^T R_i^{-1} H_i + P_i^{-1} Q P_i^{-1} \\ \mathcal{M}_1 &= diag(\Lambda_1, \Lambda_2, \ldots, \Lambda_i) \end{aligned}$$

Using this notation, then it is possible to assemble block matrices for an even simpler representation:

$$
\begin{aligned}
& -\sum_i \eta_i^T \left(H_i^T R_i^{-1} H_i + P_i^{-1} Q P_i^{-1} \right) \eta_i + 2X^T (A + BK) X \\
= \; & -\bar{\eta}^T \begin{bmatrix} -2(A+BK) & & & & \mathbf{0} \\ & \Lambda_1 & & & \\ & & \Lambda_2 & & \\ & & & \ddots & \\ \mathbf{0} & & & & \Lambda_n \end{bmatrix} \bar{\eta} \\
= \; & -\bar{\eta}^T \begin{bmatrix} -2(A+BK) & \mathbf{0} \\ \mathbf{0} & \mathcal{M}_1 \end{bmatrix} \bar{\eta}
\end{aligned} \tag{3.16}
$$

By definition, $2\gamma \sum_i \sum_{j \in \mathcal{N}_i} \left[\eta_i^T (\eta_j - \eta_i) \right] = -2\gamma \eta^T \mathcal{L} \eta$ where $\mathcal{L}$ is the Laplacian of the graph. Now let us consider the following terms:

$$
\begin{aligned}
-2\sum_i \eta_i^T P_i^{-1} \sum_j BK\eta_j &= -2\eta^T \begin{bmatrix} P_1^{-1}B\Pi_1 K & P_1^{-1}B\Pi_2 K & \dots & P_1^{-1}B\Pi_n K \\ P_2^{-1}B\Pi_1 K & P_2^{-1}B\Pi_2 K & \dots & P_2^{-1}B\Pi_n K \\ \vdots & & & \vdots \\ P_n^{-1}B\Pi_1 K & P_n^{-1}B\Pi_2 K & \dots & P_n^{-1}B\Pi_n K \end{bmatrix} \eta \\
&= -2\eta^T \mathcal{M}_2 \eta \qquad (3.17) \\
-2X^T B \sum_i \Pi_i K \eta_i &= -2X^T \begin{bmatrix} B\Pi_1 K & B\Pi_2 & \dots & B\Pi_n K \end{bmatrix} \eta \\
&= -2X^T \bar{\mathcal{B}} \eta \qquad (3.18) \\
2\sum_i \eta_i^T P_i^{-1} BK \eta_i &= 2\eta^T \begin{bmatrix} P_1^{-1}BK & & & \\ & P_2^{-1}BK & & \\ & & \ddots & \\ & & & P_n^{-1}BK \end{bmatrix} \eta \\
&= 2\eta^T \mathcal{M}_3 \eta \qquad (3.19)
\end{aligned}
$$

We can combine the previous equations to show that the time derivative of the candidate Lyapunov function be characterize by a single block matrix:

$$
\begin{aligned}
\dot{V} &= -\bar{\eta}^T \begin{bmatrix} -2(A+BK) & 2\bar{\mathcal{B}} \\ \mathbf{0} & \mathcal{M}_1 + 2\mathcal{M}_2 - 2\mathcal{M}_3 + 2\gamma\mathcal{L} \end{bmatrix} \bar{\eta} \qquad (3.20) \\
&= -\bar{\eta^T} \mathcal{H} \bar{\eta} \qquad (3.21)
\end{aligned}
$$

For this system to be stable, then $\mathcal{H} \succ 0$. Since $\mathcal{H}$ is block triagonal, this must imply that blocks on the diagonal are positive definite:

$$-2(A+BK) \succ 0 \qquad (3.22)$$

$$\mathcal{S} = \mathcal{M}_1 + 2\mathcal{M}_2 - 2\mathcal{M}_3 + 2\gamma\mathcal{L} \succ 0 \qquad (3.23)$$

By assuming that (A, B) is controllable, K was constructed to ensure that the closed loop system is stable, then $A + BK \prec 0$, and since every $\Lambda_i \prec 0$, then $\mathcal{M}_1 \succ 0$. Since the eigenvalues of each P_i are bounded from below by the Cramer-Rao bounds [40], which are nonzero because of the process noise Q, then P_i^{-1} is guaranteed to have bounded eigenvalues. Since $\mathcal{L}$ is assumed to represent a connected graph, it is positive semi-definite, and its nullspace is within $span\{\mathbf{1}\}$. Therefore, one can select a sufficiently large γ such that $\mathcal{S} \succ 0$. Intuitively this means that consensus must be reached first so that each agent will have a reliable estimate of total control input. Once this occurs, then each agent would be running an estimation scheme as if there is a singular global Kalman filter.

Thus, the system is stable if the following assumptions are satisfied:

$$(A, B)\, is\, controllable \tag{3.24}$$

$$(A, H)\, is\, observable \tag{3.25}$$

$$nul\mathcal{L} = span\{\mathbf{1}\} \tag{3.26}$$

$$\exists\gamma : \mathcal{M}_1 + 2\mathcal{M}_2 - 2\mathcal{M}_3 + 2\gamma\mathcal{L} \succ 0 \tag{3.27}$$

3.3 Design Considerations

Since the design of γ is influenced by the values of K, strictly speaking the separation principle cannot be applied. Fortunately, there is only a unidirectional dependence, as the controller can to be designed first without considering the estimator. Only the estimator depends on the controller gains. Mathematically, this can be seen in the derivation of the differential Riccati equation by Kalman and Bucy, [41] which

assumes that the control input is explicitly known at all times, and has no effect in the evolution of the error. Therefore, once there is a consensus in estimation, each agent knows exactly what the global input is to the system. However, practically speaking, in most applications, there are physical or design constraints on the gain values, therefore initial design consideration could assume the worst case bound on the gain values and then optimize as the design progresses.

In terms of selecting an appropriate γ, the last stability condition is rather cumbersome to use in designing a system because $\mathcal{S}$ has an implicit time dependence due to P_i^{-1}. Nevertheless, one can construct matrices $\bar{\mathcal{M}}_2$ and $\bar{\mathcal{M}}_3$ whose eigenvalues would be a worst case upper bound on the their respective counterparts. Using these matrices, γ could be designed offline and also ensure stability for all time. As mentioned earlier, the eigenvalues of P_i are bounded from below by the Cramer-Rao bounds [40], therefore the eigenvalues from the Fisher information matrix, J, are the upper bound for the eigenvalues of P_i^{-1}. Therefore, $\bar{\mathcal{M}}_2$ and $\bar{\mathcal{M}}_3$ can be constructed by replacing each P_i^{-1} by J:

$$\bar{\mathcal{M}}_2 = \begin{bmatrix} JB\Pi_1K & JB\Pi_2K & \dots & JB\Pi_nK \\ JB\Pi_1K & JB\Pi_2K & \dots & JB\Pi_nK \\ \vdots & & & \vdots \\ JB\Pi_1K & JB\Pi_2K & \dots & JB\Pi_nK \end{bmatrix} \tag{3.28}$$

$$\bar{\mathcal{M}}_3 = \begin{bmatrix} JBK & & & \\ & JBK & & \\ & & \ddots & \\ & & & JBK \end{bmatrix} \tag{3.29}$$

Similarly, a time-invariant version of the stability criterion can be evaluated using $\bar{\mathcal{M}}_2$ and $\bar{\mathcal{M}}_3$ and neglecting $\mathcal{M}_1$:

$$\bar{\mathcal{S}} = \bar{\mathcal{M}}_2 - \bar{\mathcal{M}}_3 + \gamma\mathcal{L} \succeq 0 \tag{3.30}$$

Since $\mathcal{M}_1$ is always positive definite, it was neglected in order to add a factor of safety to the stability of the system. One might also note the implicit assumption in stating that the eigenvalues of J are bounded. For this assumption to be valid, the designer will need to make sure that the covariance of the process noise, Q, is positive definite, which is standard practice. This prevents the eigenvalues of P_i from going to zero which allows P_i^{-1} to exist and J is bounded.

3.4 Simulation

To illustrate the implementation of this distributed control scheme on a system of satellite, the Hill-Clohessy-Wiltshire equations are considered as the natural dynamics of the environment, which are inherently unstable under most initial conditions [42]. Therefore maintaining positions in the radial, along track and cross track direction would indicate a validation of the control and estimation algorithm. The spacecraft would be orbiting at an attitude of 415 km above the surface of the Earth, which would corresponds to the orbit of the International Space Station. Each spacecraft will be able to measure its own states and the states of its neighbors. Neighbors are defined by the graph representation, given in Figure 3.2. Since the graph is connected, it ensures that (A, H) is fully observable. As for controllability, it was assumed that these spacecraft have control over all three acceleration components. The control

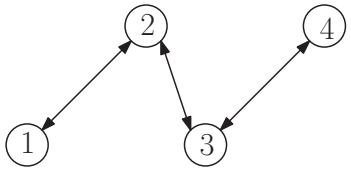


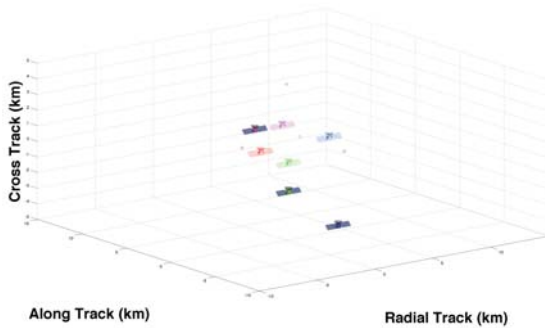
Figure 3.2: Graph representation of communication topology among spacecraft

gain was chosen by an LQR regulator which gives more weight to the states in order to show convergence of the system to the desired configuration. Though the initial states and estimated states were both randomly drawn from uniform distributions, the estimated states were given a wider range so that they would be more dispersed from the actual states.

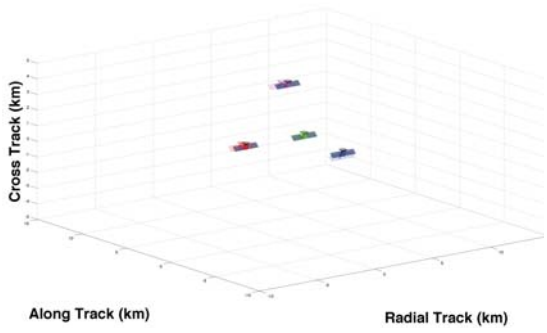
To simply assure positive-definite covariance matrices, positive scalar multiples of the identity matrix were chosen for the P, Q, and R matrices. Since the control scheme was given for the continuous Kalman-Filter, a Runge-Kutta scheme, with a time step of 0.1 seconds was utilized for a high-fidelity simulation. It was noted that changing the time step would surprisingly yet noticeably affect the trajectories because the time step would have a physical significance. During every iteration of the Runge-Kutta scheme, each spacecraft would place a zero order hold on the states communicated from its neighbors. Therefore, the time-step would represent the time lag in communication signals between the spacecraft.

The various graphs in Figure 3.3 show not only that the spacecraft can get into formation but also, through communication, each spacecraft would know where ev-

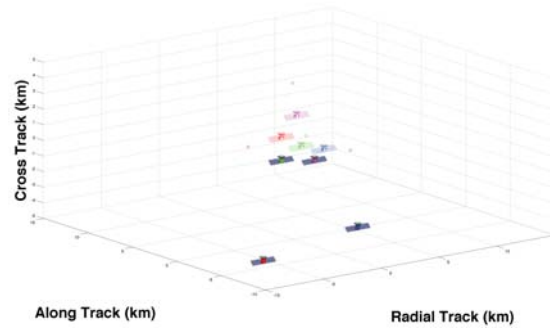
ery other one is located. Figure 3.3(a) shows a random initial configuration of the four spacecraft, transparent satellites represents the the estimated position that each satellite has for itself. Figure 3.3(b) shows the convergence of this configuration within 0.04 percent of its 92.6 minutes orbital period. Figure 3.3(c) similarly shows another random initial configuration, but now the estimated positions shown are the perception of the blue satellite over the entire formation. Finally Figure 3.3(d) again demonstrates convergence of the overall system along with the convergence of blue satellite's estimation. Randomizing the initial condition would also simulate a temporary loss of communication or memory as each spacecraft would try reestablish communication in order to find out the configuration of the formation. This also shows the resilience of the control scheme to changes in initial conditions.



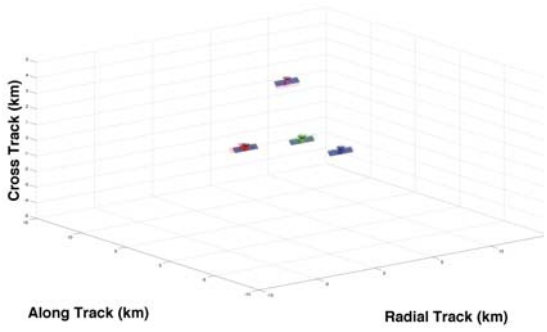
(a) Random initial configuration showing each spacecraft's own estimation of themselves



(b) Convergence to formation showing each spacecraft's own estimation of themselves



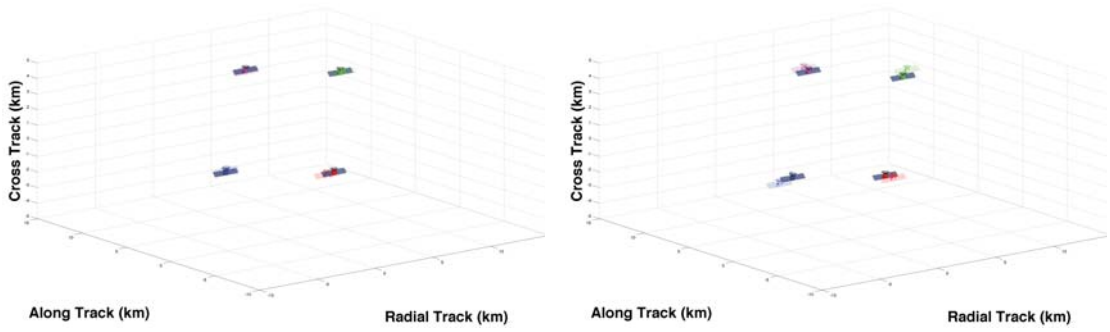
(c) Random initial configuration showing the blue spacecraft's estimation of the entire formation



(d) Convergence to formation emphasizing the blue spacecraft's estimation of the entire formation

Figure 3.3: Simulation of formation estimation and control in a 415 km circular orbit under CWH equations of motion.

This architecture is also not limited in possible ending configurations, and the formation can be made to converge to other locations in all three directions as seen in Figure 3.4(a). Whereas Figure 3.4(b) shows convergence to the same formation but with a five fold increase to each of the measurement covariance of each spacecraft. By amplifying measurement uncertainty, it leads to a higher dispersion around the equilibrium point in the estimation.



(a) Convergence to an alternate formation

(b) Convergence with five times increase to the measurement covariance

Figure 3.4: Illustration of additional convergence properties.

3.5 Summary

The stability criterion for distributed estimation and control of a collection of agents is derived in a linear continuous system. It is also shown that a fully generalized version of the separation principle is not achievable with the estimation scheme, and a weak version with one way dependence could be used for combined estima-

tion and control. Simulations were performed for satellites operating under linear orbital dynamics which validated results expected from the model. Since estimates are communicated in discrete intervals, these results would need to be extended into the discrete domain. A static graph is a major assumption taken within this work, and a generalization to state dependent graphs would be necessary as many real world applications have range limited communication and sensing.

CHAPTER 4

Distributed Assignment

In the previous chapter, it was shown that distributed estimation and control of satellite formation could result in a stable system. However, formation location and assignment were decided somewhat arbitrary, therefore a followup issue would be: *Is there an optimal way to assign a group of satellites into a formation*? By utilizing the Hungarian algorithm with a version of the ICP algorithm, McDonald developed an efficient algorithm for optimally assigning agents with a distance based cost function.

When the desired convergence states are also a function of time, then convergence to final equilibrium can be considered a tracking problem. Convergence analysis of a tracking problem would then be addressed using sliding mode control, where the switching mode are built into a Lyapanov function to derived a control law. [43, 44]. This approach has already applied to swarms, as shown by Yao et al. using artificial potentials. [45] However, the tracking trajectories are usually exogenous to the dynamical model. However, this paper will illustrate a distributed, fully nested architecture of estimation, guidance, and control that is shown to be stable even with local implementation of all three sub-processes. The overall control scheme would be based a simple state-feedback loop, thereby reducing the need to derive a complex control law.

4.1 Distributed Assignment Algorithm

Using the notation from McDonald, if P is the desired formation, given by a collection of points in $\mathbb{R}^2$, then we can define the assignment $Y : \mathbb{R}^2 \to \mathbb{R}^2$ to be a bijective relation between the a set of subvectors of x_i with P. With a cost function that would be geometric in nature, then these subvectors, $\tilde{x}_i$, would be the position components of each x_i. Succinctly, Y would map each current location of each spacecraft to a desired location for the formation. Then for $\theta \in \mathbb{R}$ and $\tau \in \mathbb{R}^2$ representing a rotation and translation respectively, the following cost function is defined:

$$\begin{aligned} L(Y, \tau, \theta) &= \sum_{i=1}^{N} \|R(\theta)Y(\tilde{x}_i) + \tau - \tilde{x}_i\|^2 \qquad (4.1) \\ R(\theta) &= \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \end{aligned}$$

McDonald's assignment algorithm then minimizes this cost function with the following steps:

1: **procedure** Assignment Algorithm
2: $k \leftarrow 0$
3: Guess $\tau \leftarrow \tau_k$
4: Guess $\theta \leftarrow \theta_k$
5: $Cost_0 \leftarrow \infty$
6: $\epsilon \leftarrow \infty$
7: **while** $\epsilon > 0$ **do**
8: $k \leftarrow k + 1$
9: Compute $Y_k \leftarrow \min_Y L(Y, \tau_{k-1}, \theta_{k-1})$

via Hungarian Algorithm

$Cost_k \leftarrow L(Y_k, \tau_{k-1}, \theta_{k-1})$

$\epsilon \leftarrow Cost_{k-1} - Cost_k$

Compute $(\tau_k, \theta_k) \leftarrow \min_{(\tau,\theta)} L(Y_k, \tau, \theta)$

$\theta \leftarrow tan^{-1}(\frac{W_2}{W_1})$

$$W_1 \leftarrow \sum_{i=1}^{N} (\tilde{x}_i - \mu_x)^T (Y(\tilde{x}_i) - \mu_y)$$

$$W_2 \leftarrow \sum_{i=1}^{N} (\tilde{x}_i - \mu_x)^T \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} (Y(\tilde{x}_i) - \mu_y)$$

$$\mu_x \leftarrow \frac{1}{N} \sum_{i=1}^{N} \tilde{x}_i$$

$$\mu_y \leftarrow \frac{1}{N} \sum_{i=1}^{N} Y(\tilde{x}_i)$$

$\tau \leftarrow \mu_x - \mu_y$

return $(Y, \tau, \theta) \leftarrow (Y_k, \tau_k, \theta_k)$

McDonald [32] proved that this algorithm would converge in finite time, therefore there is no issue of not finding an assignment. While this assignment was shown to work in a distributed setting, the complete graph requirement is a strong constraint placed on the algorithm. By incorporating the distributed Kalman filter, this assumption can be relaxed. In a distributed environment with noisy measurements, then each spacecraft would use its generated estimate of the global formation $\hat{X}_i$ and independently compute a formation pose, $(Y, \tau, \theta)_i$, for every spacecraft. It would then select its own desired goal from this estimated formation pose. Therefore, within distributed assignment, there is an additional layer of estimation, where in addition to estimating the states of every spacecraft in the formation, the global destination of all spacecraft is also simultaneously computed using the estimated state. Since the distributed assignment algorithm only uses the position components of X, then there

is additional freedom for the goals of the excluded components $V_{desired}$. The choice of notation reflects that these excluded components are usually the velocity components of the state vector X.

4.2 Complete Algorithm

All of the results described in the previous sections can be combined and summarized below in both graphical and algorithmic form. In order to save power and reduce computational cost after the convergence, the assignment algorithm would not update if the $\hat{X} - X_{goal}$ is within a ball with pre-determined radius ϵ.

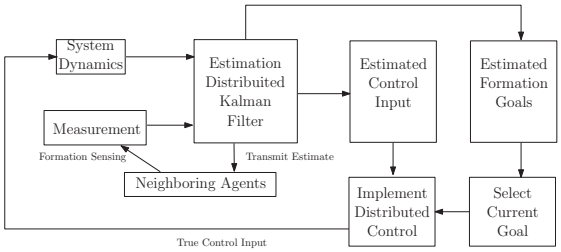


Figure 4.1: Sketch of proposed architecture

1: **procedure** Complete Distributed Algorithm

2: $\hat{X} \leftarrow X_0$

3: $P \leftarrow P_0$

4: $Y \leftarrow Y_0$

5: $\tau \leftarrow \mathbf{0}$

6: $\theta \leftarrow 0$

7: $V_{desired} \leftarrow \mathbf{0}$

8: **while** formation desired is available **do**

9: Extract positions $\{\tilde{x}_i\}$ from $\hat{X}$

Compute desired positions $\bar{x}_i \leftarrow R(\theta)Y(\tilde{x}_i) + \tau$

Assemble X_{goal} from $\{\bar{x}_i\}$ and $V_{desired}$

if New neighbor data is available **then**

    Update current neighboring estimate$\{\hat{X}_j\}$;

Update current estimate and covariance $(\hat{X}, P)$

    Using eq. 3.5 through 3.7

Implement control: $U \leftarrow \Pi K(\hat{X} - X_{goal})$

if $\|\hat{X} - X_{goal}\| > \epsilon$ **then**

    Compute (Y, τ, θ)

        Using Algorithm 0

4.3 Proof of Stability

Consider first the case where states are solely 2-D positions, with n number of agents. Therefore $X, \hat{X}_i \in \mathbb{R}^{2n}$ and $\{\tilde{x}_j\}_i$ could be could be concatenated to form $\hat{X}_i$. Within the assignment algorithm, each spacecraft forms its own estimate of the global desired position, which is a function of assignment, translation and rotation: $\hat{Z}_i = f(Y_i, \tau_i, \theta_i)$. Once again it is position to express the actual desired formation using the selection matrices and summing the result.

$$Z = \sum_{i=1}^{n} \Pi_i \hat{Z}_i$$

Based on the previous results from chapter 3, then it is known that for any given desired formation Z, along with other convergence conditions, then X will approach Z asymptotically. This proof was based on a Lyapanov function, given below where

$\mathcal{H}_3$ is a positive definite block trigonal matrix:

$$\begin{aligned} V(t) &= \sum_i \eta_i^T P_i^{-1} \eta_i + (X-Z)^T (X-Z) \\ \dot{V}(t) &= -\bar{\eta}^T \mathcal{H}_3 \bar{\eta} \end{aligned}$$

In other words, as long as Z is fixed, the Lyapanov function would be decreasing with time. The convergence rate could be estimated from below by the minimum eigenvalue of the $\mathcal{H}_3$: $V(t) \sim e^{-\lambda_1(\mathcal{H}_3)t}$. Therefore, every time Z is updated, then it would cause a step function jump in the Lyapanov function, which can be similarly represented in its approximation. Therefore as long the update time for Z is sufficiently long and the step function jump is bounded for all time, then Lyapanov function decreasing with time. To show that the jump is bounded, τ_i will first be shown to be bounded. If τ_i is bounded then we can at least ensure that the formation does not diverge to infinity.

$$\begin{aligned} \tau_i &= \mu_x - \mu_y = \frac{1}{N}\sum_{j=1}^{N} \tilde{x}_{i,j} - \frac{1}{N}\sum_{j=1}^{N} Y_i(\tilde{x}_{i,j}) \\ &= \frac{1}{N}\sum_{j=1}^{N} \tilde{x}_{i,j} - Y_i(\tilde{x}_{i,j}) \\ &\leq \frac{1}{N}\|\hat{X}_i - \hat{Z}_i\|_1 \\ &\leq \frac{1}{N}(\|\hat{X}_i - X\|_1 + \|X - \hat{Z}_i\|_1) \end{aligned}$$

The first term is going to zero because of the Lyapanov function, while the second term is at least bounded because at least $\tilde{x}_{i,i} - Y_i(\tilde{x}_{i,i})$ goes to zero and the terms in the norm sum is bounded by the geometric diameter of the desired formation. Now let's consider the convergence of the angles, θ_i. Since τ_i is bounded and the formation stays relatively localized, then one just has to wait for the estimates to converge. Feeding the same estimates values into the assignment algorithm would

mean that $\hat{Z}_i$ goes to Z. In addition convergence of the estimates, along with the Hungarian algorithm, ensures no overlap in matching. In other words, each assigned goal is unique to each agent. The final case is whether θ_i would keep change such that the agents are always chasing their goals and never truly converge. Based on the construction of the cost function, being distance-based, then when the assignment algorithm returns a new θ value, then it cannot return a value that would increase the total distance to goal convergence. Therefore, the assignment algorithm would never return a value that would rotate the formation goals away from equilibrium.

Finally for the general case, where you have additional degrees of freedoms due to the velocity components of the aggregate state vector, the controlability of (A, B) ensures that these components would also converge to zero.

4.4 Simulations

Again, to illustrate the implementation of this distributed architecture for spacecraft coordination, the natural dynamics of the environment is chosen to be the local vertical and local horizon components (LVLH) of the Hills-Clohessy-Wiltshire Equations. The cross track component is not considered since the system is stable in this direction.

$$A_i = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 2n \\ 0 & 0 & -2n & 0 \end{bmatrix} \tag{4.2}$$

$$n = \sqrt{\frac{\mu}{a^3}} \tag{4.3}$$

Following conventional notation, μ is the gravitational parameter of the Earth and a is the semi-major axis of the orbit of the origin of the LVLH coordinate system. Similarly, the International Space Station's altitude of 615 km is chosen. All other parameters, such as covariances and graph topology, remain the same as the previous chapter's.

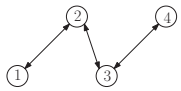
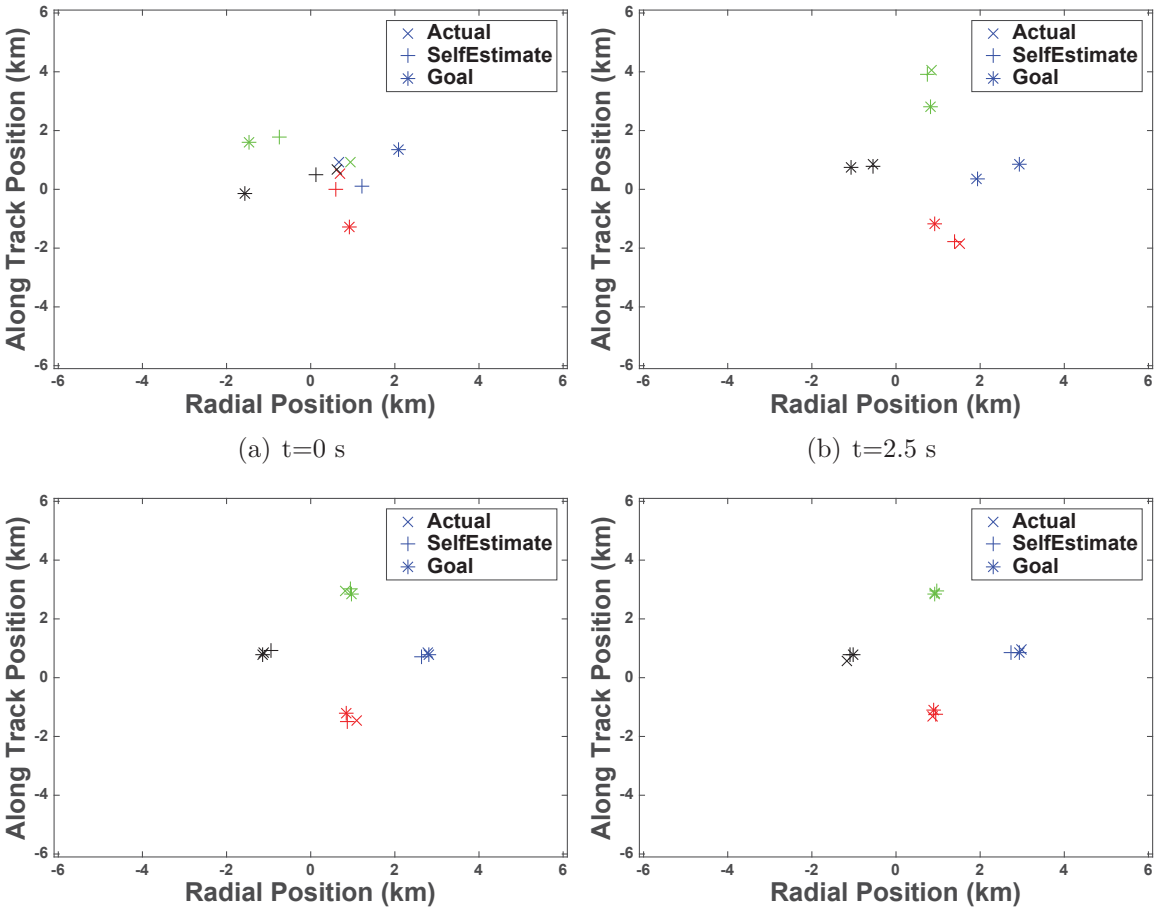


Figure 4.2: Graph representation of communication topology among agents

As seen in Figure 4.2 and 4.3, the formation is shown at every 2.5 s increments, illustrating that after 7.5 s the formation has indeed converged. For Figure 4.2 , each color corresponds to the characteristics of a single agent. In particular, the estimation markers represent the estimated location each spacecraft had for itself. On the other hand, in Figure 4.3, the estimation markers represents the estimate locations as perceived by satellite 1 (represented by the blue markers). Thus, it can be inferred that every estimate by every other satellite has also converged the corresponding true state value. Additional evidence of this can be seen in the final formation locations, which form the desired 4 by 4 square.



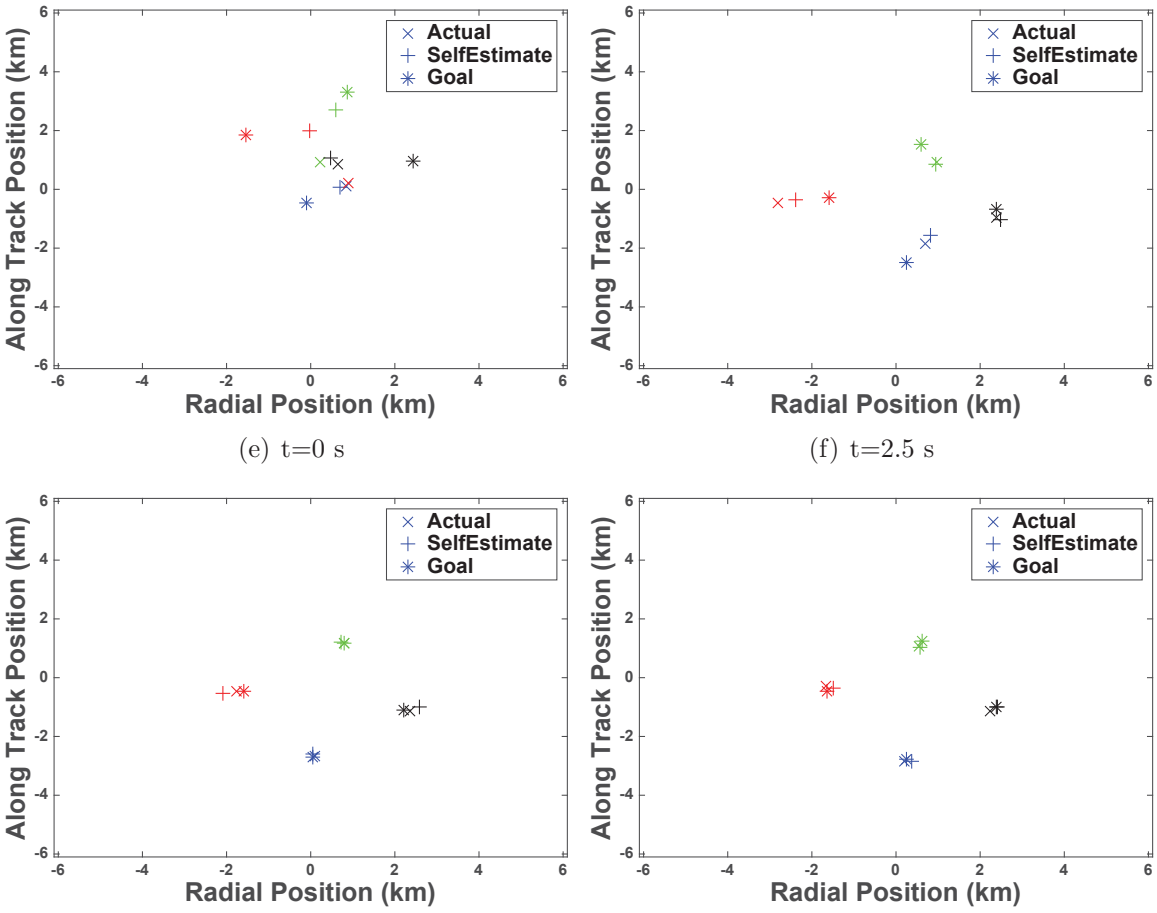
(a) t=0 s

(b) t=2.5 s

(c) t=5.0 s

(d) t=7.5 s

Figure 4.2: Convergence of self-estimate and states to assignment



(e) t=0 s

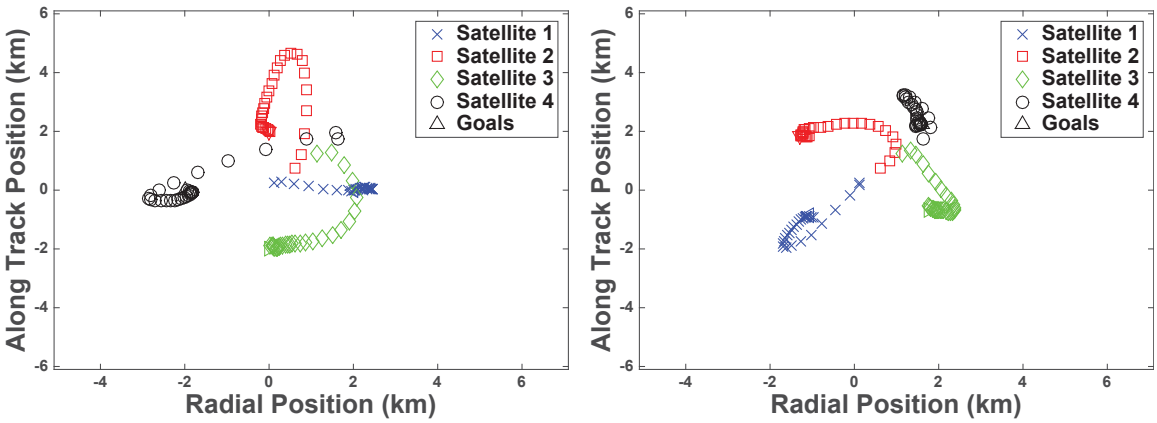
(f) t=2.5 s

(g) t=5.0 s

(h) t=7.5 s

Figure 4.3: Convergence of satellite 1 estimate and states to assignment

Since fuel is such a precious commodity for spacecraft, then reducing control effort would lead to increased operational time or decreased cost of deployment. Figure 6.2 illustrates two cases of a formation starting from the same nominal initial conditions, with the first case having a preset, goal locations and the other employing the distributed assignment algorithm. As seen in Figure 4.4(a) and 4.4(b), most of the satellites have to travel significantly farther to reach their intended goals. Figure 4.5 shows total control cost, in units of $|u|^2$, for both cases, illustrating the significant reduction in control effort.



(a) Preset goal locations (b) Distributed assignment employed

Figure 4.4: Trajectory history of formation

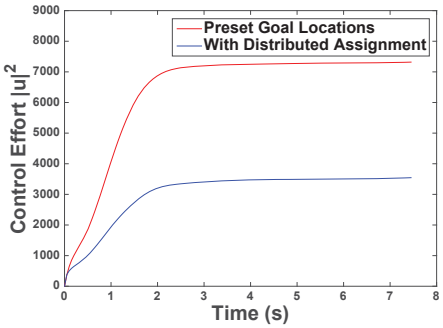


Figure 4.5: Control effort reduction via assignment algorithm

4.5 Summary

A closed loop coordination and control architecture has been shown to successfully not only keeping formation but also reducing the amount of effort needed to reach this desired formation. It is interesting to note that the assignment algorithm does provide ad-hoc collision avioidance as noted by McDonald [32]. Because the cost function of assignment is distance-based, then then intersecting trajectories are impossible in a linear cost function and with a quadratic cost function, they just become extremely unlikely. Therefore, an LQR controller still only solves an uncontrained optimization problem. In order to obtain guarantees on collision avoidance, then constraints must be added to the control law which would require either a controller based on linear programming or convex optimization. So far, considerations of how to actually measure the states of neighbors have not been elaborated on. The following chapter will propose additional non-linear filters in order to address this localization problem.

CHAPTER 5

Application to Non-linear Systems

In this chapter, we will consider an application of the distributed architecture in order to address a major engineering problem. Currently, the U.S. arsenal employs a vast array of uncontrolled munitions. In these weapon systems, adjustments to the firing initial conditions are made to account for visually observed error between desired target and actual impact point. Collateral damage in war not only affects the lives of the civilians, but can also be a detriment to morale for the soldiers. Currently, cruise missiles and air drone strikes are used to achieve precision targeting. While these methods have resulted in reduced collateral damage, they are also expensive to manufacture and maintain. In order to mitigate these cost concerns, a projectile can be equipped with canard control to create what is called a *smart munition*. Since it lacks an internal propulsion system, it would be considerably cheaper and less complex than most missiles. The challenge for these controlled munitions is the extreme environment in which these projectiles are launched, in particular the high acceleration experienced with values going as high as 200,000 g [46]. These high g forces make it extremely difficult for onboard sensors to work properly. Since the performance of any control system is intricately tied to the estimation of its state

parameters, a robust estimation estimation scheme is of paramount importance for this specific application.

5.1 Extended Kalman Filter

All the previous analysis have been conducted solely within the scope of linear dynamics and measurement. However most real systems need to be modeled via non-linear dynamics or measurements; therefore an extended kalman filter is necessary to capture the non-linearity of the system. For a complete nonlinear system described by equations 2.1 and 2.7, the extended kalman filter (EKF) greatly resembles its linear counterpart. Its state propagation and measurement are derived from the non-linear model, while its Kalman parameters are computed with the Jacobian of the non-linear system around the current estimate.

$$\dot{\hat{x}} = f(\hat{x}, u) + L(z - h(\hat{x}, u)) \tag{5.1}$$

$$L(t) = P(t)H^T R^{-1} \tag{5.2}$$

$$\dot{P} = A(t)P(t) + P(t)A^T(t) + Q - L(t)RL^T(t) \tag{5.3}$$

$$A(t) = \left.\frac{\partial f}{\partial x}\right|_{\hat{x}(t),u} \tag{5.4}$$

$$H(t) = \left.\frac{\partial h}{\partial x}\right|_{\hat{x}(t),u} \tag{5.5}$$

5.1.1 Orientation Estimation

Obtaining the orientation, or attitude, of a 3-D object is of utmost importance to both UAVs and spacecraft since their mission often relies on their sensors being pointed in the correct direction. Odometry, also called dead reckoning, is significantly

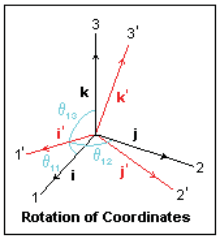
affected by changes in orientation, especially for rapidly rotating objects. However, unlike translation dynamics, where the three degrees of freedom are usually independent, the rotational equations of motion are highly non-linear due to the coupling between modes of rotation as seen in the second term of equation 5.6.

$$\mathbf{M} = I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{5.6}$$

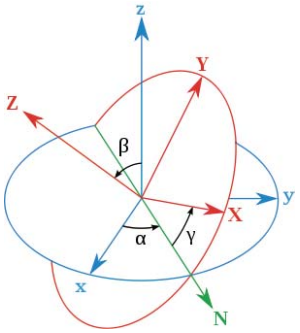
The attitude of extended object can be represented in many ways, all with their unique strengths. The most intuitive method is the direction cosine matrix (DCM), which is constructed from the cosine of angles between the global reference frame and the local one. Computation this method is not efficient because it has to keep track of six independent parameters (because the DCM is an orthogonal matrix). The two most used representations are Euler angles and quaternions. Euler angles express an attitude as a sequence of three rotations. For aerospace applications, the convention order is z-axis, y-axis, and then finally x-axis, and the rotation themselves are called yaw, pitch, and roll, respectively. However, there is a possibility of gimbal lock with this representation due to the loss of a degree of freedom. Quaternions are a quadruple generalization of complex numbers with three imaginary components instead of only one. Since it expresses an orientation as a rotation about a single axis, it does not suffer from gimbal lock but does have to satisfy a normalization condition to remain valid. Specific details on advantages are given in Table 5.1. Usually orientation is stored as a quaternion and is converted into the other forms as needed.

Table 5.1: Comparison of attitude representations

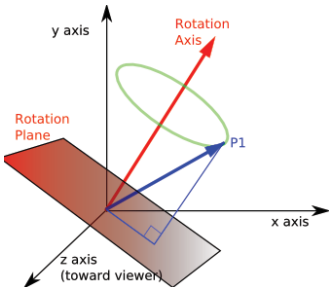
	Pros	Cons
DCM	Simplicity of sequential transformation Simple visualization	Computationally expensive
Euler Angles	Compatible with many control laws Least number of parameters	Gimbal Lock
Quaternions	No Gimbal lock Simplicity of sequential transformation	Not directly controllable Needs to be normalize



(a) Directional Cosines



(b) Euler Angles



(c) Quaternion

Figure 5.1: Representation of Orientation

5.2 Background

Accurate attitude estimation is a critical requirement for any moving vehicle and is typically accomplished by some combination of accelerometers, gyroscopes, and/or magnetometers. Because, accelerometers are the most adversely affected sensor during the launch, then gyro/mag extended kalman algorithm based on Bar and Oshman [47] is used as the baseline algorithm as shown in Algorithm 1, where $\hat{\mathbf{q}}$ is the current quaternion estimate, ϕ is the linearized transition matrix, $\delta\hat{\mathbf{q}}$ is the current quaternion error estimate. The basic premise of this algorithm is that it runs an extended Kalman filter for quaternion error and then renormalize to compute the actual quaternion.

1: **procedure** Bar-Oshman Algorithm(1)
2: Initialize $\mathbf{q}_0$
3: $P \leftarrow P_0$
4: **while** New data is available **do**
5: Between Measurements:
6: $\hat{\mathbf{q}}_{i+1/i} = \phi_i \hat{\mathbf{q}}^*_{i/i}$
7: $P_{i+1/i} = \phi_i P_{i/i} \phi_i^T + B_i Q_i B_i^T$
8: $\delta\hat{\mathbf{q}}_{i+1/i} = \phi_i \hat{\mathbf{q}}_{i/i-1} \hat{\mathbf{q}}^T_{i/i-1} \delta\hat{\mathbf{q}}_{i/i}$
9: Across Measurements:
10: $K_{i+1} = P_{i+1/i} H^T_{i+1/i} (H_{i+1/i} P_{i+1/i} H^T_{i+1/i} + R_{i+1/i})^{-1}$
11: $\delta\hat{\mathbf{q}}_{i+1/i+1} = \delta\hat{\mathbf{q}}_{i+1/i} + K_{i+1}(e_{i+1} - H_{i+1/i}\delta\hat{\mathbf{q}}_{i+1/i})$
12: Estimated Quaternion Reset and Normalization
13: $\hat{\mathbf{q}}_{i+1/i+1} = \hat{\mathbf{q}}_{i+1/i} + \delta\hat{\mathbf{q}}_{i+1/i+1}$
14: $\hat{\mathbf{q}}^*_{i+1/i+1} = \frac{\hat{\mathbf{q}}_{i+1/i+1}}{\|\hat{\mathbf{q}}_{i+1/i+1}\|}$

Update Covariance

$P_{i+1/i+1} = (I - K_{i+1}H^*_{i+1/i+1})P_{i+1/i}(I - K_{i+1}H^*_{i+1/i+1})^T + K_{i+1}R^*_{i+1/i+1}K^T_{i+1}$

$i + 1 \leftarrow i$

As gyroscopic measurements are of mission critical importance, then additional redundancies must be considered in case of failure. This naturally leads to a scenario where multiple IMU units are installed on the projectile and are operated independently. This ensures that failure of one IMU unit would not compromise the whole system. Subsequently some form of data fusion is necessary so that a single set of parameters is fed into the controller. Consensus protocols were initially used on multi-agent systems to allow distributed convergence and agreement of internal parameters [39]. Olfati-Saber [12] [13] then expanded these results into estimation by incorporating a consensus term into the standard Kalman filter. This particular distributed kalman filter first fuses all covariances and sensor data from an information perspective. This fused data is then injected into each local kalman filter.

procedure DISTRIBUTED KALMAN FILTER(2)

Initialize $\hat{x}_i(0|0) \leftarrow x(0)$

$P(0)_i \leftarrow P_0$

while New data is available **do**

Covariance Inverse Fusion and Sensor Data Fusion:

$S(k) = \frac{1}{n}\sum_{i=1}^{n} H_i^T(k)R_i^{-1}H_i(k)$

$y(k) = \frac{1}{n}\sum_{i=1}^{n} H_i^T(k)R_i^{-1}z_i(k)$

Observation Update for each agent i:

$M_i(k) = (P_i(k)^{-1} + S(k))^{-1}$

$\hat{x}_i(k+1|k) = \hat{x}_i(k|k) + M_i(k)(y(k) - S(k)\hat{x}_(k|k))$

State and Covariance Propagation

12: $P_i(k+1) = A_i M_i A_i^T + Q_i$

$\hat{x}_i(k+1|k+1) = A_i \hat{x}_i(k+1|k)$

14: $k+1 \leftarrow k$

Since each IMU unit would wired be into the projectile, then communication is not an issue, therefore a fully connected network of IMUs is going to be the base case that we are going to consider. The first failure mode under consideration is unmodeled white noise, which means noise covariances are significantly greater then originally modeled or assumed. As consensus is a weighted average over multiple observers, then if a couple of sensors fail in such a fashion, then overall performance would not be as adversely affected. The major drawback to use of consensus, especially in the case of model noise, is the estimation of ψ.

$$\dot{\phi} = p + (qsin(\phi) + rcos(\phi))tan(\theta) \tag{5.7}$$

$$\dot{\theta} = qcos(\phi) - rsin(\phi) \tag{5.8}$$

$$\dot{\psi} = (qsin(\phi) + rcos(\phi))sec(\theta) \tag{5.9}$$

An important observation is that the first two equation are coupled while ψ itself does not appear in any of the equations. Therefore, it is extremely sensitive to changes to the other two angles. Therefore ψ estimation would always accumulate errors. This problem is exacerbated with consensus as the errors are merged into the estimate with every time step. Therefore, ψ would be calculated separately using the current estimates and magnetometer data.

The second mode of failure considered is unobserved bias induced from the launch. In this case, consensus only would not be effective as errors would still accumulated

with time. This requires an external observer to estimate the bias online. As magnetometers are more robust to high acceleration environment, they are commonly used to correct bias [48, 49]. We choose to integrate the algorithm from Batista et al. [49] due to its simplicity and proof of convergence. Consider x_1 to be a known fixed vector in the body frame, x_2 to be the gyro bias and $S(x)y$ to be the matrix product corresponding to $x \times y$.

$$\begin{aligned} \dot{x_1} &= -S(\omega)x_1 + S(x_2)x_1 \\ \dot{x_2} &= \mathbf{0} \\ y(t) &= x_1 \end{aligned} \tag{5.10}$$

This system of equations can then be transformed into state space form, where $x = [x_1^T x_2^T]^T$.

$$\begin{aligned} \dot{x} &= A(y,\omega)x \\ y(t) &= Cx \\ A(y,\omega) &= \begin{bmatrix} -S(\omega) & -S(y) \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \\ C &= [\mathbf{I}\,\mathbf{0}] \end{aligned} \tag{5.11}$$

As Bar-Oshman's algorithm already utilizes magnetometers, then no additional sensors are required. As the nonlinearity of this system does not depend on the states but only measured outputs, Batista et al. were able to show that a Kalman filter based on this system is asymptotically stable.

A complete sketch of our proposed architecture is shown below in Figure 5.2.

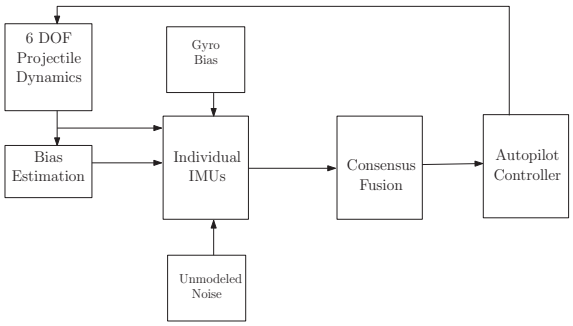


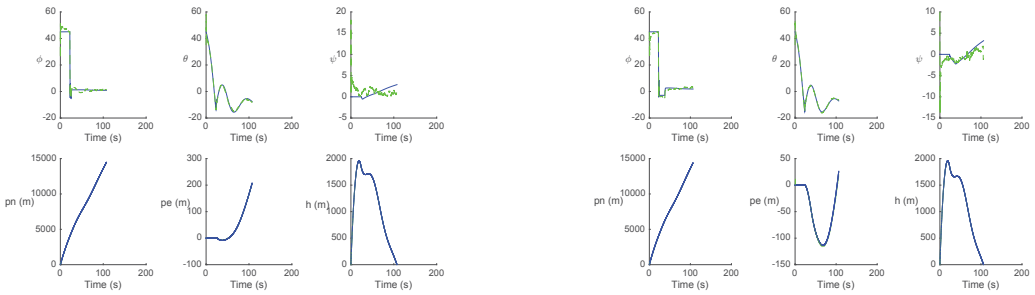
Figure 5.2: Graph representation of communication topology among agents

5.3 Simulations

The simulations are conducted for a projectile with a initial roll angle of 45°. After 12 seconds from launch, the canards would then deploy and the autopilot would first attempt to return the roll angle back to zero. Then it would adjust the pitch angle to extend the range as much as possible using a simple PID controller. To ease the transition between the multiple control modes, a ramp input was implemented to reduce drastic changes in aerodynamic parameters. The global position coordinates are calculated from a GPS Kalman filter and included in the data for completeness.

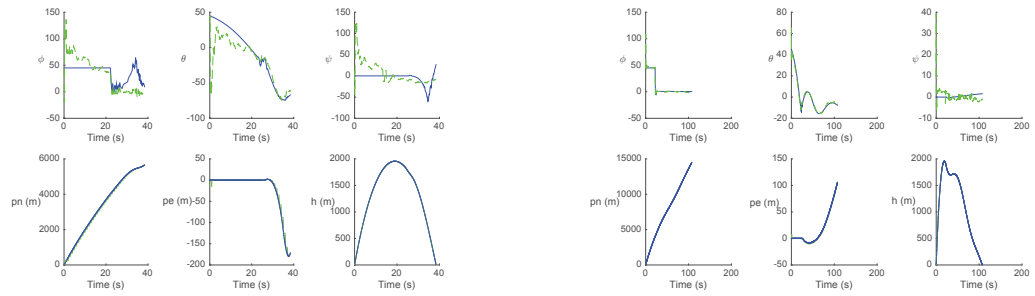
As seen in Figure 5.3 through 5.4, we consider the effect of changing the number of sensors. The solid blue plots represent true states while green dash lines represents the current estimate. In Figure 5.3, we consider the base line scenario in which all sensors are working properly. In both cases, they perform satisfactorily to estimate the true state. From here, we can see that adding additional IMUs after 3 would not improve the performance of the orientation estimator. However, in Figure 5.4, when there are 2 IMUs malfunction, and the noises coming into the IMUs are actually 50 times more than what the Kalman filter in each IMU is expecting. As one can see, having a larger population percentage of properly function IMUs adds redundancy to

the consensus estimation, which makes the system more resilient to failure of a few IMUs.



(a) 3 IMUs (b) 11 IMUs

Figure 5.3: Effect of increasing number of sensors - Base Case Simulation

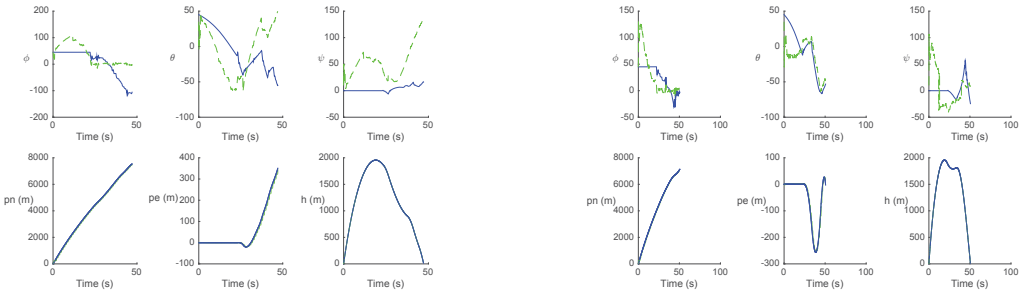


(a) 3 IMUs (b) 11 IMUs

Figure 5.4: Effect of increasing number of sensors - Two Malfunctioning IMUs

In the next scenario, this time a bias is introduced into the gyroscope at launch into each of the IMUs. Again, we will consider the case where two IMUs incorrectly outputs extremely noisy data with a total number of 11 IMUs. Figure 5.5 illustrates how inaccurate the estimator becomes if there is no bias compensation. By including the Batista's algorithm into the loop, one can see that effect of the bias is miti-

gated but not completely compensated. However, this illustrates that an integrated algorithm with online bias estimation is at least feasible.



(a) No Bias Estimation

(b) Integrated Bias Estimation

Figure 5.5: Effect of Compensated Bias

5.4 Summary

Under the extremely harsh operating conditions that a launch projectile endures, while having to control such a system, then it is necessary to address the various issues in an integrated approach. As each issue would usually have a specific subsystem to address them, then it is necessary to verify that they all function properly in the architecture loop. An integrated approach for addressing IMU failure is shown to be feasible and capable of addressing incorrect noise parameters and online bias estimation. In the following chapter this work will be extended to address the the global estimation problem in a GPS-denied environment.

CHAPTER 6

Over-determined Navigation

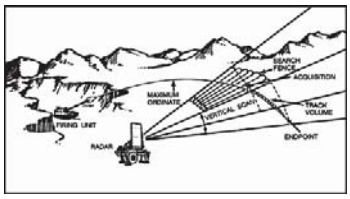
6.1 Overview

In most filtering schemes of estimation and navigation, a major assumption is the ability to combine information from dynamical processes with measurement corrections from an independent source. The major drawback to localization exclusively based on dynamic processes is the accumulation of errors because of the necessary integration of the dynamical equations (also called dead reckoning). Therefore, the external measurements serve to reset the integration error so that estimate does not drift to far from the true value. In most application, these external corrections, also called fiducials, are usually accomplished via GPS data, which was the case previously considered. Other fiducials typically used are radiation sources, ultrasound, or magnetic fields [50]. The particular fiducial is suited depending on the application. For example, for in-door navigation, a common method of navigation is called Simultaneous Localization and Mapping (SLAM), which sonar or point-radiation sources to collect data from the surroundings in order to create a map, which can then be used for navigational purposes. It relies on detecting distinctive features which it

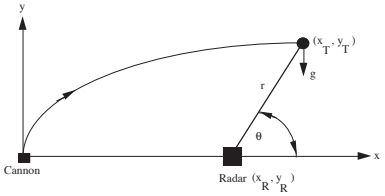
then uses as a reference marker in its map. Its obvious drawback would be areas with no distinct feature, like a long hallway.

6.2 Projectile Tracking

Radar technology has been developed to fill the void left by the lack of GPS [51,52]. By knowing the location of radar station and the radar's relative measurement,these projectile tracking systems (PTS) run an EKF where it is possible to reconstruct the global position from the linear global dynamics and nonlinear measurement. A directed radar ground station would be the most suitable for the conditions in consideration. Radar would be not be obscured by smoke and dust, as opposed to visible light. A ground station would be more energy and computational efficient than having an on-board point source and performing a SLAM algorithm, especially with the distance that would have to be traversed.



(a) PTS Illustration



(b) PTS Formulation

Figure 6.1: Projectile Tracking Problem

From Figure 6.1(b), we can deduce the geometric relationship from measurements given by the radar (r, θ) to global coordinates of the projectile (x_T, y_T). From here we can construct an measurement model so we can integrate it into our current architecture.

$$\begin{aligned} \mathbf{H} &= \begin{bmatrix} \frac{\partial \theta}{\partial x_T} & \frac{\partial \theta}{\partial \dot{x}_T} & \frac{\partial \theta}{\partial y_T} & \frac{\partial \theta}{\partial \dot{y}_T} \\ \frac{\partial r}{\partial x_T} & \frac{\partial r}{\partial \dot{x}_T} & \frac{\partial r}{\partial y_T} & \frac{\partial r}{\partial \dot{y}_T} \end{bmatrix} \\ &= \begin{bmatrix} \frac{-(y_T - y_R)}{r^2} & 0 & \frac{x_T - x_R}{r^2} & 0 \\ \frac{x_T - x_R}{r} & 0 & \frac{y_T - y_R}{r} & 0 \end{bmatrix} \end{aligned} \tag{6.1}$$

The final step is to incorporate the PTS into the simulation, while disabling GPS. Because our PTS model only incorporates range and altitude, we assume that a full 3D model is available and lateral displacement is included for completeness. We will utilize the last integrated case from the previous chapter, where there are 11 IMUs with two of them malfunctioning with excessive noise. Every IMU will have bias added to them and would include a bias estimator along with consensus. For the integration of the accelerometer, lateral displacement is estimated because due to the transformation from the body rates to global rates.

6.3 Simulations

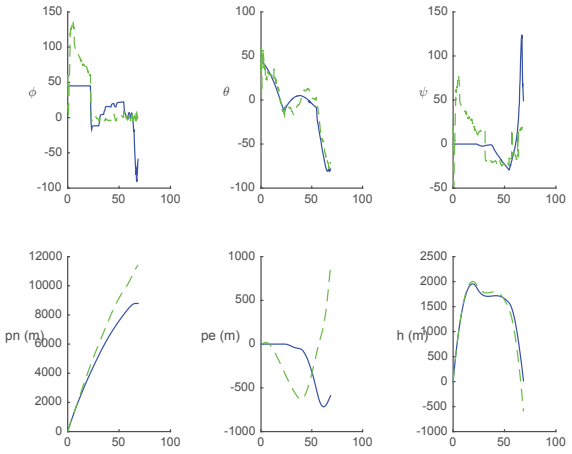
For the simulation we will consider three scenarios:

1. *No available PTS and solely by accelerometer integration*

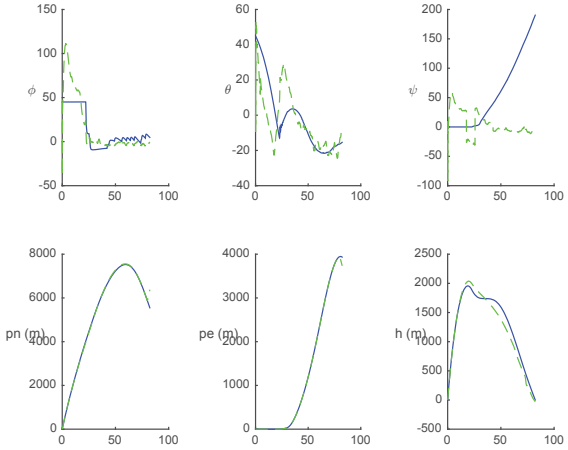
2. PTS available for 80 s and then accelerometer integration

3. PTS available until landing

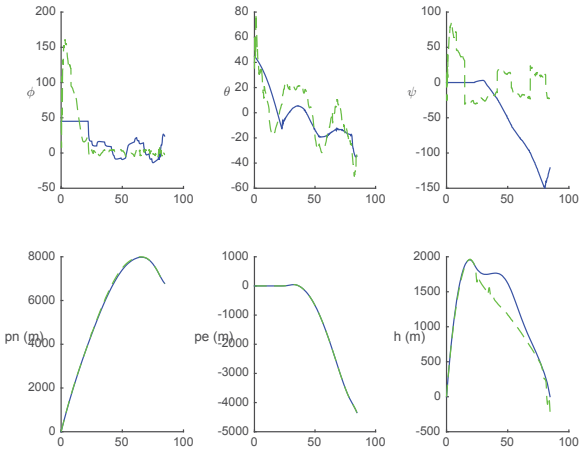
In the first scenario, there is no PTS available, and the projectile would have to perform dead reckoning by transforming the measured body-frame accelerometer data into the global frame via orientation estimate calculated previously. The transformed data would then be doubly integrated in order to obtain the global position. In the second scenario, PTS would only be allowed until 80 seconds (around the apogee of the trajectory) and then it would perform dead reckoning like in Scenario 1. In the final case, PTS would be completely available throughout the flight.



(a) No PTS



(b) PTS available for 80 s



(c) PTS available until landing

Figure 6.2: Integration of PTS

As expected, having the PTS available for a longer period improves the global estimation of the projectile. As seen in Figure 6.2(a), the accumulated errors of the accelerometer makes global estimation especially because of the compounded error from initially estimating the attitude. Having it available until landing is a rather unrealistic assumption, given distant objects that could disrupt the line of sight needed for the system to work. However, even a more realistic assumption such as PTS available until 80 s significantly improves the performance. This is due to the compounding of errors that from accelerometers that accumulate numerical errors and the residual errors from the previous attitude estimation scheme.

6.4 Summary

By integrating PTS into the architecture, a heterogeneous sensor network is established where complete global system variables are estimated using on-board and external sensors. Future extensions of this work would include a complete 3D PTS algorithm and bias compensation for accelerometers.

CHAPTER 7

Conclusions and Future Work

7.1 Completed Development and Impact

Within the realm of controls and algorithms, developers and engineers typically design from an input-output methodology, wherein inputs are exogenously generated in another subsystem, imported and processed in the current subsystem, and then exported out to other subsystems. This modularity keeps the design process simple and efficient, as subsystems are designed independently and then combined afterwards. However, the drawback to this methodology is the possibility for things to go wrong due to some unforeseen interaction(s) between the subsystems. Therefore, it is necessary to ensure that combining subsystems will indeed lead to the desire effects and consequences.

A goal of developing an integrated architecture for distributed aerospace systems is proposed and then analyzed. For the systems considered, a desired for precision and accuracy necessitated an architecture where all the system states have to be estimated. As a first step, an architecture for distributed estimation and control (Chapter 3) is shown to be effective at keeping formation for spacecraft, and a relaxed separation principle was proved for distributed systems. By combining guidance

via task assignment into the architecture and allowing the formation to adjust its orientation, an fully integrated architecture for distributed estimation, guidance and control (Chapter 4) was shown to be stable, and its benefit for spacecraft systems is shown, such as reduced bandwidth in communication and improved fuel economy.

Furthermore, a distributed sensor network of IMUs can be used to address the challenges of implementing control over a canard-actuated projectiles. Adverse effects to the gyroscopes can be mitigated with multiple IMU redundancy with consensus information fusion with magnetometers for bias correction (Chapter 5). Meanwhile an external laser tracking system could complete the estimation loop by providing an external global positioning reference in the absence of GPS (Chapter 6).

7.2 Future Work

Additional technical analysis is needed for relaxing assumptions on network topology and satisfying collision avoidance constraints. Current investigation in non-linear estimation will be instrumental in implementing an autonomous system with minimal external communication. For the canard-actuated projectile, the expansion of distributed techniques could be used to coordinate a swarm in order to maximize area of coverage.

From a hardware perspective, implementation could be conducted to validate the effectiveness of the architecture in a live test. The ground based units will use techniques discussed in the previous chapter in order localize themselves relative to the aerial vehicle and relative to each other. The aerial vehicle will act as the leader of the formation and will be the reference origin for each ground vehicle's estimate.

They will then transmit this information to their neighbors and perform consensus in order to build an estimate for the entire formation.

Ultimately this work primarily serves to illustrate the need for further insights of how to integrate subsystems within the context of distributed systems. These systems have shown, through coordination, new capabilities are possible even when it is not possible for any individual unit.

Bibliography

[1] F. Hadaegh, B. Açıkmeşe, D. Bayard, G. Singh, M. Mandic, S. Chung, and D. Morgan, "Guidance and control of formation flying spacecraft: From two to thousands," 2013.

[2] J. A. Roberts, "Satellite formation flying for an interferometry mission," pp. 1–4, 2005.

[3] H. Helvajian, "Microengineering aerospace systems," pp. 29–36, 1999.

[4] D. P. Scharf, F. Y. Hadaegh, and S. R. Ploen, "A survey of spacecraft formation flying guidance and control. part i: guidance," in *American Control Conference, 2003. Proceedings of the 2003*, vol. 2. IEEE, 2003, pp. 1733–1739.

[5] S. Saraf, A. Knoll, F. Pelletier, and M. Tafazoli, "Investigating formation flying and cots in an integrated simulation environment," in *Space Ops*, 2002.

[6] D. P. Scharf, F. Y. Hadaegh, and S. R. Ploen, "A survey of spacecraft formation flying guidance and control. part ii: control," in *American Control Conference, 2004. Proceedings of the 2004*, vol. 4. IEEE, 2004, pp. 2976–2985.

[7] M. Martin and M. Stallard, "Distributed satellite missions and technologies – the techsat 21 program," in *Space Technology Conference and Exposition*, 1999, pp. 28–30.

[8] D. Floreano and R. J. Wood, "Science, technology and the future of small autonomous drones," *Nature*, vol. 521, no. 7553, pp. 460–466, 2015.

[9] C. Boucher, A. Lahrech, and J.-C. Noyer, "Non-linear filtering for land vehicle navigation with gps outage," in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, vol. 2. IEEE, 2004, pp. 1321–1325.

[10] S. Rezaei and R. Sengupta, "Kalman filter-based integration of dgps and vehicle sensors for localization," *Control Systems Technology, IEEE Transactions on*, vol. 15, no. 6, pp. 1080–1088, 2007.

[11] B. Rao, H. F. Durrant-Whyte, and J. Sheen, "A fully decentralized multi-sensor system for tracking and surveillance," *The International Journal of Robotics Research*, vol. 12, no. 1, pp. 20–44, 1993.

[12] R. Olfati-Saber, "Distributed kalman filter with embedded consensus filters," in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*. IEEE, 2005, pp. 8179–8184.

[13] ——, "Distributed kalman filtering for sensor networks," in *Decision and Control, 2007 46th IEEE Conference on*. IEEE, 2007, pp. 5492–5498.

[14] S. Martinez, "Distributed interpolation schemes for field estimation by mobile sensor networks," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 491–500, 2010.

[15] S. Bandyopadhyay and S.-J. Chung, "Distributed estimation using bayesian consensus filtering," in *American Control Conference (ACC), 2014*. IEEE, 2014, pp. 634–641.

[16] J. P. Desai, J. P. Ostrowski, and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots," *IEEE transactions on Robotics and Automation*, vol. 17, no. 6, pp. 905–908, 2001.

[17] J. Hu and G. Feng, "Distributed tracking control of leader–follower multi-agent systems under noisy measurement," *Automatica*, vol. 46, no. 8, pp. 1382–1387, 2010.

[18] Y. Hong, G. Chen, and L. Bushnell, "Distributed observers design for leader-following control of multi-agent networks," *Automatica*, vol. 44, no. 3, pp. 846–850, 2008.

[19] R. Bachmayer and N. E. Leonard, "Vehicle networks for gradient descent in a sampled environment," in *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, vol. 1. IEEE, 2002, pp. 112–117.

[20] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[21] M. Rotkowitz and S. Lall, "A characterization of convex problems in decentralized control," *IEEE Transactions on Automatic Control*, vol. 51, no. 2, pp. 274–286, 2006.

[22] B. Açikmeşe and D. S. Bayard, "A markov chain approach to probabilistic swarm guidance," in *American Control Conference (ACC), 2012*. IEEE, 2012, pp. 6300–6307.

[23] W.-M. Shen, P. Will, A. Galstyan, and C.-M. Chuong, "Hormone-inspired self-organization and distributed control of robotic swarms," *Autonomous Robots*, vol. 17, no. 1, pp. 93–105, 2004.

[24] E. J. Hughes, "Swarm guidance using a multi-objective co-evolutionary online evolutionary algorithm," in *Evolutionary Computation, 2004. CEC2004. Congress on*, vol. 2. IEEE, 2004, pp. 2357–2363.

[25] M. Tillerson, L. Breger, and J. P. How, "Distributed coordination and control of formation flying spacecraft," in *Proc. IEEE American Control Conference*. Citeseer, 2003.

[26] A. Rantzer, "A separation principle for distributed control," in *Decision and Control, 2006 45th IEEE Conference on*. IEEE, 2006, pp. 3609–3613.

[27] R. Smith and F. Hadaegh, "Closed-loop dynamics of cooperative vehicle formations with parallel estimators and communication," *Automatic Control, IEEE Transactions on*, vol. 52, no. 8, pp. 1404–1414, 2007.

[28] A. Rahmani, O. Ching, and L. A. Rodriguez, "On separation principle for the distributed estimation and control of formation flying spacecraft," *Proceedings of Conference on Spacecraft Formation Flying Missions and Technologies*, 2013.

[29] T. Vu and A. Rahmani, "Distributed consensus-based kalman filter estimation and control of formation flying spacecraft: Simulation and validation," in *AIAA Guidance, Navigation, and Control Conference*, 2015, p. 1553.

[30] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Robotics-DL tentative*. International Society for Optics and Photonics, 1992, pp. 586–606.

[31] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image and vision computing*, vol. 10, no. 3, pp. 145–155, 1992.

[32] E. A. Macdonald, "Multi-robot assignment and formation control," Georgia Institute of Technology, 2011.

[33] M. Alighanbari and J. P. How, "Decentralized task assignment for unmanned aerial vehicles," in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*. IEEE, 2005, pp. 5668–5673.

[34] P. Milgrom, "Putting auction theory to work: The simultaneous ascending auction," *Journal of Political Economy*, vol. 108, no. 2, pp. 245–272, 2000.

[35] H.-L. Choi, L. Brunet, and J. P. How, "Consensus-based decentralized auctions for robust task allocation," *Robotics, IEEE Transactions on*, vol. 25, no. 4, pp. 912–926, 2009.

[36] D. Morgan, G. P. Subramanian, S.-J. Chung, and F. Y. Hadaegh, "Swarm assignment and trajectory optimization using variable-swarm, distributed auction assignment and sequential convex programming," *The International Journal of Robotics Research*, 2016.

[37] S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, "Inhomogeneous markov chain approach to probabilistic swarm guidance algorithm," in *5th Int. Conf. Spacecraft Formation Flying Missions and Technologies*, 2013.

[38] H. K. Khalil and J. Grizzle, *Nonlinear systems*. Prentice hall New Jersey, 1996, vol. 3.

[39] R. O. S. R. M. Murray, "Consensus protocols for networks of dynamic agents," in *Proceedings of the 2003 American Controls Conference*, 2003.

[40] J. H. Taylor, "The cramer-rao estimation error lower bound computation for deterministic nonlinear systems," in *Decision and Control including the 17th Symposium on Adaptive Processes, 1978 IEEE Conference on*, vol. 17. IEEE, 1978, pp. 1178–1181.

[41] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *Journal of Fluids Engineering*, vol. 83, no. 1, pp. 95–108, 1961.

[42] V. Kapila, A. G. Sparks, J. M. Buffington, and Q. Yan, "Spacecraft formation flying: dynamics and control," *Journal of Guidance, Control, and Dynamics*, vol. 23, no. 3, pp. 561–564, 2000.

[43] J.-J. E. Slotine and W. Li, "On the adaptive control of robot manipulators," *The international journal of robotics research*, vol. 6, no. 3, pp. 49–59, 1987.

[44] E. Jin and Z. Sun, "Robust controllers design with finite time convergence for rigid spacecraft attitude tracking control," *Aerospace Science and Technology*, vol. 12, no. 4, pp. 324–330, 2008.

[45] J. Yao, R. OrdÃ³Ã±ez, and V. Gazi, "Swarm tracking using artificial potentials and sliding mode control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 129, no. 5, pp. 749–754, 2007.

[46] W. Mermagen, "High'g'telemetry for ballistic range instrumentation," DTIC Document, Tech. Rep., 1964.

[47] I. Bar-Itzhack and Y. Oshman, "Attitude determination from vector observations: quaternion estimation," *Aerospace and Electronic Systems, IEEE Transactions on*, no. 1, pp. 128–136, 1985.

[48] N. Metni, J.-M. Pflimlin, T. Hamel, and P. Soueres, "Attitude and gyro bias estimation for a flying uav," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE, 2005, pp. 1114–1120.

[49] P. Batista, C. Silvestre, and P. Oliveira, "Partial attitude and rate gyro bias estimation: observability analysis, filter design, and performance evaluation," *International Journal of Control*, vol. 84, no. 5, pp. 895–903, 2011.

[50] L. Ojeda and J. Borenstein, "Personal dead-reckoning system for gps-denied environments," in *Safety, Security and Rescue Robotics, 2007. SSRR 2007. IEEE International Workshop on*. IEEE, 2007, pp. 1–6.

[51] M. Squire, H. Hyman, R. Trissel, G. Houghton, D. Leslie, and M. Dunn, "Projectile tracking system," Aug. 18 1998, uS Patent 5,796,474.

[52] J. Pinezich, J. Heller, and T. Lu, "Ballistic projectile tracking using cw doppler radar," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 46, no. 3, pp. 1302–1311, 2010.