2019

Decentralized Consensus-based Control Allocation For Some Dynamical Systems

August Mark
University of Central Florida

DECENTRALIZED CONSENSUS-BASED CONTROL ALLOCATION FOR SOME DYNAMICAL SYSTEMS

by

AUGUST MARK
B.S. University of Central Florida, 2013
M.S. University of Central Florida, 2015

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Mechanical and Aerospace Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2019

Major Professor: Yunjun Xu

ABSTRACT

In this dissertation, three separate studies, wherein techniques from graph theory and consensus control are used to address control allocation problems, are presented. In the first study, a decentralized allocator is presented for synthetic jet actuators and control surfaces onboard a small unmanned aerial vehicle to cooperatively generate desired aerodynamic moments. First order linear dynamics are assumed for both the synthetic jet actuators and control surfaces. A weighted consensus algorithm with limited feedback is used for the aerodynamic moment contribution allocator considering constraints. In the second study, the same allocation problem as in the first study is considered, but the actuator dynamics are now assumed to behave according to second order nonlinear dynamics. In the third study, a spray allocator is presented for an array of nozzles used to cool a large heated surface in order to address the local disagreement in surface temperature within sprayed sections. Within each study, the stability of each system is proven, and the performance of each allocator is demonstrated via simulations.

ACKNOWLEDGMENTS

I would like to thank Dr. Yunjun Xu for his advice, support, and friendship throughout my time at graduate school.

I would like to thank Dr. Benjamin T. Dickinson for his expert advice and criticism.

I would like to thank the rest of the faculty and staff at University of Central Florida for their instruction and the engineering skills they have passed on to me.

I would like to thank Dr. Kuo-Chi Lin, Dr. Jeffrey L. Kauffman, and Dr. Shawn Putnam for being on my dissertation committee.

I would like to thank my current lab members, Andong Dai, Xiangling Kong, Qiang Li, and Jonah Mapes, as well as my former lab members, He Shen, Ni Li, Puneet Vishwakarma, Sinem Defterli, Adrian Merchan, Pablo Menendez-Aponte, Christian Garcia, Douglas Freese, Pradens Pierre-Louis, Kenneth Thompson, and anyone I may have forgotten, for their insight, colaboration, and patience.

I would like to thank my parents and my sister for their continuous support.

I would like to thank my close personal friends, John, Jess, Chris, Meg, Adrian, Mike, Jenn, Nick,

Alex, and Daniel, for nearly 10 years of amazing camaraderie.

The work presented in Chapter 2 originally appeared in [1]. ©2018 IEEE. Reprinted, with permission, from Mark, A., Xu, Y., and Dickinson, B. T., "Consensus-based decentralized aerodynamic moment allocation among synthetic jets and control surfaces". IEEE Transactions on Control Systems Technology, August/2018.

The work presented in Chapter 3 originally appeared in [2]; reprinted by permission of the American Institute of Aeronautics and Astronautics, Inc.

The work presented in Chapter 4 originally appeared in [3].

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Distributed Flow Sensing and Actuation for SUASs

Naturally evolved fliers, such as birds, bats, and insects, are able to achieve graceful flight in a wide-range of flight regimes by directly sensing the airflow and controlling their current flight status through a distributed array of highly specialized organs. Mechanoreceptors located in the wings of birds are used to detect turbulences [4] and stall [5] and measure the air speed [6]. Short hairs located on the wings of bats are used to detect stall and make rapid course corrections to avoid collisions [7]. Wind-sensitive setae located on the heads of locusts are used to measure wind speed and direction [8], and similar sensory organs located on the legs of crickets are used to quickly detect oncoming predators via oncoming gusts [9].

In contrast to this distributive flow sensing approach, man-made aerial vehicles mainly rely on centralized sensing systems that measure the rigid body states of the vehicle. However, such approaches limit the agility and stability of small unmanned aerial systems (SUAS), a class of small, low weight, and low cost unmanned aerial systems. Due to their small size, wind perturbations can make SUAS unstable, leading to catastrophic failures that occur before a centralized sensing and control system can detect and rectify the situation purely based on rigid body state information. Therefore there is a growing interest to develop a distributed and local sensing and control strategy for use in these SUAS. Recent developments have been made in the field of micro-sensors to provide this needed distributed sensing capability. Advances in Micro-Electrical-Mechanical Systems technologies have enabled the development of small pressure sensors [10] and hair sensors that mimic the wind-sensitive hairs on many flying creatures [11]. Increasing efforts have been placed on integrating distributed direct flow sensing systems using these sensors into SUASs, as such sensing systems can be used to extract aerodynamic parameters such as angle of attack [12–14],

sideslip angle [14], airspeed [13–15], lift [12, 16], moment [12], and flow features like flow separation [15, 17] and stagnation points [15]. Additionally, as such a system can directly observe the local surface flow conditions, it is anticipated that SUASs with this capability will be able to react to disturbances faster than traditional rigid body sensors like IMUs, greatly improving SUAS stability and viability for tasks in a variety of flight regimes [18].

As increased effort is placed on directly measuring the local flow, directly manipulating the local flow also becomes interesting. Dielectric barrier discharge plasma actuators have been demonstrated within the laboratory setting, and can achieve flow reattachment [17]. Synthetic jet actuators (SJAs) have been used for a variety of tasks, such as delaying flow separation [19, 20], addressing limit cycle oscillations in SUASs [21], or acting as replacements for traditional control surfaces [22, 23]. To this end, it is proposed that an array of distributed pressure sensors with SJAs could be used for active flight control over an SUAS. Along with the micro-scale sensor [24] and actuator development [25], one challenge from the control perspective is to how to rapidly compute the contributions among SJAs and three major control surfaces (i.e. aileron, rudder, and elevator) so the overall aerodynamics moment and/or force is desired.

While controllers utilizing airflow information on SUAS surfaces measured through distributed sensing arrays [26–28] have already been proposed, the actuation of these controllers typically relies on a limited set of control surfaces. SJAs provide an opportunity for a distributed actuation scheme based on local flow information, but also impose the need for a scheme that will quickly collect distributed flow information, compute the required aerodynamic moments and forces, and then allocate control commands to the distributed actuators.

1.2 Spray Cooling

Spray cooling is used as a method for cooling processes and equipment with thermal dissipation needs which cannot be economically satisfied by air cooling or liquid cooling means. Within manufacturing, spray cooling has been widely used in metallurgical processes including continuous casting processes, cool mill roll processes, and hot strip mill processes [29–31]. Additionally, this method can lead to reduced tool wear rates and tool failure rates in micro-machining processes [32]. Within electronics, particularly in supercomputing applications, spray cooling has been proposed as a method to deal with rising heat dissipation needs imposed by ever increasing computing power and ever decreasing chip sizes [33, 34]. It has also been proposed for various high performance electronic systems within hybrid electric vehicles in order to decrease the size and increase the economic viability of these vehicles [34–37]. In addition to these primary applications, spray cooling has also been proposed for cooling laser systems [34], rail guns [38], radar systems [39], and spacecraft [34, 40].

Within the body of research regarding spray cooling, most efforts focus on either developing algebraic models to predict heat flux out of a sprayed surface or on improving heat flux generated through the introduction of surfactants to the fluid or microstructures to the sprayed surface. Additionally, most efforts focus on single spray nozzle cases and on small spray areas (on the order of $1cm^2$), rather than on arrays of multiple nozzles or over larger spray areas. This will become a pressing concern in the future, as it is estimated that power generation and conversion systems, as well as directed energy, communication, and surveillance systems will require high heat fluxes over surface areas measuring in the hundreds of cm^2 [41, 42]. While a single nozzle can cover a larger area if it is sufficiently far away, this is not practical for many applications where space is limited [43].

Instead, an array of multiple nozzles can be used, but this introduces several ramifications. (1)

The spacing between individual nozzles must be carefully considered. If the nozzles are too close, the total area sprayed is decreased due to overlapping sprays. If the nozzles are too far apart, the total heat flux decreases as the sprays cover a larger area while the total number of droplets remains constant [43]. (2) In many applications, it is required that the surface temperature of the sprayed surface remain uniform. However it is seen that non-uniformity in the surface temperature increases when using multiple nozzles at higher heat fluxes [41,44]. Non-uniformity in the temperature can lead to spatial variances in thermal and mechanical properties, and can allow warping to manifest during the solidification of molten metals. (3) The local interaction between sprays must be considered, as otherwise the sprayed liquid cannot drain from the surface, leading to the creation of a liquid film which the sprayed droplets can have difficulty penetrating [41, 42, 44], mitigating thermal dissipation. In [45], the sprayed surface is inclined, and the liquid is naturally able to drain off the surface via gravity; however this further worsens the temperature non-uniformity. In [42], the spray is carefully optimized to ensure the sprayed liquid always vaporizes when impacting the surface, but controlling the spray to ensure this occurs while achieving desired heat fluxes can be very challenging.

1.3 Control Allocation

Control allocation, as a relatively mature research area, has been extensively studied over the past two decades as a means to distribute calculated control commands to redundant actuators while satisfying additional criteria such as minimizing energy consumption or computational time [46]. Since the majority of control allocation studies focus on systems involving a limited number of actuators, control allocation problems are typically viewed as offline constrained optimization problems [47]. In such a centralized scheme, every agent (e.g. SJAs and control surfaces) is able to communicate directly to the CPU, which can then optimally allocate control commands

to generate the desired aerodynamic moment. Different optimization algorithms have been investigated for both linear and nonlinear actuation systems, and these include mixed integer linear programming [46], enhanced interior-point algorithms [48], least squares and robust least squares algorithms [49], linear matrix inequality approaches [50], and multiparametric programming [51]. Stability issues have also been investigated to compensate for the mismatch between desired and allocated control commands [52]. However, many of these types of methods do not scale well as the number of actuators increases, due to the fact that the computational capacity available for many applications is limited. As an alternative approach, techniques from consensus control can be learned to solve the control allocation problem in a decentralized way.

A consensus algorithm describes any protocol applied to a multiagent system with the intention of driving one or more of the information states of the system's agents to a common value [53]. The field of consensus control has also been well-explored [54]. Consensus algorithms are typically used to achieve consensus among cooperative vehicles, and can be used in systems with nonlinear dynamics that are poorly-characterized [55], with model uncertainty [56], with constrained control [57], with time delays [58], with constraints (e.g. collision avoidance) [59], or with switching topologies [60]. Consensus algorithms are not without drawbacks; in the absence of external feedback information, the agents in a consensus system will converge on a final configuration that depends on their initial configuration. This final configuration cannot be predicted without knowing the initial configuration and may itself be undesirable [61]. Depending on how complex the communication topology is, the agents must be able to receive information from many sources.

1.4 Dissertation Outline

The rest of this dissertation is outlined as follows. In Chapter 2, a decentralized, consensus based moment allocation scheme for an SUAS equipped with traditional control surfaces and a redundant

array of SJAs with 1st order linear actuator dynamics is presented. In Chapter 3, the moment allocation scheme in the previous chapter is expanded to consider 2nd order nonlinear actuator dynamics. In Chapter 4, two decentralized, consensus based spray allocation schemes for spray cooling are presented. Finally, in Chapter 5, concluding remarks are given, and possible future work is outlined.

CHAPTER 2: CONSENSUS BASED, DECENTRALIZED AERODYNAMIC MOMENT ALLOCATION AMONG SYNTHETIC JETS AND CONTROL SURFACES

2.1 Chapter Outline

While consensus algorithms have been widely studied for groups of unmanned aerial vehicles in formation flight for predefined formations [62], [63], the scheme in studied in this chapter is primarily concerned with the macroscopic goal of achieving a desired moment among actuators via consensus within a single SUAS. In addition to the aforementioned rationales of designing a consensus based allocation method instead of a centralized, optimization based method, the contributions of this work are: (1) to the best of the authors' knowledge, it is the first time a consensus allocation algorithm is designed and analyzed for the new application of consensus among SJAs and control surfaces within a SUAS; (2) a weighting scheme, modified from those presented in [53, 64], is introduced to enable the SJAs and control surfaces to reach consensus while considering their respective capabilities; and (3) the conditions to maintain the effectiveness of this weighting scheme are presented.

In this study, a distributed, feedback-type allocation scheme, distinct from the previously mentioned centralized optimization-based approaches, is used to achieve weighted consensus in aero-

dynamic moment control allocation between the SJAs and control surfaces. This scheme uses the local pressure information to determine the necessary contribution from the SJAs to achieve the desired overall moment while reducing the communication complexity and the computational requirements on a central microcontroller. The rationales behind this decision are as follows: (1) As the number of onboard SJAs increases, centralized schemes will add extra weight (e.g. communication wires), increase the construction complexity, and may randomly shift the center of gravity due to the need for additional wiring. (2) For a large number of SJAs, both the communication complexity and the computational cost imposed on a CPU in a centralized scheme will be very high. (3) The proposed consensus based control allocation is in a feedback form using real-time sensor information. Compared with optimization-based control allocation methods, this allocation scheme is more robust to model uncertainty and noise (e.g. the associated noise for the local flow information around the SJAs). (4) The stability of the proposed allocation consensus is proven and the allocation convergence is guaranteed, which cannot be said for most nonlinear constrained optimization approaches. (5) Via tuning the control gains, the convergence speed can be improved.

The rest of this chapter is organized as follows. In Section 2.2, the SUAS platform is introduced. An invariant mapping function for calculating the aerodynamic moments from pressure and shear information is discussed. First order models for the moment generated by SJAs and control surfaces are derived. Finally, the moment allocation problem is defined. In Section 2.3, relevant background topics from Graph Theory are introduced. In Section 2.4, a state-space model is derived for the moment generated by the SJAs, the moment generated by the control surface, and the error between the current and desired moments. A decentralized, constrained moment allocation controller capable of achieving the desired moment, in consensus, is then introduced, and the stability of this controller is proven. In Section 2.5, the consensus moment allocation controller proposed in Section 2.4 is tested and compared with a centralized approach.

2.2 Small Unmanned Aerial Vehicle Platform with Distributed Synthetic Jet and Flow Sensor Arrays

2.2.1 Small Unmanned Aerial System Platform

Recently, many SUAS designs have taken inspiration from biologically evolved flyers like birds and bats, with the hope of mimicking the agile and stable flights their nature counterparts can achieve in a turbulent airflow environment. As discussed in [4–7], there are distributive flow sensors (e.g. mechanoreceptors) densely packed within feather follicles throughout the wings of birds and skins of bats for the purpose of measuring flow speed, flow separation, stall, and other flow features. As shown in Fig. 2.1, based on real-time flow information, joints can actuate to influence the close proximity flow and achieve the desired aerodynamic forces and moments.

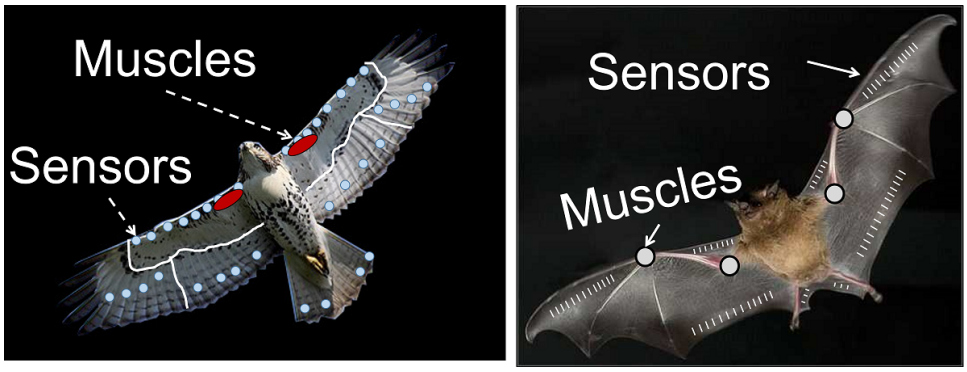

Figure 2.1: Common sensor and muscle locations on (a) birds and (b) bats

The following conceptual SUAS platform, intended to mimic birds and bats, is studied in this paper, and is shown in Fig. 2.2. (1) It is assumed that the wing-fuselage area is separated into N sections, and each section is covered by $N_i^s, i = 1, ..., N$ microscale flow sensors measuring pressure and shear stresses in real-time. (2) There is one SJA in each section to perturb the local pressure field

when needed, and it is assumed that the pressure field is perturbed uniformly over this section. (3) The control surfaces (i.e. aileron, rudder, and elevator) are still the main contributors for aircraft three-axis control. It is worth noting that the control surface areas are not considered part of the above mentioned N sections.

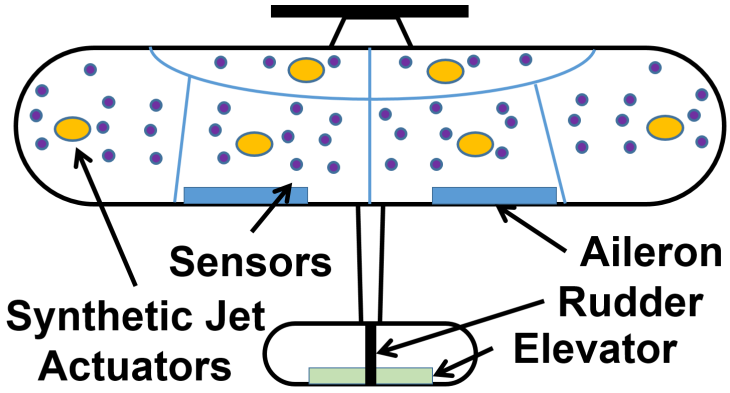

Figure 2.2: Sensor Locations on SUAS

2.2.2 *Invariant Mapping Function for Aerodynamic Moments*

As shown in [27], the aerodynamic moment contribution $\vec{m}_{i,j}$ from a small area $s_{i,j}$ covered by the j^{th} microscale sensor in section i can be approximated as

$$\vec{m}_{i,j} = s_{i,j}[\vec{r}_{i,j} \times (R_{i,j}\vec{q}_{i,j})] \tag{2.1}$$

where $i = 1, ..., N$, $j = 1, ..., N_i^s$, $\vec{m}_{i,j} \in \mathbb{R}^3$ is the aerodynamic moment in the body frame, $\vec{r}_{i,j} \in \mathbb{R}^3$ is the position vector of sensor j within section i with respect to the center of mass of the SUAS in the body frame, $R_{i,j}$ is the rotation matrix from the surface coordinate system to the body coordinate system, and $\vec{q}_{i,j} = [p_{i,j}, \tau_{i,j}^x, \tau_{i,j}^y]^T \in \mathbb{R}^3$ is the vector for the pressure, x-direction

shear stress, and y-direction shear stress at sensor j within section i in the surface coordinate. For a fixed-wing SUAS, Eq. (2.1) can be simplified by noting that all the terms except $\vec{q}_{i,j}$ are geometry related constant terms. These terms can be replaced by an invariant mapping function as discussed in [18] as

$$\vec{m}_{i,j} = M_{i,j}\vec{q}_{i,j} \tag{2.2}$$

where $M_{i,j}$ is the mapping function at sensor j within section i, calculated as

$$M_{i,j} = s_{i,j}\tilde{\vec{r}}_{i,j}R_{i,j} \tag{2.3}$$

where $\tilde{\vec{r}}_{i,j}$ is the skew symmetric matrix of $\vec{r}_{i,j}$. The invariant mapping function will be separated into three column vectors as

$$M_{i,j} = \begin{bmatrix} \vec{M}^p_{i,j} & \vec{M}^x_{i,j} & \vec{M}^y_{i,j} \end{bmatrix} \tag{2.4}$$

and then the aerodynamic moment can be rewritten in order to isolate the moment contribution of the pressure from the shear stresses as

$$\vec{m}_{i,j} = \vec{M}^p_{i,j}p_{i,j} + \vec{M}^x_{i,j}\tau^x_{i,j} + \vec{M}^y_{i,j}\tau^y_{i,j} \tag{2.5}$$

2.2.3 First Order Models for Synthetic Jet Actuator and Control Surface

SJAs are vibrating diaphragms that alternately suck in and eject air in order to increase the momentum of the surrounding air stream. SJAs have been successfully used in the past to control flow separation [19] and to suppress limit cycle oscillations [21]. In this study, all the SJAs are modeled as identically controlling the local pressure field according to a simple first order ODE [65] as

$$\dot{u}_i = -c_1 u_i + c_2 \epsilon_i \tag{2.6}$$

where u_i is the change in pressure at section i due to the SJA, c_1 and c_2 are positive constants, and ϵ_i is the voltage applied to the SJA. It can be seen that, if no voltage is supplied to an SJA, the output from that SJA will decay exponentially with a time constant of $1/c_1$, whereas if the change in pressure needs to be held constant, a voltage of $c_1 u_i / c_2$ is needed to cancel out the effects of this decay. If an SJA is present at a section i, Eq. (2.5) can be appropriately modified to represent the total aerodynamic moment contribution within a section i, $\vec{m}_i$, as

$$\vec{m}_i = \vec{M}_i^p u_i + \sum_{j=1}^{N_i^s} (\vec{M}_{i,j}^p p_{i,j} + \vec{M}_{i,j}^x \tau_{i,j}^x + \vec{M}_{i,j}^y \tau_{i,j}^y) \tag{2.7}$$

where $\vec{M}_i^p \in \mathbb{R}^3$ is the equivalent of $\vec{M}_{i,j}^p$ but for the total area affected by the SJA at section i. The effects of the distributed pressures, x-direction shear stresses, and y-direction shear stresses can be combined to simplify Eq. (2.7), as

$$\vec{m}_i = \vec{M}_i^p (p_i + u_i) + \vec{M}_i^x \tau_i^x + \vec{M}_i^y \tau_i^y \tag{2.8}$$

where $\vec{M}_i^p p_i$, $\vec{M}_i^x \tau_i^x$, and $\vec{M}_i^y \tau_i^y$ are the moments caused by the approximate pressures, x-direction shear stresses, and y-direction shear stresses acting uniformly over section i, and are equivalent to the respective summations in Eq. (2.7).

The control surface (i.e. elevator, aileron, and rudder) deflections can also be modeled as first order ODEs

$$\dot{\delta}_l = -c_3 \delta_l + c_4 \epsilon_{N+l} \tag{2.9}$$

where $\delta_l, l = 1, 2, 3$ is the deflection of the control surface l, c_3 and c_4 are positive constants, and ϵ_{N+l} is the voltage applied to the control surface l. The aerodynamic moment contribution of the

combined control surface deflection is

$$\vec{m}^{cs} = \vec{m}^{cs,0} + M^{cs}\vec{\delta},\ \vec{\delta} = \begin{bmatrix} \delta_1 & \delta_2 & \delta_3 \end{bmatrix}^T \tag{2.10}$$

where $\vec{m}^{cs} \in \mathbb{R}^3$ is the roll, pitch, and yaw aerodynamic moment due to the control surface deflections, $\vec{m}^{cs,0} \in \mathbb{R}^3$ is the moment with no control surface deflection, and $M^{cs} \in \mathbb{R}^{3\times3}$ is the control derivative matrix. In typical aircraft flight scenarios, both $\vec{m}^{cs,0}$ and M^{cs} are known ahead of time via CFD analyses, wind tunnel tests, and flight tests. It is worth noting that due to the coupling between roll and yaw, M^{cs} is not a diagonal matrix. However, for typical aircraft configurations, the structure of M^{cs} is such that it is invertible.

It should be noted that the first order models selected for the actuator dynamics are ideal representations of the true dynamics, and as an example the control surface model ignores the effects of control saturation. However, for now these simplified models are sufficient for demonstrating the concept of consensus moment allocation across an SUAS.

2.2.4 Aerodynamic Moment Allocation Problem

A consensus algorithm will be designed to allocate desired aerodynamic moment contributions among distributed SJAs and control surfaces. It is assumed there are N SJAs and three control surfaces on an SUAS, and the summation of the aerodynamic moment contribution from them should asymptotically approach the desired value $\vec{m}^d \in \mathbb{R}^3$ as

$$\vec{e} = \vec{m}^d - \sum_{i=1}^{N} \vec{m}_i - \vec{m}^{cs}, \lim_{t\to\infty} \|\vec{e}(t)\| = 0 \tag{2.11}$$

Two constraints are considered: (1) The information flow should follow the communication topology to be defined in Section III, and (2) the N SJAs can only actuate along the corresponding $\vec{M}_i^p$

vectors.

2.3 Graph Based Communication Architecture Analysis

Each actuator, including each individual SJA and the set of control surfaces, is assumed to be one "agent" but with a different constraint. They are connected via a fixed communication topology and are assumed to have no time delay. Based on graph theory [61], these $N+1$ "agents" on the SUAS surface, are contained by the vertex set $\mathcal{V}$, represented as

$$\mathcal{V} = \begin{Bmatrix} \nu_1 & ... & \nu_{N+1} \end{Bmatrix} \tag{2.12}$$

The $N+1$ vertices within $\mathcal{V}$ are connected via an edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. If agent i can receive state information from agent j, then $(\nu_j, \nu_i) \in \mathcal{E}$. The degree of a vertex, $d(\nu_i), i = 1, ..., N+1$, represents the number of edges within $\mathcal{E}$ that connect to ν_i, i.e. the number of agents that agent i can receive information from. The degree matrix D of a graph is the diagonal matrix whose diagonal entries contain the degree of each vertex in $\mathcal{V}$, represented as

$$D = diag \begin{Bmatrix} d(\nu_1) & ... & d(\nu_{N+1}) \end{Bmatrix} \tag{2.13}$$

The adjacency matrix $A = [a_{i,j}]$ is defined such that

$$a_{i,j} = \begin{cases} 1 & \text{if } (\nu_j, \nu_i) \in \mathcal{E} \\ 0 & \text{else} \end{cases} \tag{2.14}$$

The graph Laplacian, L, is then defined as

$$L = D - A \tag{2.15}$$

A typical consensus algorithm [61] uses the following state-space form

$$\dot{\vec{x}} = -L\vec{x} \tag{2.16}$$

where $\vec{x}$ is a vector, with the goal that $\vec{x}$ asymptotically approaches a consensus value as

$$\lim_{t\to\infty} \vec{x}(t) = c\vec{1} \tag{2.17}$$

where c is a constant, whose value depends on $\vec{x}(0)$ and the properties of the communication topology, and $\vec{1}$ is a vector whose elements are all 1. A successful consensus algorithm requires the graph Laplacian, representing the communication topology, to have a simple zero eigenvalue with all other eigenvalues having positive real parts. This can be achieved in undirected graphs if the graph is connected, and in directed graphs if a directed spanning tree is present [61]. From [61], it is seen that the convergence rate is related to the smallest positive eigenvalue of L.

In this study, a Nearest Neighbor communication topology, an undirected topology similar to that discussed in [66], is used, and is shown in Fig. 2.3. In the Nearest Neighbor topology, the agents are connected in a linear chain, and agents can exchange information with either of their neighbors, but agents at the beginning and end of the communication chain cannot "loop"; that is, agents 1 and $N + 1$ do not share a direct connection. The first agent directly receives information from the CPU, and this information is passed down from agent to agent. The agent directly connected with the CPU is the nearest SJA to the CPU. The eigenvalues of L for the graph Laplacian have non-negative real parts and only a simple eigenvalue at zero, as the graph is undirected and connected.

Therefore consensus is reached using this topology. This topology was chosen in order to address the limitations imposed by the small size and low weight of SUAS. Extra wiring and length of wires will increase the weight of an SUAS, move the location of the center of gravity, and increase the integration difficulty.

Figure 2.3: Nearest Neighbor Communication Topology

2.4 Decentralized Aerodynamic Moment Allocation

2.4.1 State-Space Representation of Aerodynamic Moment Allocation Dynamics

The error between the desired aerodynamic moment, $\vec{m}^d$, and the sum of the individual moments can be rewritten using Eq. (2.8), Eq. (2.10), and Eq. (2.11) as

$$\vec{e} = \vec{m}^d - \sum_{i=1}^{N} (\vec{M}_i^p p_i + \vec{M}_i^x \tau_i^x + \vec{M}_i^y \tau_i^y) - \sum_{i=1}^{N} \vec{M}_i^p u_i - \vec{m}^{cs,0} - M^{cs}\vec{\delta} \tag{2.18}$$

The difference between the desired moment and the moments caused by the pressures, shear stresses, and zero control surface deflections can be combined to form an augmented desired moment that is meant to be achieved through the SJAs and control surface deflections, $\vec{m}^e$, as

$$\vec{m}^e = \vec{m}^d - \sum_{i=1}^{N} (\vec{M}_i^p p_i + \vec{M}_i^x \tau_i^x + \vec{M}_i^y \tau_i^y) - \vec{m}^{cs,0} \tag{2.19}$$

and Eq. (2.18) can be rewritten as

$$\vec{e} = \vec{m}^{e} - \sum_{i=1}^{N} \vec{m}_i^{p} - \vec{m}^{csd} \tag{2.20}$$

where $\vec{m}_i^p$ is the moment at section i caused by the increase in pressure due to the SJA i and $\vec{m}^{csd}$ is the moment caused by the combined control surface deflections. $\vec{m}_i^p$ is calculated by

$$\vec{m}_i^{p} = \vec{M}_i^{p} u_i, i = 1, ..., N \tag{2.21}$$

and $\vec{m}^{csd}$ is

$$\vec{m}^{csd} = M^{cs}\vec{\delta} \tag{2.22}$$

The vector of the pressure moments from each SJA, $\vec{m}^p \in \mathbb{R}^{3N}$, can be defined as

$$\vec{m}^{p} = \begin{bmatrix} \vec{m}_1^{p,T} & ... & \vec{m}_N^{p,T} \end{bmatrix}^T \tag{2.23}$$

Using Eq. (2.6), Eq. (2.21), and Eq. (2.23), $\dot{\vec{m}}^p$ can be expressed as

$$\dot{\vec{m}}^{p} = -c_1\vec{m}^{p} + c_2\vec{\eta} \tag{2.24}$$

where $\vec{\eta} \in \mathbb{R}^{3N}$ is the vector of applied voltages such that

$$\vec{\eta} = \begin{bmatrix} \vec{M}_1^{p,T}\epsilon_1, & ..., & \vec{M}_N^{p,T}\epsilon_N \end{bmatrix}^T \tag{2.25}$$

Using Eq. (2.9) and Eq. (2.22), $\dot{\vec{m}}^{csd}$ can be expressed as

$$\dot{\vec{m}}^{csd} = -c_3\vec{m}^{csd} + c_4 M^{cs}\vec{\epsilon}^{cs} \tag{2.26}$$

where

$$\vec{\epsilon}^{cs} = \begin{bmatrix} \epsilon_{N+1} & \epsilon_{N+2} & \epsilon_{N+3} \end{bmatrix}^T \tag{2.27}$$

Equations (2.24) and (2.26) can then be combined into a state space representation as

$$\begin{bmatrix} \dot{\vec{m}}^p \\ \dot{\vec{m}}^{csd} \end{bmatrix} = - \begin{bmatrix} c_1 I_{3N} & 0 \\ 0 & c_3 I_3 \end{bmatrix} \begin{bmatrix} \vec{m}^p \\ \vec{m}^{csd} \end{bmatrix} + \begin{bmatrix} c_2 I_{3N} & 0 \\ 0 & c_4 M^{cs} \end{bmatrix} \begin{bmatrix} \vec{\eta} \\ \vec{\varepsilon}^{cs} \end{bmatrix} \tag{2.28}$$

The total error from the desired moment can then be written using Eq. (2.20) and Eq. (2.23) as

$$\vec{e} = \vec{m}^e - J \begin{bmatrix} \vec{m}^p \\ \vec{m}^{csd} \end{bmatrix} \tag{2.29}$$

where $J = \vec{1}^T_{N+1} \otimes I_3$, where $\otimes$ is the Kronecker product. J is then the matrix composed of $(N+1)$ 3×3 identity matrices repeated horizontally.

2.4.2 Decentralized Regulation Controller

Define $K \in \mathbb{R}^{3(N+1)\times 3(N+1)}$, to be a weighting matrix inspired by [64] [53], as

$$K = \begin{bmatrix} K_1 & 0 & ... & 0 \\ 0 & K_2 & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & K_{N+1} \end{bmatrix} \tag{2.30}$$

in which $K_i, i = 1, ..., N+1$ is a 3×3 diagonal matrix, where

$$K_i = k_i I_3, i = 1, ..., N+1 \tag{2.31}$$

Here $k_i > 0, i = 1, ..., N+1$, is a user-defined weighting scalar, chosen to influence the final moment generated by agent i, based on its constraint.

Remark 2.1: Each k_i weighting scalar for K is selected considering the moment constraint of agent i. If an agent is expected to produce a larger moment, then the corresponding value in K should be given a relatively lower magnitude.

Let $\hat{M}_i^p$ be the unit vector of $\vec{M}_i^p$ as

$$\hat{M}_i^p = \frac{\vec{M}_i^p}{\|\vec{M}_i^p\|}, i = 1, ..., N \tag{2.32}$$

Let $\hat{M} \in \mathbb{R}^{3(N+1)\times 3(N+1)}$ be defined as

$$\hat{M} = \begin{bmatrix} \hat{M}_1^p \hat{M}_1^{p,T} & ... & 0 & 0 \\ 0 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \hat{M}_N^p \hat{M}_N^{p,T} & 0 \\ 0 & ... & 0 & I_3 \end{bmatrix} \tag{2.33}$$

Lemma 2.1: K is commutative with $\hat{M}$ (i.e. $K\hat{M} = \hat{M}K$). Additionally, $\hat{M} \begin{bmatrix} \vec{m}^{p,T} & \vec{m}^{csd,T} \end{bmatrix} = \begin{bmatrix} \vec{m}^{p,T} & \vec{m}^{csd,T} \end{bmatrix}$.

Proof: Due to the block diagonal structure of both K and $\hat{M}$, and the diagonal structure of the

block matrices within K, $K\hat{M}$ can be written as

$$K\hat{M} = \begin{bmatrix} k_1\hat{M}_1^p\hat{M}_1^{p,T} & ... & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & k_N\hat{M}_N^p\hat{M}_N^{p,T} & 0 \\ 0 & ... & 0 & k_{N+1}I_3 \end{bmatrix} \tag{2.34}$$

Similarly, $\hat{M}K$ can be written as

$$\hat{M}K = \begin{bmatrix} k_1\hat{M}_1^p\hat{M}_1^{p,T} & ... & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & k_N\hat{M}_N^p\hat{M}_N^{p,T} & 0 \\ 0 & ... & 0 & k_{N+1}I_3 \end{bmatrix} \tag{2.35}$$

Thus, the first half of Lemma 2.1 holds true. For the second half, Eq. (2.21) and Eq. (2.23) can be used to express $\hat{M}\begin{bmatrix} \vec{m}^{p,T} & \vec{m}^{csd,T} \end{bmatrix}^T$ as

$$\hat{M}\begin{bmatrix} \vec{m}^p \\ \vec{m}^{csd} \end{bmatrix} = \begin{bmatrix} \hat{M}_1^p\hat{M}_1^{p,T}\vec{M}_1^p u_1 \\ \vdots \\ \hat{M}_N^p\hat{M}_N^{p,T}\vec{M}_N^p u_N \\ \vec{m}^{csd} \end{bmatrix} \tag{2.36}$$

As $\hat{M}_i^{p,T}\vec{M}_i^p = \left\|\vec{M}_i^p\right\|, i = 1, ..., N$ and $\hat{M}_i^p\left\|\vec{M}_i^p\right\| = \vec{M}_i^p, i = 1, ..., N$, it is then clear that the second half of Lemma 2.1 holds true.

Lemma 2.2: The matrix

$$F_1 = \hat{M}\left(L \otimes I_3\right)\hat{M} \tag{2.37}$$

is positive semi-definite.

Proof: Recall that L was selected such that its eigenvalues all have non-negative real parts with a simple zero eigenvalue. Additionally, L is symmetric. Thus, L is positive semi-definite, and, by extension, $L \otimes I_3$ is positive semi-definite. Also, note that $\hat{M}$ is symmetric, as it is a block-diagonal matrix composed of symmetric blocks. Thus, F_1 can be expressed as

$$F_1 = \hat{M}\left(L \otimes I_3\right)\hat{M}^T \tag{2.38}$$

It is clear then that F_1 consists of a positive semi-definite matrix multiplied on the left by a matrix and on the right by the transpose of that same matrix. Therefore, F_1 is positive semi-definite.

Lemma 2.3: The matrix

$$F_2 = K^{-1}J^T J K^{-1} \tag{2.39}$$

is positive semi-definite.

Proof: Suppose both a vector $\vec{x} \in \mathbb{R}^n$ and a matrix $X \in \mathbb{R}^{m \times n}$ exist. Then $\vec{x}^T X^T X \vec{x}$ is equivalent to $\left(X\vec{x}\right)^T X\vec{x}$, and this product is greater than 0 for all $X\vec{x} \neq \vec{0}$ and equals 0 only when $X\vec{x} = \vec{0}$ (that is, either $\vec{x}$ is the eigenvector of X associated with a zero eigenvalue or $\vec{x} = \vec{0}$). Therefore, any matrix times its transpose is at least positive semi-definite. As $K^{-1}J^T J K^{-1} = \left(JK^{-1}\right)^T\left(JK^{-1}\right)$, F_2 is positive semi-definite.

Definition 2.1: Actuator consensus is said to have been achieved between the $N+1$ actuators

when

$$\hat{M}\left(L \otimes I_3\right) K \begin{bmatrix} \vec{m}^p \\ \vec{m}^{cs} \end{bmatrix} = \vec{0} \tag{2.40}$$

Let $\vec{m}^f \in \mathbb{R}^{3(N+1)}$ be defined as

$$\vec{m}^f = \left[u_1 \vec{M}_1^{p,T} \quad \ldots \quad u_N \vec{M}_N^{p,T} \quad \vec{\delta}^T M^{cs,T} \right] \Big|_{t=\infty}^{T} \tag{2.41}$$

where $u_i|_{t=\infty}, i = 1, ..., N$ and $\vec{\delta}\,|_{t=\infty}$ are the steady state values for the N SJAs and the steady state control surface deflections, respectively. Let $f_1 \in \mathbb{R}^{3(N+1)}$ be defined as

$$\vec{f_1} = \left(L \otimes I_3\right) K \vec{m}^f \tag{2.42}$$

Then Eqs. (2.30) and (2.41) can be substituted into Eq. (2.42) as

$$\vec{f_1} = \left(L \otimes I_3\right) \left[\begin{array}{c} k_1 \vec{M}_1^p u_1 \\ \vdots \\ k_N \vec{M}_N^p u_N \\ k_{N+1} M^{cs} \vec{\delta} \end{array} \right] \Bigg|_{t=\infty} \tag{2.43}$$

which can be rewritten as

$$\vec{f_1} = \left[\vec{g}_1^T \quad \ldots \quad \vec{g}_N^T \quad \vec{g}_{N+1}^T \right]^T \tag{2.44}$$

where $\vec{g}_i, i = 1, \ldots, N+1$ is equal to the weighted actuation consensus error between actuator i

and the actuators it can communicate with. Define $\vec{f_2} \in \mathbb{R}^{3(N+1)}$ as

$$\vec{f_2} = \hat{M}\vec{f_1} \tag{2.45}$$

which can be rewritten using Eq. (2.44) as

$$\vec{f_2} = \begin{bmatrix} \left(\hat{M}_1^p \hat{M}_1^{p,T} \vec{g}_1\right)^T & \dots & \left(\hat{M}_N^p \hat{M}_N^{p,T} \vec{g}_N\right)^T & \vec{g}_{N+1}^T \end{bmatrix}^T \tag{2.46}$$

Suppose a vector $\vec{z} \in \mathbb{R}^n$ exists such that it is perpendicular to a vector $\vec{y}$. Then it is clear that $\vec{y}\vec{y}^T\vec{z} = \vec{0}$. It should also be clear that $\vec{y}\vec{y}^T\vec{0} = \vec{0}$. Then, let one criteria for choosing the weights in K be that each $\vec{g}_i \in \mathbb{R}^3, i = 1, ..., N+1$ is either equal to $\vec{0}$ or is perpendicular with $\hat{M}_i^p$. That is, $\hat{M}_i^{p,T}\vec{g}_i = \vec{0}, i = 1, \dots, N$ and $\vec{g}_{N+1} = \vec{0}$. Then it is clear that $\vec{f_2}$ is equal to $\vec{0}$.

The problem of choosing appropriate values for K, such that actuator consensus is achieved and the desired moment is achieved then becomes a nonlinear programming problem, defined as

$$\begin{aligned} \min \quad & \sum_{i=1}^{N} |u_i| \Big|_{t=\infty} + \sum_{i=1}^{3} |\delta_i| \Big|_{t=\infty} \\ \text{subject to} \quad & \vec{m}^e - J\vec{m}^f = \vec{0} \\ & \hat{M}_i^{p,T}\vec{g}_i = 0, i = 1, ..., N \\ & \vec{g}_{N+1} = \vec{0} \\ & u_i|_{t=\infty} \in \mathbb{R}, i = 1, ..., N \\ & \delta_i|_{t=\infty} \in \mathbb{R}, i = 1, 2, 3 \\ & k_i \in \mathbb{R}_+, i = 1, ..., N+1 \end{aligned} \tag{2.47}$$

Remark 2.2: It can be seen that the above NLP imposes $N + 6$ constraints on the selection of

$k_i, i = 1, ..., N+1$, $u_i|_{t=\infty}, i = 1, ..., N$, and $\delta_i|_{t=\infty}, i = 1, 2, 3$. However, there are $2N + 4$ unknowns available to satisfy these requirements. Thus, a solution for the $N+1$ unknown k_i terms that drives both the final error to zero and the moments to consensus may not be unique. However, a result of this non-uniqueness is that if numerical methods are used to find an optimal K matrix, the same K matrix can be used to achieve consensus for different $\vec{m}^e$ vectors if the communication topology is left unchanged.

Remark 2.3: If $\vec{m}^e$ is non-zero and numerical methods are used to solve for the $2N+4$ unknowns, particular care should be taken to prevent k_{N+1} from approaching zero, as this will result in trivial solutions where the aerodynamic moment is produced solely by the control surfaces. It is worth mentioning that *fmincon* in MATLAB can be used to find the numerical solution for the NLP.

Remark 2.4: Due to the nonlinearity of the NLP in Eq. (2.47), guaranteeing the existence of a solution to the NLP is quite challenging. However, finding a solution can be simplified by relaxing the equality constraints on actuator consensus and the error minimization to be inequality constraints. The authors have tested the NLP under the NN communication topology with a wide range of potential $\vec{m}^e$ values, and solutions to the NLP could be easily found in each case when using these relaxed constraints.

Lemma 2.4: If a solution to the NLP in Eq. (2.47) exists, then $\vec{e}$ can be expressed as

$$\vec{e} = -JK^{-1}K \left(\begin{bmatrix} \vec{m}^p \\ \vec{m}^{csd} \end{bmatrix} - \vec{m}^f \right) \tag{2.48}$$

Proof: Begin by rewriting Eq. (2.29) as

$$\vec{e} = \vec{m}^e - J\left(\begin{bmatrix}\vec{m}^p \\ \vec{m}^{csd}\end{bmatrix} - \vec{m}^f + \vec{m}^f\right) \tag{2.49}$$

Because a solution to Eq. (2.47) exists, it can be said that $\vec{m}^e = J\vec{m}^f$. Then Eq. (2.49) can be simplified as

$$\vec{e} = -J\left(\begin{bmatrix}\vec{m}^p \\ \vec{m}^{csd}\end{bmatrix} - \vec{m}^f\right) \tag{2.50}$$

Then, by including $K^{-1}K$, it can be seen that Eq. (2.48) holds true.

Theorem 2.1: Driven by the following aerodynamic moment allocation controller

$$\begin{aligned}\begin{bmatrix}\vec{\eta} \\ \vec{\varepsilon}^{cs}\end{bmatrix} = \begin{bmatrix}c_2 I_{3N} & 0 \\ 0 & c_4 M^{cs}\end{bmatrix}^{-1} \Bigg(& -c_5 \hat{M}\left(L \otimes I_3\right) K \begin{bmatrix}\vec{m}^p \\ \vec{m}^{csd}\end{bmatrix} \\ & + \begin{bmatrix}c_1 I_{3N} & 0 \\ 0 & c_3 I_3\end{bmatrix}\begin{bmatrix}\vec{m}^p \\ \vec{m}^{csd}\end{bmatrix} + c_6 K^{-1}\hat{M}J^T\vec{e}\Bigg)\end{aligned} \tag{2.51}$$

where c_5 and c_6 are positive tuning constants, the closed-loop aerodynamic moment allocation dynamics are asymptotically stable. $\vec{e}$ will asymptotically approach the origin and the $N+1$ weighted moment vectors will asymptotically reach consensus. It is important to note that $\hat{M}$ forces the applied moment from each SJA to act along the direction of $\hat{M}_i^p$, thus acting as a constraint. The controller defined in Eq. (2.51) can be written as sub controllers using Eqs. (2.21), (2.22), (2.23),

(2.25), (2.30) and (2.33) and multiplying through the matrix term to rewrite Eq. (2.51) as

$$\begin{bmatrix} c_2\vec{M}^p\epsilon_1 \\ \vdots \\ c_2\vec{M}^p_N\epsilon_N \\ c_4M^{cs}\vec{\epsilon}^{cs} \end{bmatrix} = -c_5\hat{M}\left(L\otimes I_3\right)\begin{bmatrix} k_1\vec{M}^pu_1 \\ \vdots \\ k_N\vec{M}^p_Nu_N \\ k_{N+1}M^{cs}\vec{\delta}^{cs} \end{bmatrix} + \begin{bmatrix} c_1\vec{M}^pu_1 \\ \vdots \\ c_1\vec{M}^p_Nu_N \\ c_3M^{cs}\vec{\delta}^{cs} \end{bmatrix} + \begin{bmatrix} c_6k_1^{-1}\hat{M}^p_1\hat{M}^{p,T}_1\vec{e} \\ \vdots \\ c_6k_N^{-1}\hat{M}^p_N\hat{M}^{p,T}_N\vec{e} \\ c_6k_{N+1}^{-1}\vec{e} \end{bmatrix} \tag{2.52}$$

Next, the definition of the graph Laplacian can be used, and Eq. (2.52) can be broken down, allowing the sub controllers to be found as

$$\begin{aligned} \epsilon_i = \frac{c_5\vec{M}^{p,T}_i}{c_2\left\|\vec{M}^p_i\right\|^2}\Bigg(\sum_{j=1}^{N}A_{i,j}\left(k_j\vec{M}^p_ju_j - k_i\vec{M}^p_iu_i\right) \\ +A_{i,N+1}\left(k_{N+1}M^{cs}\vec{\delta} - k_i\vec{M}^p_iu_i\right)\Bigg) + \frac{c_1}{c_2}u_i + \frac{c_6\vec{M}^{p,T}_i}{c_2k_i\left\|\vec{M}^p_i\right\|^2}\vec{e}, i = 1, ..., N \end{aligned} \tag{2.53}$$

and

$$\vec{\epsilon}^{cs} = \frac{c_5}{c_4}\sum_{j=1}^{N}A_{N+1,j}\left(k_jM^{cs,-1}\vec{M}^p_ju_j - k_{N+1}\vec{\delta}\right) + \frac{c_3}{c_4}\vec{\delta} + \frac{c_6}{c_4k_{N+1}}M^{cs,-1}\vec{e} \tag{2.54}$$

Proof: After applying Eq. (2.51), Eq. (2.28) becomes

$$\begin{bmatrix} \dot{\vec{m}}^p \\ \dot{\vec{m}}^{csd} \end{bmatrix} = -c_5\hat{M}\left(L\otimes I_3\right)K\begin{bmatrix} \vec{m}^p \\ \vec{m}^{csd} \end{bmatrix} + c_6K^{-1}\hat{M}J^T\vec{e} \tag{2.55}$$

Assume that a solution to the NLP was found. Then an $\vec{m}^f$ exists such that $\hat{M}\left(L\otimes I_3\right)K\vec{m}^f = \vec{0}$,

and Eq. (2.55) can be expressed as

$$\begin{bmatrix} \dot{\vec{m}}^p \\ \dot{\vec{m}}^{csd} \end{bmatrix} = -c_5 \hat{M} \left(L \otimes I_3\right) K \left(\begin{bmatrix} \vec{m}^p \\ \vec{m}^{csd} \end{bmatrix} - \vec{m}^f \right) + c_6 K^{-1} \hat{M} J^T \vec{e} \tag{2.56}$$

Now, define $\vec{\mu} \in \mathbb{R}^{3(N+1)}$ as

$$\vec{\mu} = K \left(\begin{bmatrix} \vec{m}^p \\ \vec{m}^{csd} \end{bmatrix} - \vec{m}^f \right) \tag{2.57}$$

Note, though, that $\dot{\vec{m}}^f = \vec{0}$. Then, after using Eq. (2.48) and Eq. (2.57), Eq. (2.56) can be expressed in terms of $\vec{\mu}$ as

$$\begin{bmatrix} \dot{\vec{m}}^p \\ \dot{\vec{m}}^{csd} \end{bmatrix} = -c_5 \hat{M} \left(L \otimes I_3\right) \vec{\mu} - c_6 K^{-1} \hat{M} J^T J K^{-1} \vec{\mu} \tag{2.58}$$

Then, take a time derivative of Eq. (2.57) and use Eq. (2.58) to arrive at

$$\dot{\vec{\mu}} = -c_5 K \hat{M} \left(L \otimes I_3\right) \vec{\mu} - c_6 \hat{M} J^T J K^{-1} \vec{\mu} \tag{2.59}$$

Now, select a Lyapunov candidate function V as

$$V = \frac{1}{2} \vec{\mu}^T K^{-1} \vec{\mu} \tag{2.60}$$

Note that $k_i > 0, i = 1, ..., N+1$, and as K is diagonal it is positive definite and its matrix inverse is also positive definite. Thus Eq. (2.60) is a positive definite function. Next, take a time derivative

of Eq. (2.60)

$$\dot{V} = -\vec{\mu}^T \left(c_5 \hat{M} \left(L \otimes I_3 \right) + c_6 K^{-1} \hat{M} J^T J K^{-1} \right) \vec{\mu} \tag{2.61}$$

Recall from Lemma 2.1 that K commutes with $\hat{M}$, and that $\hat{M}\vec{\mu} = \vec{\mu}$. Then Eq. (2.61) can be rewritten as

$$\dot{V} = -\vec{\mu}^T \left(c_5 \hat{M} \left(L \otimes I_3 \right) \hat{M} + c_6 K^{-1} J^T J K^{-1} \right) \vec{\mu} \tag{2.62}$$

From Lemmas 2.2 and 2.3, it can be seen that the inner matrix terms in Eq. (2.62) are positive semi-definite, and thus Eq. (2.62) is a negative semi-definite function. $\dot{V} \leq 0$ for all $\vec{\mu}$ and $\dot{V} = 0$ only when both $\hat{M} \left(L \otimes I_3 \right) \vec{\mu} = \vec{0}$ and $JK^{-1}\vec{\mu} = \vec{0}$. These conditions occur, respectively, once both actuator consensus has been achieved and $\vec{e} = \vec{0}$, or once $\vec{\mu} = \vec{0}$ Both are trivial cases. Thus, according to the global invariant set theorem in [67], the system will asymptotically converge to the set of $\vec{\mu}$ values that satisfy these conditions, and the closed-loop system is asymptotically stable. Note that $\vec{\mu}$ does not necessarily have to equal $\vec{0}$, as the solution to the NLP in Eq. (2.47) is not necessarily unique.

2.5 Simulation Results and Discussion

The simulated SUAS is initially in flight at 20 m/s with $\alpha = 0^\circ$ and $\beta = 0^\circ$. A trimmed elevator deflection of -13° is initially given, with no other control surfaces initially deflected. Ten sections are assumed to be placed on the SUAS surface, with each section containing an SJA directly in its center. The parameters describing the simulated SUAS are listed in Table 2.1. A subroutine

programmed in MATLAB is used to generate the mapping functions based on the SUAS configuration and locations of the SJAs. A wireframe image of the SUAS, including the locations of SJAs, is given in Fig. 2.4. It is noted that the aerodynamic moments over the wing is approximated as in Eq. (2.8). With the advancement of MEMS flow sensors, more sensors can be embedded on the surface of a wing, and the accuracy can be improved via an online calibration method as proposed in [18].

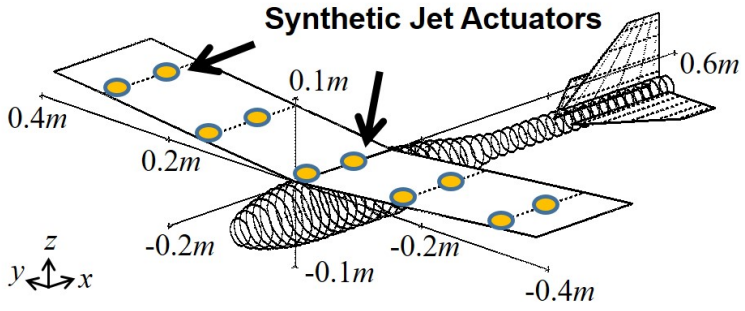

Figure 2.4: Wireframe SUAS with SJA Locations

Table 2.1: SUAS Parameters Used in the Simulation

Wing Airfoil	NACA 4412	**Dihedral**	5°
Wing Span	$0.762\ m$	**Wing Chord**	$0.152\ m$
Aileron Length	$0.231\ m$	**Aileron Chord**	$0.038\ m$
Rudder Airfoil	NACA 0012	**Rudder Root Chord**	$0.152\ m$
Rudder Height	$0.100\ m$	**Rudder Tip Chord**	$0.050\ m$
Elevator Airfoil	NACA 0012	**Elevator Offset**	$0.4223\ m$
Elevator Span	$0.152\ m$	**Elevator Tip Chord**	$0.050\ m$
Elevator Root Chord	$0.152\ m$		
Inertia Matrix	$\begin{bmatrix} 0.0526 & 0 & 0.0049 \\ 0 & 0.35 & 0 \\ 0.0049 & 0 & 0.0405 \end{bmatrix}$		kgm^3

The NN communication topology is given an $\vec{m}^e = [1, 0.5, 1]\, Nm$ for roll, pitch, and yaw mo-

ments, respectively. The simulation results are shown in Figs. 2.5 and 2.6. The final values for $u_i, i = 1, ..., N$ and $\vec{\delta}$, as well as those found from the NLP, are shown in Table 2.2.

As mentioned in Remark 2.3, the *fmincon* function in MATLAB is used to determine an optimal K matrix. c_1 and c_3 are set to 0.01, while c_2 and c_4 are set to 5. c_5 is set to 1 and c_6 is set to 10. The simulation is performed for 2 s with a fixed time step of 0.01 s using a Runga-Kutta integration scheme. When solving the NLP, trivial initial conditions selected for the actuation values as $u_i|_{t=\infty} = 0, i = 1, ..., N$ and $\vec{\delta}\,|_{t=\infty} = \vec{0}$. The initial K values are equal to the final K values, which can be seen in Table 2.3.

Table 2.2: NLP Solution Versus Final Simulated Values

	NLP Solution	**Final**
$u_1\,(Pa)$	0	-24.1032
$u_2\,(Pa)$	0	-22.8916
$u_3\,(Pa)$	0	-19.3915
$u_4\,(Pa)$	0.795×10^{-3}	-15.9264
$u_5\,(Pa)$	0	-159.9104
$u_6\,(Pa)$	0	6.9768
$u_7\,(Pa)$	-1.2015×10^{-3}	20.1837
$u_8\,(Pa)$	0	13.6383
$u_9\,(Pa)$	0	77.0311
$u_{10}\,(Pa)$	0.1489×10^{-3}	12.2971
$\vec{\delta}\,(^\circ)$	$\begin{bmatrix} 3.3190 \\ -0.0541 \\ 1.4055 \end{bmatrix}$	$\begin{bmatrix} 0.3005 \\ -0.1602 \\ 2.6596 \end{bmatrix}$

Table 2.3: Weights within K

k_1	k_2	k_3	k_4	k_5	k_6
0.4868	0.4524	0.3835	0.2716	0.1517	0.6106
k_7	k_8	k_9	k_{10}	k_{11}	
0.2698	0.5536	0.1372	1.0000	0.0500	

From Fig. 2.5, it can be seen that the NN communication topology is capable of quickly achieving the desired moment, with rise times of $0.6204s$, $0.1973s$, and $0.6418s$ in the roll, pitch, and yaw directions, respectively.

From Fig. 2.6, it can be seen that the actuator outputs stabilize evenly and reach weighted consensus, as desired. Note that the control surfaces have no difficulty producing moments along all three directions, and produce half of the desired roll moment and nearly all of the desired pitch and yaw moments. Meanwhile, the SJAs produce the remaining half of the roll moment, very little pitch moment, and almost no yaw moment. This is due to the distribution of the SJAs; had some of the SJAs been located on the vertical and horizontal stabilizers, the SJAs would have been able to make contributions to the yaw and pitch moments, respectively.

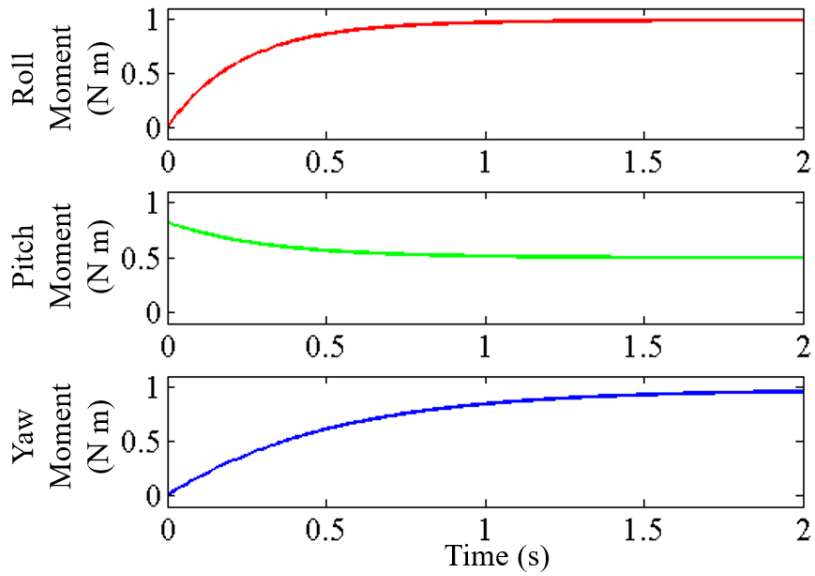

Figure 2.5: Total Moments Using the Proposed Controller

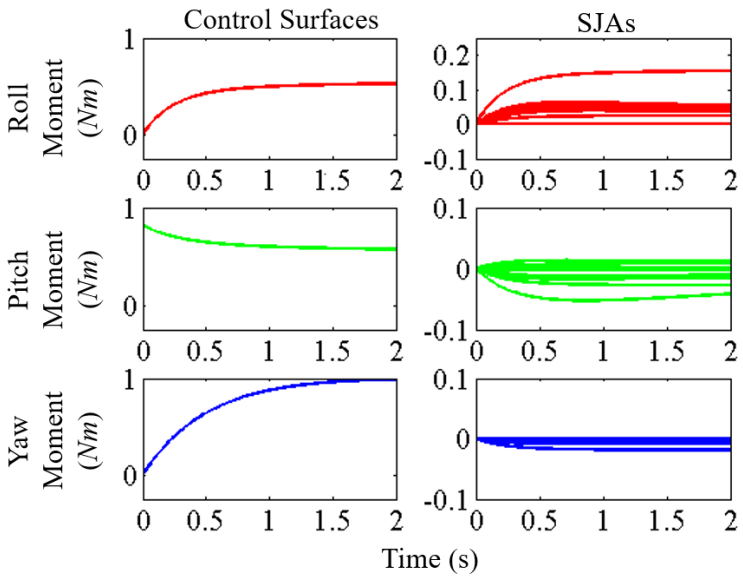

Figure 2.6: Individual Moments From Control Surfaces and SJAs

It is worth mentioning that the CPU time required is very short; 4.8 s are required to solve the NLP offline, and an additional 0.83 s are needed for the simulation when the allocation software is programmed in MATLAB. Thus it is anticipated that the algorithm programmed in C/C++ can definitely meet the real-time need of allocation in consensus.

For comparison, a centralized approach was taken to solve the same problem. The actuations and control voltages for each of the $N + 3$ actuators (N SJAs and 3 control surfaces) were discretized with respect to time with a time step of 0.1 s between each node. The final time was cut from 2 s to 1 s. The same cost function as in Eq. (2.47) is used, but the consensus requirement has been removed; in exchange a trapezoid integration scheme is used to link the control voltage at time node i with the actuation at node $i + 1$. From Fig. 2.7, it can be seen that this method is still able to achieve the desired moment. However, it was seen that this method is slower than our proposed decentralized approach, as it requires 11.5 seconds of CPU time. Further, it is an

open-loop solution, while the proposed controller is in a feedback form.

Although the simulation results show the promising aspects of the proposed consensus algorithm, there are still a number of concerns that must be taken into consideration when conducting hardware experiments. Currently, there is no commercial quality, miniaturized SJA that can be used onboard an SUAS. In the near future, if such SJAs are available, the SUAS must be able to supply sufficient power to the sensors and actuators. The actuators, sensors, power, and wiring required must fit within the limited volume of the SUAS, without going over the limited mass budget of the SUAS. Additionally, as the wiring can shift during flight, this can introduce shifts in the SUAS center of gravity, which must be accounted for when allocating moment actuations.

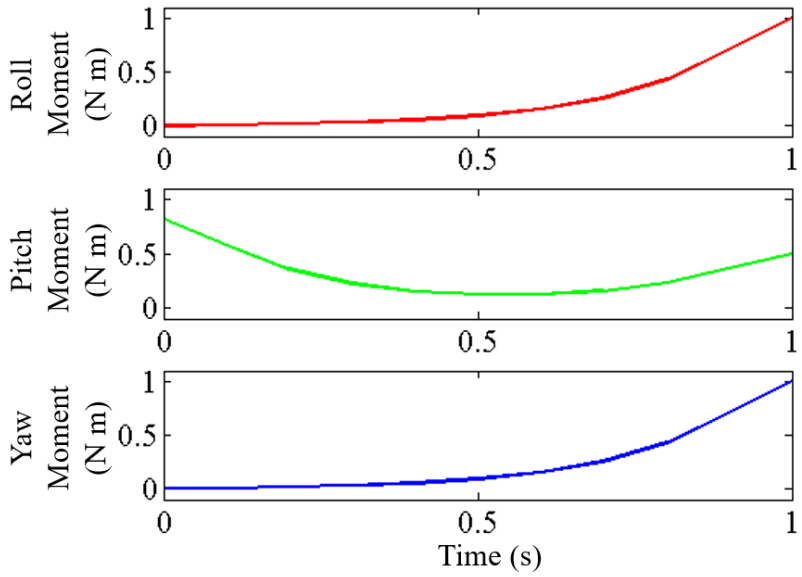

Figure 2.7: Total Moments Generated by a Typical Centralized Approach

CHAPTER 3: CONSENSUS-BASED MOMENT ALLOCATOR FOR DISTRIBUTED NONLINEAR ACTUATORS ONBOARD A CONCEPTUAL SUAS

This work originally appeared in [2]; reprinted by permission of the American Institute of Aeronautics and Astronautics, Inc.

3.1 Chapter Outline

In this study, a consensus-based control allocation scheme is presented for the task of generating a desired moment over an SUAS equipped with control surfaces and augmented by a series of SJAs. This work is an expansion of the work in [1] in that the control surfaces are assumed to follow linear second order dynamics, while the SJAs follow nonlinear second order dynamics. Nonlinearity, consensus, and command tracking are all considered in the proposed consensus allocator, and the closed-loop system is proven to be asymptotically stable. It is shown that the scheme is able to reach the desired moment, and since it is in a feedback sense, real-time local flow information is considered, which can be robust with respect to uncertainties such as gusts and flow separation. The chapter is organized as follows: In Section 3.2, the SUAS platform is discussed, and the dynamics for both SJAs and control surfaces are introduced. In Section 3.3, the control allocation scheme is introduced, and its ability to achieve consensus between agents and to reach a commanded moment asymptotically is proven. In Section 3.4, the moment allocation scheme is simulated, and a short discussion is provided to show the effectiveness of this method handling control constraints.

3.2 SUAS Platform and Invariant Mapping Function Overview

A conceptual, fixed-wing SUAS equipped with a distributed array of pressure and shear sensors, a distributed array of synthetic jet actuators (SJAs), and three control surfaces (aileron, rudder, and elevator) is considered. The whole SUAS surface (i.e. wing-fuselage-tail) area is divided into n areas plus the area covered by the control surfaces. The i^{th} area, $i = 1, ..., n$ contains one SJA, which is able to influence the pressure acting over the area. The measured pressure and shear information from the i^{th} area is expressed as $\vec{q}_i = [p_i, \tau_i^x, \tau_i^y]^T$, where p_i is the measured pressure, and τ_i^x and τ_i^y are the corresponding x- and y-direction shear stresses. For convenience, $\vec{m}_i^a$, the aerodynamic moment acting on the i^{th} area expressed in the body frame, can be expressed using the methods discussed in [27] as

$$\vec{m}_i^a = M_i^a \vec{q}_i \tag{3.1}$$

where M_i^a is a matrix operator defined as

$$M_i^a = s_i \tilde{\vec{r}}_i R_i \tag{3.2}$$

where s_i is the area of section i, $\tilde{\vec{r}}_i$ is the skew symmetric matrix of the position vector of the i^{th} section, expressed in the body frame, and R_i is the rotation matrix from the surface frame to the body frame. In order to isolate the effects of the local pressure on the aerodynamic moment, $\vec{m}_i^a$ can be separated as

$$\vec{m}_i^a = (u_i + p_i)\, \vec{M}_i^{\,p} + \tau_i^x \vec{M}_i^{\,x} + \tau_i^y \vec{M}_i^{\,y} \tag{3.3}$$

where $\vec{M}_i^{\,p}$, $\vec{M}_i^{\,x}$, and $\vec{M}_i^{\,y} \in \mathbb{R}^3$ are the columns of M_i^a, and u_i is the additional pressure in the i^{th} area caused by the SJA. The SJAs are assumed to follow a second order ODE with nonlinear

dynamics as

$$\begin{aligned} \dot{u}_i &= w_i \\ \dot{w}_i &= f\left(u_i, w_i\right) + \varepsilon_i \end{aligned} \tag{3.4}$$

where w_i is the time derivative of u_i, $f\left(u_i, w_i\right) : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a known nonlinear function, and ε_i is the controllable voltage applied to the i^{th} SJA.

The control surface deflections are modeled as following a second order ODE as

$$\begin{aligned} \dot{\vec{\delta}} &= \vec{\gamma} \\ \dot{\vec{\gamma}} &= -b\vec{\delta} + \vec{\varepsilon}^b \end{aligned} \tag{3.5}$$

where b is a positive scalar defined by the control surface mechanics, $\vec{\delta} \in \mathbb{R}^3$ is the vector of aileron, elevator, and rudder deflections and $\vec{\varepsilon}^b$ is the vector of the voltages applied to the aileron, elevator, and rudder, respectively. The total aerodynamic moment due to the control surfaces, $\vec{m}^b \in \mathbb{R}^3$, is given as

$$\vec{m}^b = \vec{m}^{b,0} + M^b\vec{\delta} \tag{3.6}$$

where $\vec{m}^{b,0}$ is the moment with no control surface deflection and $M^b \in \mathbb{R}^{3\times3}$ is the control power matrix. In most aircraft considerations, both $\vec{m}^{b,0}$ and M^b are well characterized via prior test data, and typically M^b is invertible.

Using Eq. 3.4 and Eq. 3.5, a state space representation for the actuators can be built as

$$\begin{bmatrix} \dot{\vec{u}} \\ \dot{\vec{\delta}} \\ \dot{\vec{w}} \\ \dot{\vec{\gamma}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & I_n & 0 \\ 0 & 0 & 0 & I_3 \\ 0 & 0 & 0 & 0 \\ 0 & -bI_3 & 0 & 0 \end{bmatrix} \begin{bmatrix} \vec{u} \\ \vec{\delta} \\ \vec{w} \\ \vec{\gamma} \end{bmatrix} + \begin{bmatrix} \vec{0}_n \\ \vec{0}_3 \\ \vec{f} \\ \vec{0}_3 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ I_n & 0 \\ 0 & I_3 \end{bmatrix} \begin{bmatrix} \vec{\varepsilon} \\ \vec{\varepsilon}^b \end{bmatrix} \tag{3.7}$$

where

$$\vec{u} = [u_1, \ldots, u_n]^T \tag{3.8}$$

$$\vec{w} = [w_1, \ldots, w_n]^T \tag{3.9}$$

and

$$\vec{f} = [f(u_1, w_1), ..., f(u_n, w_n)]^T \tag{3.10}$$

In this paper, it is assumed that the graph used to represent the interconnections between the actuators is undirected. Each SJA is treated as 1 agent, while the combined array of control surfaces is treated as just 1 actuator. As a result, the graph contains $n + 1$ vertices, and its graph Laplacian L is symmetric and positive semi-definite.

3.3 Consensus-Based Moment Allocation

A consensus algorithm is used to determine the moment allocation among the n SJAs and the control surfaces in order to meet a desired final moment, $\vec{m}^d$. The total actuation error, $\vec{e}$, can then be given as

$$\vec{e} = \vec{m}^d - \sum_{i=1}^{n} \vec{m}_i^a - \vec{m}^b \tag{3.11}$$

Using Eq. 3.3 and Eq. 3.6, Eq. 3.11 can be restated as

$$\vec{e} = \vec{m}^e - \sum_{i=1}^{n} u_i \vec{M}_i^p - M^b \vec{\delta} \tag{3.12}$$

It is assumed that, while the SJAs and control surfaces are actuating, the desired moment, as well as the sensed pressure and shear information, is both known and unchanging. For convenience,

these terms have been combined as $\vec{m}^e$.

Let $P_1 \in \mathbb{R}^{(3n+3)\times(n+3)}$, $P_1^u \in \mathbb{R}^{3n\times n}$, $P_2 \in \mathbb{R}^{(3n+3)\times(n+3)}$, and $P_2^u \in \mathbb{R}^{3n\times n}$ be defined as

$$P_1 = \begin{bmatrix} P_1^u & 0 \\ 0 & M^b \end{bmatrix} \tag{3.13}$$

$$P_1^u = \begin{bmatrix} \vec{M}_1^p & \dots & \vec{0}_3 \\ \vdots & \ddots & \vdots \\ \vec{0}_3 & \dots & \vec{M}_n^p \end{bmatrix} \tag{3.14}$$

$$P_2 = \begin{bmatrix} P_2^u & 0 \\ 0 & M^{b,-1,T} \end{bmatrix} \tag{3.15}$$

and

$$P_2^u = \begin{bmatrix} \left\|\vec{M}_1^p\right\|_2^{-2} \vec{M}_1^p & \dots & \vec{0}_3 \\ \vdots & \ddots & \vdots \\ \vec{0}_3 & \dots & \left\|\vec{M}_n^p\right\|_2^{-2} \vec{M}_n^p \end{bmatrix} \tag{3.16}$$

noting that $P_2^T P_1 = I_{n+3}$.

Define a weighting matrix $K \in \mathbb{R}^{(n+3)\times(n+3)}$, inspired by [53] and the formation generation task in [64], as

$$K = \begin{bmatrix} K^u & 0 \\ 0 & k_{n+1} I_3 \end{bmatrix} \tag{3.17}$$

where

$$K^u = diag\left(k_1, \dots, k_n\right) \tag{3.18}$$

where k_i, $i = 1, ..., n+1$ is a positive, offline computed weighting scalar, chosen to influence the magnitude of the final moment generated by agent i. The following Lemma is slightly modified

from Lemmas 1 and 3 in [1].

Lemma 3.1: K is commutative with $P_1^T P_1$ (i.e. $K P_1^T P_1 = P_1^T P_1 K$). Additionally, $K^{-1} P_1^T P_1$ is positive semi-definite.

Proof: Evaluate $K P_1^T P_1$ as

$$K P_1^T P_1 = \begin{bmatrix} k_1 \vec{M}_1^{p,T} \vec{M}_1^p & \dots & 0 & \vec{0}_3^T \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & k_n \vec{M}_n^{p,T} \vec{M}_n^p & \vec{0}_3^T \\ \vec{0}_3 & \dots & \vec{0}_3 & k_{n+1} M^{b,T} M^b \end{bmatrix} \tag{3.19}$$

Similarly, evaluating $P_1^T P_1 K$ will give identical results to Eq. 3.19. Thus, the first half of Lemma 3.1 is proven. Note that K is diagonal, and each of its diagonal entries are constrained to be positive. K^{-1} is similarly diagonal, and each of its diagonal entries are also positive. Square roots can be taken of these diagonal entries, and $K^{-1} P_1^T P_1$ can be rewritten as $\left(P_1 K^{-1/2}\right)^T P_1 K^{-1/2}$. Thus it is clear that $K^{-1} P_1^T P_1$ can be decomposed as a matrix times its transpose, and is positive semi-definite. Thus the second half of Lemma 3.1 is proven.

Definition 3.1: Modified from [1] to directly consider the applied control instead of the resulting aerodynamic moment, a modified form of consensus, called actuator consensus, is said to occur when

$$P_2^T \left(L \otimes I_3\right) P_1 K \begin{bmatrix} \vec{u} \\ \vec{\delta} \end{bmatrix} = \vec{0}_{n+3} \tag{3.20}$$

Let $\vec{u}^f$ and $\vec{\delta}^f$ represent potential steady state actuation values of $\vec{u}$ and $\vec{\delta}$. Let $\vec{h} \in \mathbb{R}^{n+3}$ be defined as

$$\vec{h} = P_2^T \left(L \otimes I_3\right) P_1 K \begin{bmatrix} \vec{u}^f \\ \vec{\delta}^f \end{bmatrix} \tag{3.21}$$

Let $\vec{g}_i \in \mathbb{R}^3, i = 1, ..., n+1$ be defined as the consensus error between agent i and its neighbors. Then, when the agents' actuation values are equal to $\vec{u}^f$ and $\vec{\delta}^f$, $\vec{g} = \begin{bmatrix} \vec{g}_1^T & ... & \vec{g}_{n+1}^T \end{bmatrix}$ can be given in matrix form as

$$\vec{g} = (L \otimes I_3) P_1 K \begin{bmatrix} \vec{u}^f \\ \vec{\delta}^f \end{bmatrix} \tag{3.22}$$

Now, rewrite $\vec{h}$ using the definition of P_2 as

$$\vec{h} = \begin{bmatrix} \left\| \vec{M}_1^p \right\|_2^{-2} \vec{M}_1^{p,T} \vec{g}_1 \\ \vdots \\ \left\| \vec{M}_n^p \right\|_2^{-2} \vec{M}_n^{p,T} \vec{g}_n \\ M^{b,-1} \vec{g}_{n+1} \end{bmatrix} \tag{3.23}$$

Remark 3.1: While the SJAs can only produce moments within the span of their respective $\vec{M}_i^p$ vector, the control surfaces have no such constraints and can freely produce moments about all three axes of motion.

Suppose a vector $\vec{z} \in \mathbb{R}^n$ exists such that it is perpendicular to $\vec{y} \in \mathbb{R}^n$. Then $\vec{y}^T \vec{z} = 0$. From this, if the $n+1$ k_i scalars are chosen such that each $\vec{g}_i$ is either equal to $\vec{0}_3$ or is perpendicular to its corresponding $\vec{M}_i^p$, then $\vec{g}$ is within the null space of P_2, and $\vec{h} = \vec{0}_{n+3}$, and actuator consensus can be guaranteed at $\vec{u}^f$ and $\vec{\delta}^f$. In order to find appropriate values for the k_i terms, as well as to find values for $\vec{u}^f$ and $\vec{\delta}^f$ where not only does actuator consensus occur but the error is also driven to $\vec{0}_3$, nonlinear programming techniques can be used. This problem can be posed as an optimization problem, modified from the nonlinear programming problem in [1] to use a different cost function,

as

$$\begin{aligned} \min \quad & \left\| \begin{bmatrix} \vec{u}^f \\ \vec{\delta}^f \end{bmatrix} \right\|_2^2 \\ \text{subject to} \quad & \vec{M}_i^{p,T} \vec{g}_i = 0, i = 1, ..., n \\ & \vec{g}_{n+1} = \vec{0}_3 \\ & \vec{m}^e - JP_1 \begin{bmatrix} \vec{u}^f \\ \vec{\delta}^f \end{bmatrix} = \vec{0}_3 \\ & \vec{u}^f \in \mathbb{R}^n \\ & \vec{\delta}^f \in \mathbb{R}^3 \\ & k_i > 0, i = 1, ..., n+1 \end{aligned} \tag{3.24}$$

where $J = \vec{1}_{n+1}^T \otimes I_3$.

Remark 3.2: In simulation, the SJA and control surface actuations do not converge to $\vec{u}^f$ and $\vec{\delta}^f$, and their final actuations depend only on the agreement between agents.

The following Lemma is modified from Lemma 4 in [1] to directly consider the applied control.

Lemma 3.2: Assume a solution to Eq. 3.24 exists. Then $\vec{e}$ be rewritten using $\vec{u}^f$ and $\vec{\delta}^f$ as

$$\vec{e} = -JP_1 \left(\begin{bmatrix} \vec{u} \\ \vec{\delta} \end{bmatrix} - \begin{bmatrix} \vec{u}^f \\ \vec{\delta}^f \end{bmatrix} \right) \tag{3.25}$$

Proof: First, rewrite Eq. 3.12 using P_1, J, $\vec{u}^f$, and $\vec{\delta}^f$ as

$$\vec{e} = \vec{m}^e - JP_1 \left(\begin{bmatrix} \vec{u} \\ \vec{\delta} \end{bmatrix} - \begin{bmatrix} \vec{u}^f \\ \vec{\delta}^f \end{bmatrix} + \begin{bmatrix} \vec{u}^f \\ \vec{\delta}^f \end{bmatrix} \right) \tag{3.26}$$

However, if a solution to Eq. 3.24 exists, then $\vec{m}^e - JP_1 \begin{bmatrix} \vec{u}^{f,T} & \vec{\delta}^{f,T} \end{bmatrix}^T = \vec{0}_3$, and Eq. 3.26 can be rewritten as Eq. 3.25. Thus, Lemma 3.2 is proven.

Lemma 3.3: The matrices $P_1^T (L \otimes I_3) P_1$ and $K^{-1}P_1^T J^T J P_1 K^-1$ are both positive semi-definite.

Proof: Recall from the discussion of Graph Theory that it is assumed that for the selected graph, L is symmetric and positive semi-definite. Then $P_1^T (L \otimes I_3) P_1$ is a positive semi-definite matrix multiplied on the left and right by a matrix and its transpose, and is itself positive semi-definite. Also note that $K^{-1}P_1^T J^T J P_1 K^{-1}$ can be written as $(JP_1K^{-1})^T JP_1K^{-1}$, and noting that it is a matrix times its transpose, it is also positive semi-definite. Thus Lemma 3.3 is proven.

Theorem 3.1: Assume a solution exists for Eq 3.24. Driven by the following consensus controller

$$\begin{bmatrix} \vec{\varepsilon} \\ \vec{\varepsilon}^b \end{bmatrix} = -c_1 P_2^T (L \otimes I) P_1 K \begin{bmatrix} \vec{u} \\ \vec{\delta} \end{bmatrix} + c_2 K^{-1} P_2^T J^T \vec{e} + \begin{bmatrix} -\vec{f} - c_3 \vec{w} \\ b\vec{\delta} - c_3 \vec{\gamma} \end{bmatrix} \tag{3.27}$$

where c_1, c_2, and c_3 are user defined control parameters, the closed-loop aerodynamic moment allocation dynamic system is stable. $\vec{e}$ will approach the origin and the $n+1$ actuators will reach actuator consensus.

Proof: Insert Eq. 3.27 into Eq. 3.7, multiply through the control power, and cancel out some of the

common terms to arrive at

$$\begin{bmatrix}\begin{bmatrix}\dot{\vec{u}}\\ \dot{\vec{\delta}}\end{bmatrix}\\ \begin{bmatrix}\dot{\vec{w}}\\ \dot{\vec{\gamma}}\end{bmatrix}\end{bmatrix} = \begin{bmatrix}0 & I_{n+3}\\ 0 & -c_3 I_{n+3}\end{bmatrix}\begin{bmatrix}\begin{bmatrix}\vec{u}\\ \vec{\delta}\end{bmatrix}\\ \begin{bmatrix}\vec{w}\\ \vec{\gamma}\end{bmatrix}\end{bmatrix} - c_1\begin{bmatrix}\vec{0}_{n+3}\\ P_2^T\left(L\otimes I\right)P_1 K\begin{bmatrix}\vec{u}\\ \vec{\delta}\end{bmatrix}\end{bmatrix} + c_2\begin{bmatrix}\vec{0}_{n+3}\\ K^{-1}P_2^T J^T\vec{e}\end{bmatrix} \tag{3.28}$$

Next, use the results from Lemma 3.2 to express $\vec{e}$ in terms of $\vec{u}$ and $\vec{\delta}$ as

$$\begin{bmatrix}\begin{bmatrix}\dot{\vec{u}}\\ \dot{\vec{\delta}}\end{bmatrix}\\ \begin{bmatrix}\dot{\vec{w}}\\ \dot{\vec{\gamma}}\end{bmatrix}\end{bmatrix} = \begin{bmatrix}0 & I_{n+3}\\ 0 & -c_3 I_{n+3}\end{bmatrix}\begin{bmatrix}\begin{bmatrix}\vec{u}\\ \vec{\delta}\end{bmatrix}\\ \begin{bmatrix}\vec{w}\\ \vec{\gamma}\end{bmatrix}\end{bmatrix} - c_1\begin{bmatrix}\vec{0}_{n+3}\\ P_2^T\left(L\otimes I\right)P_1 K\begin{bmatrix}\vec{u}\\ \vec{\delta}\end{bmatrix}\end{bmatrix} - c_2\begin{bmatrix}\vec{0}_{n+3}\\ K^{-1}P_2^T J^T J P_1\left(\begin{bmatrix}\vec{u}\\ \vec{\delta}\end{bmatrix} - \begin{bmatrix}\vec{u}^f\\ \vec{\delta}^f\end{bmatrix}\right)\end{bmatrix} \tag{3.29}$$

For convenience, define $\vec{\mu} \in \mathbb{R}^{n+3}$ and $\vec{\zeta} \in \mathbb{R}^{n+3}$ as

$$\vec{\mu} = K\left(\begin{bmatrix}\vec{u}\\ \vec{\delta}\end{bmatrix} - \begin{bmatrix}\vec{u}^f\\ \vec{\delta}^f\end{bmatrix}\right) \tag{3.30}$$

$$\vec{\zeta} = K\begin{bmatrix}\vec{w}\\ \vec{\gamma}\end{bmatrix} \tag{3.31}$$

As a solution to Eq. 3.24 exists, then $P_2^T \left(L \otimes I_3\right) P_1 K \begin{bmatrix} \vec{u}^{f,T} & \vec{\delta}^{f,T} \end{bmatrix}^T = \vec{0}_{n+3}$. Additionally, note that $\vec{u}^f$ and $\vec{\delta}^f$ are time invariant. Then Eq. 3.29 can be rewritten using $\vec{\mu}$ and $\vec{\zeta}$ as

$$\begin{bmatrix} K^{-1}\dot{\vec{\mu}} \\ K^{-1}\dot{\vec{\zeta}} \end{bmatrix} = \begin{bmatrix} 0 & I_{n+3} \\ 0 & -c_3 I_{n+3} \end{bmatrix} \begin{bmatrix} K^{-1}\vec{\mu} \\ K^{-1}\vec{\zeta} \end{bmatrix} - c_1 \begin{bmatrix} \vec{0}_{n+3} \\ P_2^T \left(L \otimes I\right) P_1 \vec{\mu} \end{bmatrix} - c_2 \begin{bmatrix} \vec{0}_{n+3} \\ K^{-1} P_2^T J^T J P_1 K^{-1} \vec{\mu} \end{bmatrix} \tag{3.32}$$

Next, multiply both the 1^{st} and 2^{nd} order terms by K

$$\begin{bmatrix} \dot{\vec{\mu}} \\ \dot{\vec{\zeta}} \end{bmatrix} = \begin{bmatrix} 0 & I_{n+3} \\ 0 & -c_3 I_{n+3} \end{bmatrix} \begin{bmatrix} \vec{\mu} \\ \vec{\zeta} \end{bmatrix} - c_1 \begin{bmatrix} \vec{0}_{n+3} \\ K P_2^T \left(L \otimes I\right) P_1 \vec{\mu} \end{bmatrix} - c_2 \begin{bmatrix} \vec{0}_{n+3} \\ P_2^T J^T J P_1 K^{-1} \vec{\mu} \end{bmatrix} \tag{3.33}$$

Define a Lyapunov function V as

$$V = \frac{1}{2} \begin{bmatrix} \vec{\mu} \\ \vec{\zeta} \end{bmatrix}^T \begin{bmatrix} H & 0 \\ 0 & K^{-1} P_1^T P_1 \end{bmatrix} \begin{bmatrix} \vec{\mu} \\ \vec{\zeta} \end{bmatrix} \tag{3.34}$$

where $H = c_1 P_1^T \left(L \otimes I\right) P_1 + c_2 K^{-1} P_1^T J^T J P_1 K^{-1}$. From Lemma 3.3 it can be seen that H is positive semi-definite, and from Lemma 3.1 it can be seen that $K^{-1} P_1^T P_1$ is also positive semi-definite, thus V is itself a positive semi-definite function. Using Eq. 3.33, $\dot{V}$ can be expressed as

$$\dot{V} = \begin{bmatrix} \vec{\mu} \\ \vec{\zeta} \end{bmatrix}^T \begin{bmatrix} H & 0 \\ 0 & K^{-1} P_1^T P_1 \end{bmatrix} \left(\begin{bmatrix} 0 & I_{n+3} \\ 0 & -c_3 I_{n+3} \end{bmatrix} \begin{bmatrix} \vec{\mu} \\ \vec{\zeta} \end{bmatrix} - c_1 \begin{bmatrix} \vec{0}_{n+3} \\ K P_2^T \left(L \otimes I\right) P_1 \vec{\mu} \end{bmatrix} - c_2 \begin{bmatrix} \vec{0}_{n+3} \\ P_2^T J^T J P_1 K^{-1} \vec{\mu} \end{bmatrix} \right) \tag{3.35}$$

which, using the commutative property of K found from Lemma 3.1 and the fact that $P_2^T P_1 =$

I_{n+3}, can be expressed as

$$\dot{V} = \begin{bmatrix} \vec{\mu} \\ \vec{\zeta} \end{bmatrix}^T \begin{bmatrix} 0 & H \\ -H & -c_3 K^{-1} P_1^T P_1 \end{bmatrix} \begin{bmatrix} \vec{\mu} \\ \vec{\zeta} \end{bmatrix} \tag{3.36}$$

which reduces to

$$\dot{V} = -c_3 \vec{\zeta}^T K^{-1} P_1^T P_1 \vec{\zeta} \tag{3.37}$$

As $K^{-1}P_1^T P_1$ is positive semi-definite, it is apparent that $\dot{V}$ is negative semi-definite, and V will continue to decrease until $\vec{\zeta}^T K^{-1} P_1^T P_1 \vec{\zeta} = 0$. However, as $\vec{M}_i^a, i = 1, ..., n$ cannot be orthogonal with itself, nor can M^b, $\dot{V}$ will only equal 0 when $\vec{\zeta} = \vec{0}_{n+3}$. From Eq. 3.33, this means that V will reach its lowest value when $\vec{\mu}$ satisfies the following algebraic equation

$$\vec{0}_{n+3} = \left(c_1 K P_2^T \left(L \otimes I \right) P_1 + c_2 P_2^T J^T J P_1 K^{-1} \right) \vec{\mu} \tag{3.38}$$

As a result, the system will stabilize at a final value of $\vec{\mu}$ such that $\vec{\mu}$ is within the nullspace of $c_1 K P_2^T \left(L \otimes I \right) P_1 + c_2 P_2^T J^T J P_1 K^{-1}$. This condition is always met when both actuator consensus is achieved and the error is simultaneously driven to $\vec{0}_3$, though it can occur more generally at other values of $\vec{\mu}$ when the consensus error exactly cancels out the error between the current and desired moment over the SUAS. However, by tuning c_1 and c_2, it can be ensured that the error will approach the origin and actuator consensus will be achieved.

3.4 Simulation and Results

In simulation, a test actuation array of 10 SJAs and 3 control surfaces (aileron, elevator, rudder) is considered. The SJAs are arranged in a 5×2 grid on the wing of an SUAS, which uses a NACA 4412 airfoil. The 11 agents are connected via a Modified Cyclic Pursuit scheme, wherein an agent

i is able to both receive and transfer information to agents $i-1$ and $i+1$, where the end agents (1 and 11) are explicitly able to communicate. It is worth noting that different communication schemes can be used.

The actuation array is given an initial desired moment command of $\vec{m}^e = \begin{bmatrix} 0.1 & 0.1 & 0.1 \end{bmatrix} Nm$, and Matlab's *fmincon* function, a nonlinear programming tool, is used to solve the optimization problem in Eq. 3.24 and find values for a weighting matrix K. Each of the equality constraints in Eq. 3.24 was relaxed to an inequality constraint with a tolerance value of $\sigma = 10^{-3}$. Additionally, in order to more evenly spread the actuation among the SJAs and the control surfaces, a lower bound of 0.1 and an upper bound of 1 were placed on the selection of the $n+1$ k_i terms. The control gains were set at $c_1 = 0.1$, $c_2 = 0.005$, and $c_3 = 70$. On the test computer used, solving this NLP required 27.82 seconds of computation time. When simulated in Simulink, the moment allocation scheme was able to achieve actuator consensus within 0.2 seconds, minimizing the norm of the actuation error to $1.134 \times 10^{-6} Nm$. Additionally, the simulation only requires 0.5711 seconds to simulate 0.5 seconds of actuation at a frequency of $100Hz$ (with about 0.4297 seconds spent initializing Simulink), indicating that, after a K matrix has been found offline, the allocation scheme can be implemented in real time.

In Fig. 3.1, the individual SJA and control surface moments are plotted against time. From this, it can be seen that most of the SJAs are able to make some contribution to the roll moment, some of the SJAs are able to contribute to the pitch moment, and none of the SJAs are able to contribute to the yaw moment. In the roll and pitch direction, just under half of the required actuation is achieved by the control surfaces, with the control surfaces responsible for nearly all of the yaw moment. This can be explained by examining the placement of the SJAs; as all of the SJAs are located close to the center of gravity relative to the z-axis, they cannot produce much moment in the pitch and yaw directions, but the SJAs on the wing tips are well-suited to producing roll moments and the SJAs close to the leading edge is capable of producing small pitch moments. If

SJAs were placed on the vertical and horizontal stabilizers, then those SJAs would be well-suited to producing yaw and improved pitch moments.

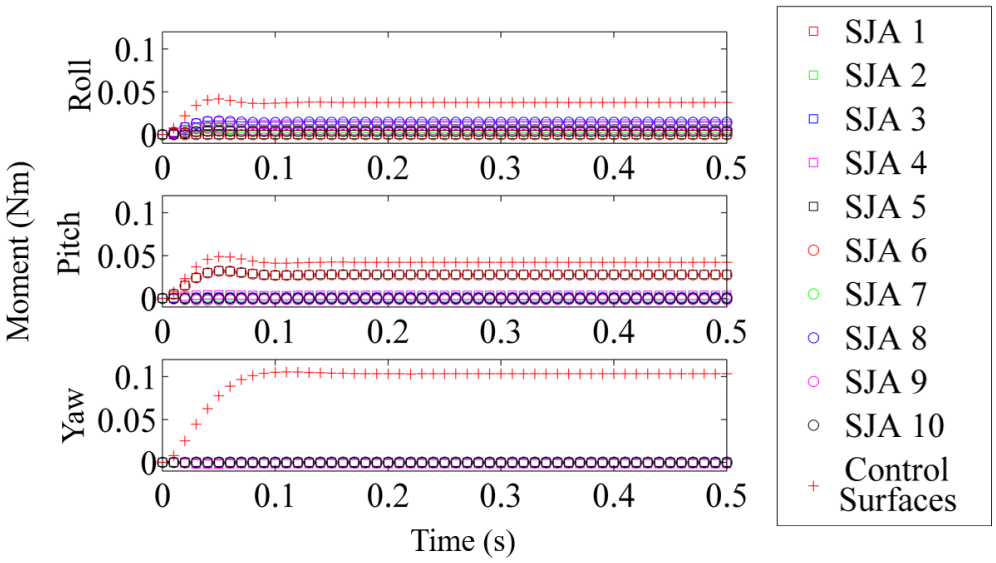

Figure 3.1: Individual Moments using k_i Set 1

In Fig. 3.2, the weighted consensus errors g_i from the actuators are plotted. It can be seen that the weighted moments generally converge in the roll and pitch directions, with a final maximum absolute consensus error of $0.0054Nm$ in the roll direction, $0.0084Nm$ in the pitch direction, and $0.021Nm$ in the yaw direction. This error can be attributed to the constraint placed on the axis of actuation for each of the SJAs; as they can only actuate along a given direction, this creates some residual error that is irreducible. This is particularly noticeable in the yaw direction, where the SJAs are particularly ineffective at producing moment. However, as the leftover error is orthogonal to the respective $\vec{M}_i^p$ of each actuator, it can still be said that actuator consensus is achieved. In Fig. 3.3, the total actuated moment is shown against the desired moment. From this it can be seen that the actuators converge to the desired moment in roughly the same amount of time it takes for consensus to be achieved.

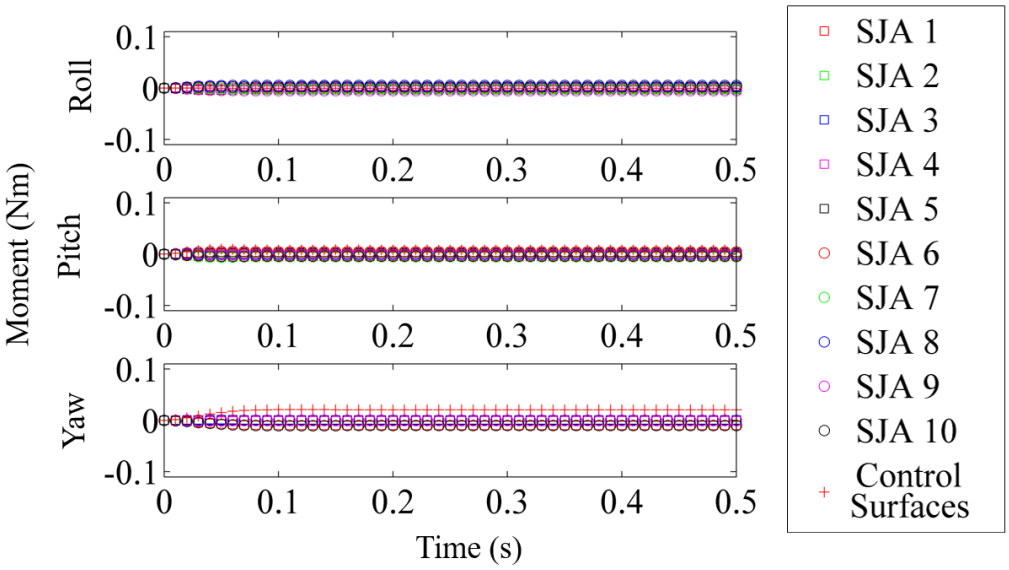

Figure 3.2: Consensus Error between Actuators using k_i Set 1

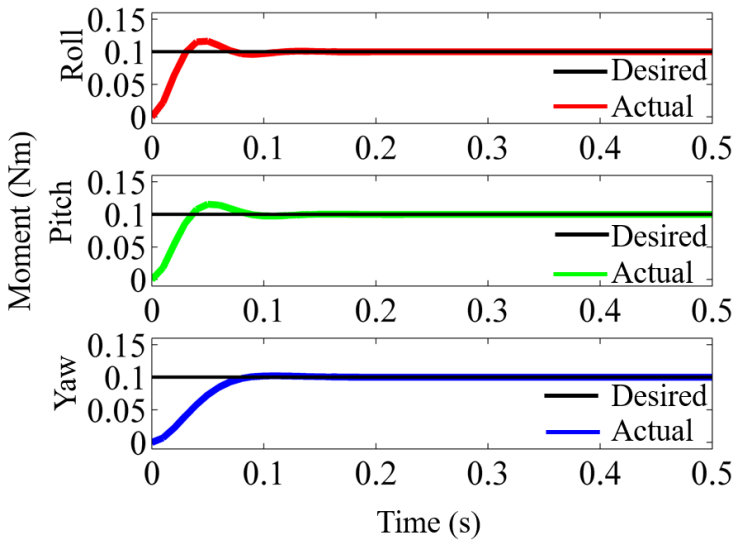

Figure 3.3: Total SUAV Moment using k_i Set 1

To test the effects of the values used for K against the final actuations, three different K matrices were used to achieve the same desired moment. k_i set 1 was generated using the parameters specified above, but in k_i set 2, a lower bound of 0.01 and an upper bound of 10 were placed on the selection of the k_i terms. In k_i set 3, a lower bound of 0.4 and an upper bound of 1.0 were used. The resulting k_i values, as well as the final resulting actuations, are shown in Table 3.1. The individual moments, consensus error, and total SUAV moments are plotted in Figs. 3.4 through 3.9 for k_i sets 2 and 3. In all cases, the same $\vec{m}_e$, c_1, c_2, and c_3 values were used.

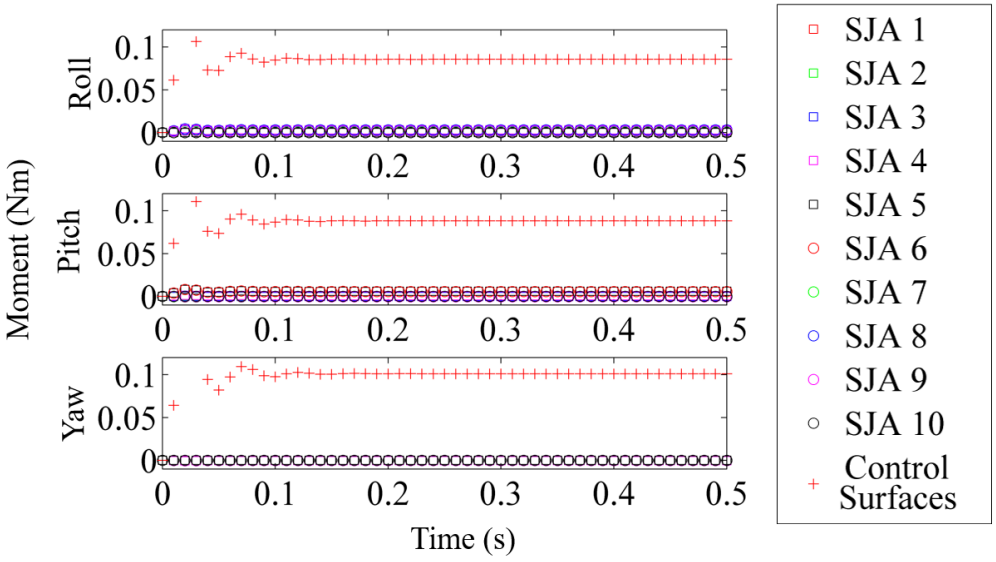

Figure 3.4: Individual Moments using k_i Set 2

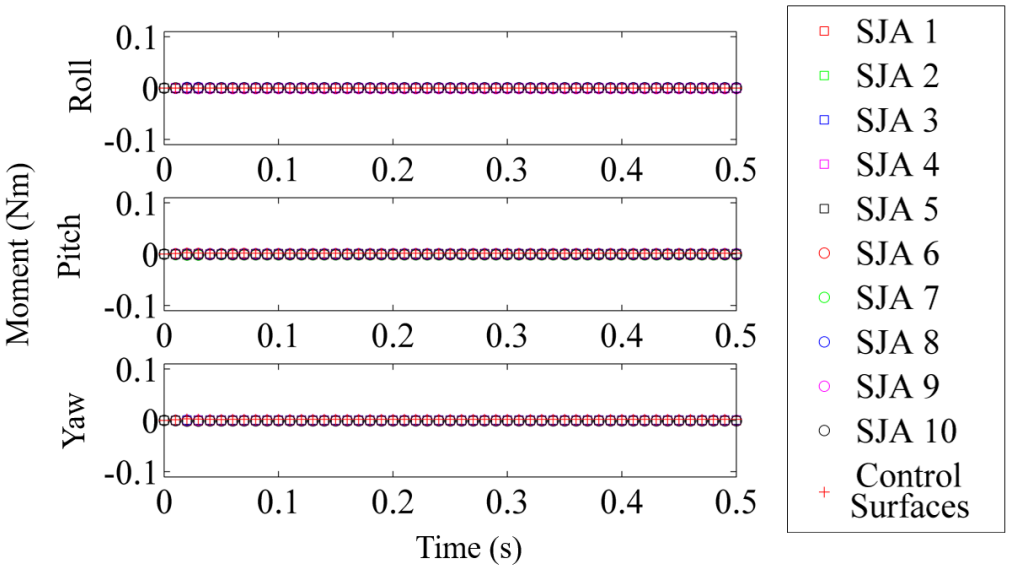

Figure 3.5: Consensus Error between Actuators using k_i Set 2

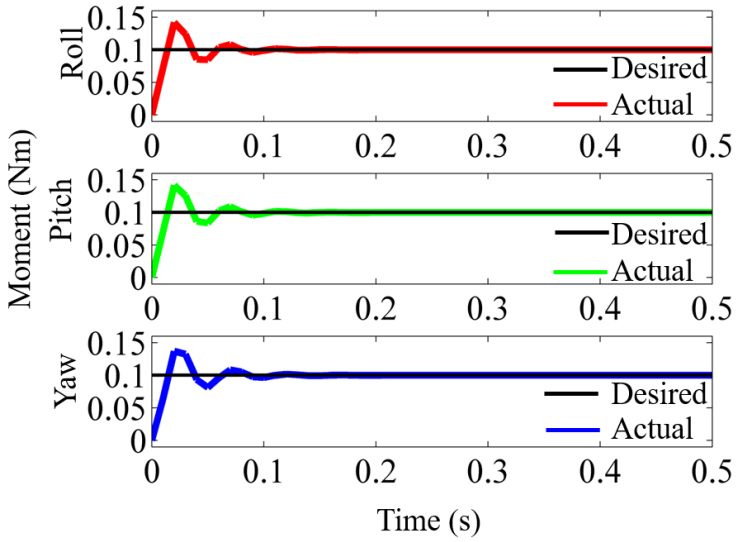

Figure 3.6: Total SUAV Moment using k_i Set 2

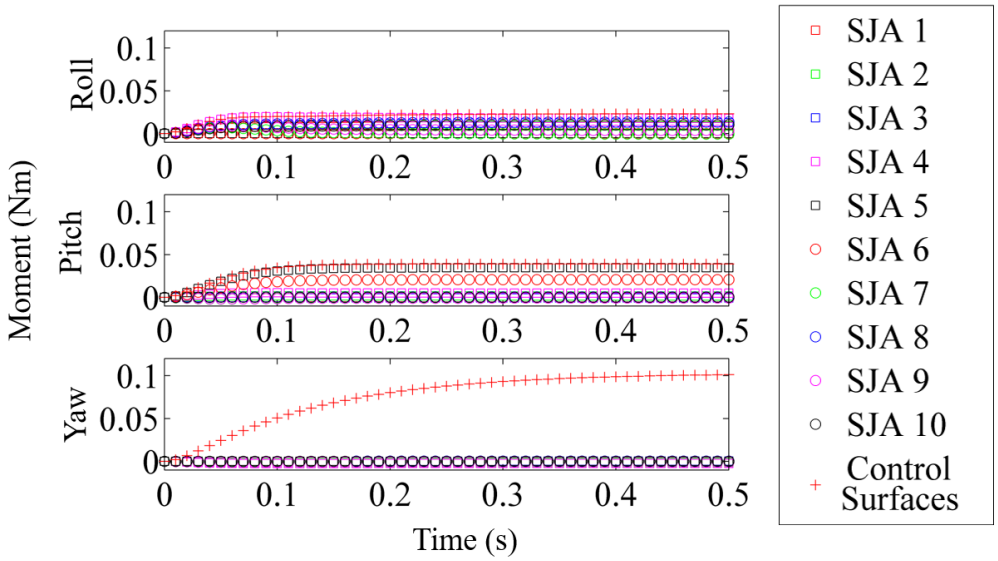

Figure 3.7: Individual Moments using k_i Set 3

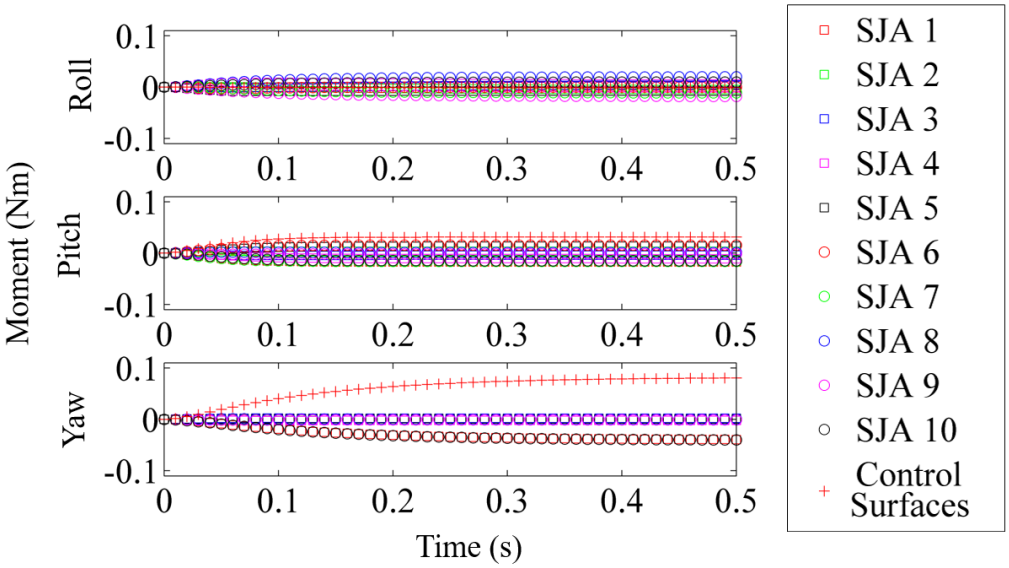

Figure 3.8: Consensus Error between Actuators using k_i Set 3

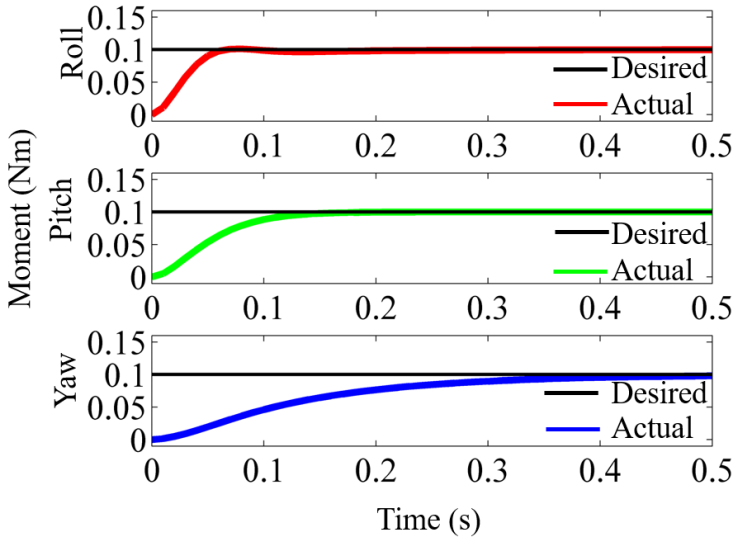

Figure 3.9: Total SUAV Moment using k_i Set 3

In each case, the allocation scheme converged to the desired moment, though when more restrictive bounds for k_i are put in place, the convergence rate, as well as the final error, suffers. By restricting the range of feasible values for k_i, the actuation generated by the SJAs can be increased, more evenly allocating the control, however, this results in poorer consensus. As expected, agents with smaller k_i terms will generate more actuation, while agents with higher k_i terms will generate lower actuations. However, agents towards the middle of the communication topology will generate less moment, for similar reasons as stated above.

Table 3.1: Comparison of Final Actuations with Different k_i sets

	Case 1		**Case 2**		**Case 3**	
	K	Actuation	K	Actuation	K	Actuation
SJA 1	1.0000	-1.8629 Pa	5.8925	-0.0705 Pa	1.0000	-4.6530 Pa
SJA 2	0.4462	-3.4551 Pa	0.4535	-0.9303 Pa	0.7518	-4.5225 Pa
SJA 3	0.3678	-3.7593 Pa	0.3812	-0.8081 Pa	0.7254	-4.7286 Pa
SJA 4	0.2627	-4.6554 Pa	0.2732	-1.2113 Pa	0.4261	-6.6901 Pa
SJA 5	0.1517	107.0147 Pa	0.1516	22.3710 Pa	0.4510	132.9051 Pa
SJA 6	0.1524	-18.9693 Pa	0.1524	-3.9640 Pa	0.7534	-14.1631 Pa
SJA 7	0.2453	1.7582 Pa	0.2453	0.6748 Pa	0.4000	-0.1790 Pa
SJA 8	0.2855	5.3784 Pa	0.2855	1.1907 Pa	0.7828	5.1003 Pa
SJA 9	0.2762	3.0005 Pa	0.2762	0.9880 Pa	0.4000	1.9200 Pa
SJA 10	1.0000	1.9954 Pa	10.0000	0.0442 Pa	1.0000	5.1425 Pa
Control Surfaces	0.1000	$0.0216°$ $-0.0118°$ $0.2749°$	0.0100	$0.0480°$ $-0.0247°$ $0.2741°$	0.4000	$0.0138°$ $-0.0109°$ $0.2677°$

CHAPTER 4: ACHIEVING A UNIFORM SURFACE TEMPERATURE ON LARGE SURFACES VIA COOPERATIVE SPRAY COOLING

This work originally appeared in [3].

4.1 Chapter Outline

The contributions of this study are summarized here. (1) A first-of-its-kind graph theory based approach for allocating fluid to the nozzles in a cooperative spray cooling system is proposed in this paper. Such an approach uses the disagreements between surface sections of a sprayed surface to configure the output of each nozzle in real time, and in a feedback form. (2) To address cases where a constant external heat flux is supplied, an integral term is included. Without this term, the external heat flux would introduce a steady state error, and the surface temperature would maintain a constant offset from the desired temperature. (3) As this controller is based only on the local difference between states, the computational complexity of the controller increases linearly with the number of nozzles, and can be easily implemented in real time. (4) It is mathematically proven that the proposed controllers are stable in the norm sense, and can drive the surface temperature of a surface to uniformity. (5) This work draws inspiration from the consensus-based control approach in [68] for diffusion PDEs with boundary controls, and expands it to consider PDE systems with two spatial dimensions and to consider systems with local interactions inherent to their dynamics.

This chapter is organized as follows. In Section 4.2, terminologies and concepts from graph theory are briefly introduced and reviewed, the heat transfer dynamics are listed, and the cooperative control problem is defined. In Section 4.3, two graph theory based controllers are presented, and their stabilities are proven for PDE systems. In Section 4.4, simulations for these controllers are

used to show the effectiveness of the controllers and their advantages against a conventional on-off spray controller.

4.2 Elementary Concepts and Problem Definition

4.2.1 Graph Theory Review for Cooperative Nozzles

The following concepts are discussed in greater detail in [61]. A system of n cooperative nozzles can be represented by a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, consisting of a vertex set $\mathcal{N}$ representing the n nozzles and an edge set $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ representing the connections among the nozzles. It can be said that an edge (i, j) exists if information can flow from nozzle i to nozzle j. The adjacency matrix $A = [a_{i,j}] \in \mathbb{R}^{n \times n}$ can be defined such that $a_{i,j} > 0$ if $(j, i) \in \mathcal{E}$ exists and $a_{i,j} = 0$ otherwise. For a nozzle i, its in-degree, represented by d_i, represents the total number of nozzles from which it can receive information. Then the in-degree matrix can be defined as $D = diag\,(d_1, \ldots, d_n) \in \mathbb{R}^{n \times n}$. From this, the graph Laplacian can be found as $L = D - A$.

For an unweighted graph, $a_{i,j} > 0 \implies a_{i,j} = 1, \forall i, j$, and for an undirected graph, $(i, j) \in \mathcal{E} \iff (j, i) \in \mathcal{E}, \forall i, j$. As a result, A and L are symmetric. A graph is said to be connected if a path exists between every pair of nozzles. If a graph is undirected and is connected, L will be positive semi-definite [61].

Assumption 1: For each of the proposed control algorithms, it is assumed that a nearest neighbor communication topology is used, as shown in Fig. 4.1. In this topology, nozzles are able to share information about their surface temperature within their sprayable region with their neighbors. This topology is undirected and connected and as a result L will be positive semi-definite [61].

Assumption 2: For each of the proposed control algorithms, it is assumed that the surface tempera-

ture of each section can be measured at a sufficiently high sample rate. In real-world applications, such measurements can be taken using IR cameras, provided that the spray from each nozzle is constrained so as to avoid the creation of a liquid film. Such a method would be particularly convenient for this application, as surface integrals could be easily calculated using the measured temperature information at each pixel location in the thermal image.

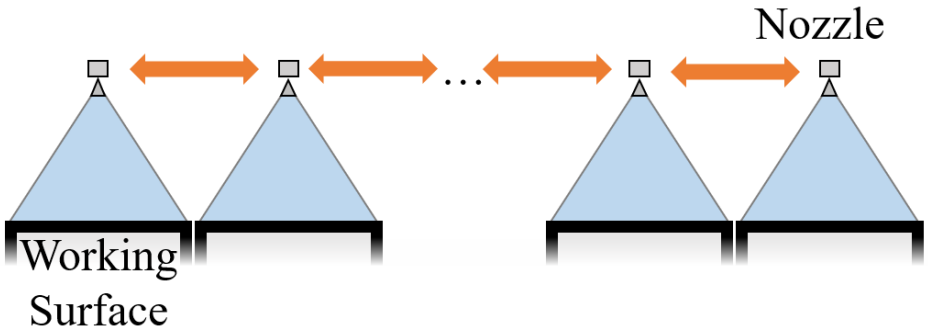

Figure 4.1: Diagram of the nearest neighbor communication topology

4.2.2 Heat Transfer Dynamics

In this study, the large surface to be sprayed is separated into n sections. The temperature within each section, $T_i, i = 1, \ldots, n$, behaves according to the well-known heat equation [69]. Two separate systems are considered.

In the first system, denoted by S_1, heat can flow between neighboring sections and can be removed by spray cooling through the top surface, but heat cannot be dissipated via natural convection through the top, bottom, or side surfaces, nor is there any external heating or internal heat genera-

tion. The following partial differential equation (PDE)

$$\frac{\partial T_i}{\partial t} - \alpha \nabla^2 T_i = 0, i = 1, \ldots, n \tag{4.1}$$

describes this behavior, with the following boundary conditions

$$\left.\frac{\partial T_1}{\partial x}\right|_{x=-w} = 0 \tag{4.2}$$

$$\left.\frac{\partial T_i}{\partial x}\right|_{x=w} = \left.\frac{\partial T_{i+1}}{\partial x}\right|_{x=-w}, i = 1, \ldots, n-1 \tag{4.3}$$

$$\left.\frac{\partial T_n}{\partial x}\right|_{x=w} = 0 \tag{4.4}$$

$$k \left.\frac{\partial T_i}{\partial y}\right|_{y=-h} = 0, i = 1, \ldots, n \tag{4.5}$$

and

$$k \left.\frac{\partial T_i}{\partial y}\right|_{y=h} + U_i = 0, i = 1, \ldots, n. \tag{4.6}$$

while subject to the following initial conditions

$$T_i|_{t=0} = T_{i,0}, i = 1, \ldots, n \tag{4.7}$$

Here, $\alpha > 0$ is the thermal diffusivity of the sprayed surface, shown in Fig. 4.2, w is the half-width of each section within the sprayed surface, h is the half-height of the sprayed surface, $k > 0$ is the thermal conductivity of the sprayed surface, and U_i is the thermal flux out of each section.

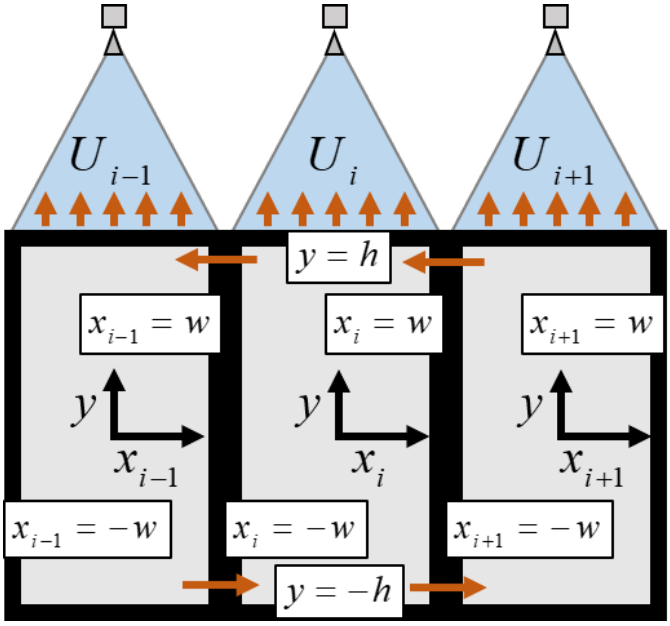

Figure 4.2: Diagram of heated surface

In the second system, denoted by S_2, external heating is introduced to the top surface. The PDE in Eq. (4.1), the boundary conditions in Eqs. (4.2) through (4.5), and the initial conditions in Eq. (4.7) can be used for S_2, replacing Eq. (4.6) with

$$k \left. \frac{\partial T_i}{\partial y} \right|_{y=h} + U_i - Q_i = 0, i = 1, \ldots, n. \tag{4.8}$$

where $Q_i \triangleq Q_i(x), \forall i = 1, \ldots, n$ is the external heating supplied to the top each section. It is assumed that Q_i is constant with time, but can vary spatially.

Remark 4.1: This paper does not assume any particular model for the spray cooling nozzles, but does assume that, given the temperature information of the sprayed surface, a model exists that can be used to determine how fluid must be sprayed to generate a desired heat flux $U_i, i = 1, \ldots, n$. In this way, this paper is left open for use with either existing empirical models describing a one-to-

one relationship between the heat flux and the volumetric flow rate, such as those seen in [70, 71], or future experimentally-verified theoretical models.

In both systems, the n individual T_i, U_i, and Q_i terms can be aggregated as $\vec{T} = [T_1, \ldots, T_n]^T$, $\vec{U} = [U_1, \ldots, U_n]^T$, and $\vec{Q} = [Q_1, \ldots, Q_n]^T$, respectively.

4.2.3 Problem Definition

For S_1, the objective is to drive the temperature of the object to a common temperature, such that

$$\begin{aligned} &\int_{-w}^{w}\int_{-h}^{h} \left[T_i\left(x_i, y_i\right) - T_j\left(x_j, y_j\right)\right]^2 dydx = 0 \\ &\forall i, j = 1, \ldots, n, \forall x_i, x_j \in [-w, w], \forall y_i, y_j \in [-h, h] \end{aligned} \tag{4.9}$$

For S_2, it is required that the system is driven to a desired temperature T_{des}, such that

$$\begin{aligned} &\int_{-w}^{w}\int_{-h}^{h} \left[T_i((x_i, y_i) - T_{des}\right]^2 dydx = 0 \\ &\forall i = 1, \ldots, n, \forall x_i \in [-w, w], \forall y_i \in [-h, h] \end{aligned} \tag{4.10}$$

Additionally, it is desired to minimize the variance in the temperature throughout the object while it is driven to a desired temperature, such that each point on the surface of the object reaches T_{des} at the same time.

Assumption 3: The initial conditions in Eq. (4.7) are assumed to be square integrable and have up to second order derivatives which are themselves square integrable. Thus, the initial conditions are assumed to belong to the Sobolev space $H^2\left(\Omega, \mathbb{R}^n\right)$, where $\Omega = \{x, y \in \mathbb{R} | -w \leq x \leq w, -h \leq y \leq h\}$ [72]. The initial conditions are further assumed to be compatible with the boundary conditions.

4.3 Cooperative Spray Cooling Algorithm

4.3.1 Cooperative Spray Cooling for S_1

Theorem 4.1: For S_1, given the following boundary control algorithm

$$\vec{U} = c_0 L \int_{-w}^{w} \vec{T}\Big|_{y=h} dx \tag{4.11}$$

which can be written in scalar form as

$$U_i = c_0 \sum_{j=1}^{n} a_{i,j} \int_{-w}^{w} (T_i - T_j)|_{y=h}\, dx \tag{4.12}$$

the surface temperature of the n sections in the sprayed surface will asymptotically converge towards a common temperature. Here, c_0 is a user-determined, positive gain.

Proof: Theorem 4.1 is proven using the LaSalle's Invariant Set Theorem. Define a Lyapunov candidate function as

$$V_1 = \frac{1}{2\alpha} \int_{-w}^{w} \int_{-h}^{h} \vec{T}^T \vec{T} dy dx \tag{4.13}$$

From Eq. (4.13), it can be seen that $V_1 \geq 0$ for all values of $\vec{T}$ which are square integrable, and thus for all values of $\vec{T}$ belonging to the Sobolev space $H^0(\Omega, \mathbb{R}^n)$ [72]. Additionally, V_1 only equals zero when $\vec{T}$ is zero at all points, and goes to infinity as $\vec{T}$ goes to infinity.

After taking a time derivative, and taking advantage of the governing equation in Eq. (4.1), $\dot{V}_1$ can be expressed as

$$\dot{V}_1 = \int_{-w}^{w} \int_{-h}^{h} \vec{T}^T \frac{\partial^2 \vec{T}}{\partial x^2} dy dx + \int_{-w}^{w} \int_{-h}^{h} \vec{T}^T \frac{\partial^2 \vec{T}}{\partial y^2} dy dx \tag{4.14}$$

For ease, let F_1 and F_2 be defined as

$$F_1 \triangleq \int_{-w}^{w}\int_{-h}^{h} \vec{T}^T \frac{\partial^2 \vec{T}}{\partial x^2} dy dx \tag{4.15}$$

and

$$F_2 \triangleq \int_{-w}^{w}\int_{-h}^{h} \vec{T}^T \frac{\partial^2 \vec{T}}{\partial y^2} dy dx \tag{4.16}$$

Perform integration by parts on Eq. (4.15),

$$F_1 = -\int_{-h}^{h} \left[\int_{-w}^{w} \frac{\partial \vec{T}}{\partial x}^T \frac{\partial \vec{T}}{\partial x} dx + \left(\vec{T}^T \frac{\partial \vec{T}}{\partial x} \right)\Big|_{x=-w} - \left(\vec{T}^T \frac{\partial \vec{T}}{\partial x} \right)\Big|_{x=w} \right] dy \tag{4.17}$$

Now, after inserting the boundary conditions Eqs. (4.2) through (4.4), and briefly breaking the vector notation used previously, F_1 can be expressed as

$$F_1 = -\int_{-h}^{h} \left\{ \int_{-w}^{w} \frac{\partial \vec{T}}{\partial x}^T \frac{\partial \vec{T}}{\partial x} dx + \sum_{i=1}^{n-1} \left[\frac{\partial T_i}{\partial x}\Big|_{x=w} \left(T_{i+1}|_{x=-w} - T_i|_{x=w} \right) \right] \right\} dy \tag{4.18}$$

Recall that sections i and $i+1, \forall i = 1, \ldots, n-1$ are in direct contact. As a result, it can be said that

$$\frac{\partial T_i}{\partial x}\Big|_{x=w} = \lim_{\Delta x \to 0} \frac{T_{i+1}|_{x=-w+\Delta x} - T_i|_{x=w-\Delta x}}{2\Delta x} \tag{4.19}$$

From this, F_1 can be rewritten as

$$F_1 = -\int_{-h}^{h}\int_{-w}^{w} \frac{\partial \vec{T}}{\partial x}^T \frac{\partial \vec{T}}{\partial x} dx dy - \int_{-h}^{h} \sum_{i=1}^{n-1} \lim_{\Delta x \to 0} \frac{\left(T_{i+1}|_{x=-w+\Delta x} - T_i|_{x=w-\Delta x} \right)^2}{2\Delta x} dy \tag{4.20}$$

As before, perform integration by parts on Eq. (4.16)

$$F_2 = -\int_{-w}^{w}\left[\int_{-h}^{h}\frac{\partial\vec{T}}{\partial y}^T\frac{\partial\vec{T}}{\partial y}dy + \left(\vec{T}^T\frac{\partial\vec{T}}{\partial y}\right)\Big|_{y=-h} - \left(\vec{T}^T\frac{\partial\vec{T}}{\partial y}\right)\Big|_{y=h}\right]dx \tag{4.21}$$

Now, after inserting the boundary conditions Eqs. (4.5) and (4.6) with each U_i as defined in Eq. (4.12), F_2 can be expressed as

$$F_2 = -\int_{-w}^{w}\int_{-h}^{h}\frac{\partial\vec{T}}{\partial y}^T\frac{\partial\vec{T}}{\partial y}dydx - \frac{c_0}{k}\left(\int_{-w}^{w}\vec{T}^T\Big|_{y=h}dx\right)L\left(\int_{-w}^{w}\vec{T}\Big|_{y=h}dx\right) \tag{4.22}$$

Combining Eqs. (4.20) and (4.22), $\dot{V}_1$ can be expressed as

$$\begin{aligned}\dot{V}_1 = &-\int_{-h}^{h}\int_{-w}^{w}\left(\frac{\partial\vec{T}}{\partial x}^T\frac{\partial\vec{T}}{\partial x} + \frac{\partial\vec{T}}{\partial y}^T\frac{\partial\vec{T}}{\partial y}\right)dxdy \\ &-\int_{-h}^{h}\sum_{i=1}^{n-1}\lim_{\Delta x\to 0}\frac{\left(T_{i+1}|_{x=-w+\Delta x} - T_i|_{x=w-\Delta x}\right)^2}{2\Delta x}dy \\ &-\frac{c_0}{k}\left(\int_{-w}^{w}\vec{T}^T\Big|_{y=h}dx\right)L\left(\int_{-w}^{w}\vec{T}\Big|_{y=h}dx\right)\end{aligned} \tag{4.23}$$

From Eq. (4.23), it can be seen that $\dot{V}_1 \leq 0$ for all values of $\vec{T}$ which are square integrable and have first order derivatives which are themselves square integrable, and thus for all values of $\vec{T}$ belonging to the Sobolev space $H^1(\Omega, \mathbb{R}^n)$ [72]. Additionally, $\dot{V}_1 = 0$ only when the temperature is uniform in both the $x-$ and $y-$ directions, and as the initial conditions are in $H^2(\Omega, \mathbb{R}^n)$ according to Assumption 3, the system will asymptotically converge towards the largest invariant set such that $\dot{V}_1$ is equal to zero, following from Theorem 4.3.4 of [72].

From Eq. (4.1), it can be seen that the time derivative of $\vec{T}$ at a given point is equal to zero only when $\nabla^2\vec{T}$ is equal to zero at that point, which can also be said to occur once the temperature is

uniform throughout the surface. Thus, the system will asymptotically converge towards a common temperature, and the system is stable in the space $H^2(\Omega, \mathbb{R}^n)$.

Remark 4.2: The coefficient c_0 determines the speed at which S_1 reaches a uniform temperature. Higher values of c_0 will decrease the amount of time required for the sections to converge on a common temperature. Note that this comes at the cost of increasing the heat flux, with higher values of c_0 imposing potentially unreasonably high heat fluxes in the transient stage.

4.3.2 Cooperative Spray Cooling for S_2

Theorem 4.2: For S_2, given the following control algorithm

$$\begin{aligned}\vec{U} =& c_0 L \int_{-w}^{w} \left(\vec{T}\Big|_{y=h} - T_{des} \right) dx + c_1 \int_{-w}^{w} \left(\vec{T}\Big|_{y=h} - T_{des} \right) dx \\ &+ c_2 \int_0^t \int_{-w}^{w} \left(\vec{T}\Big|_{y=h} - T_{des} \right) dx dt\end{aligned} \tag{4.24}$$

which can be written in scalar form as

$$\begin{aligned}U_i =& c_0 \sum_{j=1}^{n} a_{i,j} \int_{-w}^{w} (T_i - T_j)|_{y=h} \, dx + c_1 \int_{-w}^{w} \left(T_i|_{y=h} - T_{des} \right) dx \\ &+ c_2 \int_0^t \int_{-w}^{w} \left(T_i|_{y=h} - T_{des} \right) dx dt\end{aligned} \tag{4.25}$$

, the surface temperature of the n sections within the sprayed surface will asymptotically converge towards a desired temperature T_{des}. Here c_0, c_1, and c_2 are user-defined, positive gains.

Proof: Theorem 4.2 is similarly proven using the LaSalle's Invariant Set Theorem. Define $\vec{E} = [E_1, \ldots, E_n]^T$ as

$$\vec{E} \triangleq \vec{T} - T_{des} \tag{4.26}$$

Define $\vec{\xi} = [\xi_1, \ldots, \xi_n]^T$ as

$$\vec{\xi} \triangleq \int_0^t \vec{E} dt \tag{4.27}$$

Next, define $\vec{P} = [P_1, \ldots, P_n]^T$, where $P_i \triangleq P_i(x), \forall i = 1, \ldots, n$ is such that

$$c_2 \int_{-w}^{w} \vec{P} dx = \vec{Q} \tag{4.28}$$

Finally, define $\vec{R} = [R_1, \ldots, R_n]^T$ as

$$\vec{R} \triangleq \vec{\xi} - \vec{P} \tag{4.29}$$

With these substitutions, the PDE described by Eqs. (4.1) through (4.5) and Eq. (4.8) can then be rewritten as

$$\frac{\partial^2 R_i}{\partial t^2} = \alpha \left(\frac{\partial^3 R_i}{\partial x^2 \partial t} + \frac{\partial^3 R_i}{\partial y^2 \partial t} \right) \tag{4.30}$$

with the boundary conditions modified as

$$\left. \frac{\partial^2 R_1}{\partial x \partial t} \right|_{x=-w} = 0 \tag{4.31}$$

$$\left. \frac{\partial^2 R_i}{\partial x \partial t} \right|_{x=w} = \left. \frac{\partial^2 R_{i+1}}{\partial x \partial t} \right|_{x=-w}, i = 1, \ldots, n-1 \tag{4.32}$$

$$\left. \frac{\partial^2 R_n}{\partial x \partial t} \right|_{x=w} = 0 \tag{4.33}$$

$$k \left. \frac{\partial^2 R_i}{\partial y \partial t} \right|_{y=-h} = 0 \tag{4.34}$$

and

$$k \left. \frac{\partial^2 R_i}{\partial y \partial t} \right|_{y=h} = -c_0 \sum_{j=1}^{n} a_{i,j} \int_{-w}^{w} \left(\frac{\partial R_i}{\partial t} - \frac{\partial R_j}{\partial t} \right) \Bigg|_{y=h} dx - c_1 \int_{-w}^{w} \left. \frac{\partial R_i}{\partial t} \right|_{y=h} dx - c_2 \int_{-w}^{w} R_i|_{y=h} \, dx \tag{4.35}$$

Define a Lyapunov candidate function as

$$V_2 = \frac{1}{2\alpha} \int_{-w}^{w} \int_{-h}^{h} \frac{\partial \vec{R}}{\partial t}^T \frac{\partial \vec{R}}{\partial t} dy dx + \frac{c_2}{2k} \int_{-w}^{w} \vec{R}^T \Big|_{y=h} dx \int_{-w}^{w} \vec{R} \Big|_{y=h} dx \tag{4.36}$$

From Eq. (4.36), it can be seen that $V_2 \geq 0$ for all values of $\frac{\partial \vec{R}}{\partial t} \in H^0(\Omega, \mathbb{R}^n)$, only equals zero when $\vec{R}$ is constant with time and when the integral of $\vec{R}$ along the top surface equals zero, and goes to infinity as the time derivative of $\vec{R}$ and the integral of $\vec{R}$ along the top surface go to infinity.

After taking a time derivative of Eq. (4.36), $\dot{V}_2$ can be expressed as

$$\dot{V}_2 = \int_{-w}^{w} \int_{-h}^{h} \frac{\partial \vec{R}}{\partial t}^T \frac{\partial^3 \vec{R}}{\partial x^2 \partial t} dy dx + \int_{-w}^{w} \int_{-h}^{h} \frac{\partial \vec{R}}{\partial t}^T \frac{\partial^3 \vec{R}}{\partial y^2 \partial t} dy dx + \frac{c_2}{k} \int_{-w}^{w} \left. \frac{\partial \vec{R}^T}{\partial t} \right|_{y=h} dx \int_{-w}^{w} \vec{R} \Big|_{y=h} dx \tag{4.37}$$

As in Theorem 4.1, let F_3 and F_4 be defined as

$$F_3 \triangleq \int_{-h}^{h} \int_{-w}^{w} \frac{\partial \vec{R}}{\partial t}^T \frac{\partial^3 \vec{R}}{\partial x^2 \partial t} dx dy \tag{4.38}$$

and

$$F_4 \triangleq \int_{-h}^{h} \int_{-w}^{w} \frac{\partial \vec{R}}{\partial t}^T \frac{\partial^3 \vec{R}}{\partial y^2 \partial t} dx dy \tag{4.39}$$

Perform an integration by parts on Eq. (4.38)

$$F_3 = -\int_{-h}^{h} \left[\int_{-w}^{w} \frac{\partial^2 \vec{R}}{\partial x \partial t}^T \frac{\partial^2 \vec{R}}{\partial x \partial t} dx + \left(\frac{\partial \vec{R}}{\partial t}^T \frac{\partial^2 \vec{R}}{\partial x \partial t} \right) \Bigg|_{x=-w} - \left(\frac{\partial \vec{R}}{\partial t}^T \frac{\partial^2 \vec{R}}{\partial x \partial t} \right) \Bigg|_{x=w} \right] dy \tag{4.40}$$

Using the boundary conditions Eqs. (4.31) through (4.33), and again breaking the vector notation, F_3 can be expressed as

$$\begin{aligned} F_3 = &- \int_{-h}^{h} \Bigg\{ \int_{-w}^{w} \frac{\partial^2 \vec{R}}{\partial x \partial t}^T \frac{\partial^2 \vec{R}}{\partial x \partial t} dx \\ &+ \sum_{i=1}^{n-1} \left[\left(\left. \frac{\partial R_{i+1}}{\partial t} \right|_{x=-w} - \left. \frac{\partial R_i}{\partial t} \right|_{x=w} \right) \left. \frac{\partial^2 R_i}{\partial x \partial t} \right|_{x=w} \right] \Bigg\} dy \end{aligned} \tag{4.41}$$

Similar to before, recall that sections i and $i+1, \forall i = 1, \ldots, n-1$ are in direct contact. As a result, it can be said that

$$\left. \frac{\partial^2 R_i}{\partial x \partial t} \right|_{x=w} = \lim_{\Delta x \to 0} \frac{\left. \frac{\partial R_{i+1}}{\partial t} \right|_{x=-w+\Delta x} - \left. \frac{\partial R_i}{\partial t} \right|_{x=w-\Delta x}}{2\Delta x} \tag{4.42}$$

and thus F_3 can be restated as

$$\begin{aligned} F_3 = &- \int_{-h}^{h} \int_{-w}^{w} \frac{\partial^2 \vec{R}}{\partial x \partial t}^T \frac{\partial^2 \vec{R}}{\partial x \partial t} dx dy \\ &- \int_{-h}^{h} \sum_{i=1}^{n-1} \lim_{\Delta x \to 0} \frac{\left(\left. \frac{\partial R_{i+1}}{\partial t} \right|_{x=-w+\Delta x} - \left. \frac{\partial R_i}{\partial t} \right|_{x=w-\Delta x} \right)^2}{2\Delta x} dy \end{aligned} \tag{4.43}$$

Similarly, perform integration by parts on Eq. (4.39)

$$F_4 = - \int_{-w}^{w} \left[\int_{-h}^{h} \frac{\partial^2 \vec{R}}{\partial y \partial t}^T \frac{\partial^2 \vec{R}}{\partial y \partial t} dy + \left(\frac{\partial \vec{R}}{\partial t}^T \frac{\partial^2 \vec{R}}{\partial y \partial t} \right) \Bigg|_{y=-h} - \left(\frac{\partial \vec{R}}{\partial t}^T \frac{\partial^2 \vec{R}}{\partial y \partial t} \right) \Bigg|_{y=h} \right] dx \tag{4.44}$$

Using the boundary conditions Eqs. (4.34) and (4.35), F_4 can be expressed as

$$\begin{aligned} F_4 = &- \int_{-w}^{w}\int_{-h}^{h} \frac{\partial^2 \vec{R}}{\partial y \partial t}^T \frac{\partial^2 \vec{R}}{\partial y \partial t} dy dx \\ &- \int_{-w}^{w} \frac{\partial \vec{R}}{\partial t}^T \Bigg|_{y=h} \left(\frac{c_0}{k} L \int_{-w}^{w} \frac{\partial \vec{R}}{\partial t}\Bigg|_{y=h} dx + \frac{c_1}{k} \int_{-w}^{w} \frac{\partial \vec{R}}{\partial t}\Bigg|_{y=h} dx + \frac{c_2}{k} \int_{-w}^{w} \vec{R}\Big|_{y=h} dx \right) dx \end{aligned} \tag{4.45}$$

,which can be simplified as

$$\begin{aligned} F_4 = &- \int_{-w}^{w}\int_{-h}^{h} \frac{\partial^2 \vec{R}}{\partial y \partial t}^T \frac{\partial^2 \vec{R}}{\partial y \partial t} dy dx - \frac{c_0}{k} \left(\int_{-w}^{w} \frac{\partial \vec{R}}{\partial t}^T \Bigg|_{y=h} dx \right) L \left(\int_{-w}^{w} \frac{\partial \vec{R}}{\partial t}\Bigg|_{y=h} dx \right) \\ &- \frac{c_1}{k} \left(\int_{-w}^{w} \frac{\partial \vec{R}}{\partial t}^T \Bigg|_{y=h} dx \right) \left(\int_{-w}^{w} \frac{\partial \vec{R}}{\partial t}\Bigg|_{y=h} dx \right) - \frac{c_2}{k} \int_{-w}^{w} \frac{\partial \vec{R}}{\partial t}^T \Bigg|_{y=h} dx \int_{-w}^{w} \vec{R}\Big|_{y=h} dx \end{aligned} \tag{4.46}$$

Using Eqs. (4.43) and (4.46), $\dot{V}_2$ can now be expressed as

$$\begin{aligned} \dot{V}_2 = &- \int_{-w}^{w}\int_{-h}^{h} \left(\frac{\partial^2 \vec{R}}{\partial x \partial t}^T \frac{\partial^2 \vec{R}}{\partial x \partial t} + \frac{\partial^2 \vec{R}}{\partial y \partial t}^T \frac{\partial^2 \vec{R}}{\partial y \partial t} \right) dy dx \\ &- \int_{-h}^{h} \sum_{i=1}^{n-1} \lim_{\Delta x \to 0} \frac{\left(\frac{\partial R_{i+1}}{\partial t}\Big|_{x=-w+\Delta x} - \frac{\partial R_i}{\partial t}\Big|_{x=w-\Delta x} \right)^2}{2\Delta x} dy \\ &- \frac{c_0}{k} \left(\int_{-w}^{w} \frac{\partial \vec{R}}{\partial t}^T \Bigg|_{y=h} dx \right) L \left(\int_{-w}^{w} \frac{\partial \vec{R}}{\partial t}\Bigg|_{y=h} dx \right) \\ &- \frac{c_1}{k} \left(\int_{-w}^{w} \frac{\partial \vec{R}}{\partial t}^T \Bigg|_{y=h} dx \right) \left(\int_{-w}^{w} \frac{\partial \vec{R}}{\partial t}\Bigg|_{y=h} dx \right) \end{aligned} \tag{4.47}$$

From Eq. (4.47), it can be seen that $\dot{V}_2 \leq 0$ for all values of $\frac{\partial \vec{R}}{\partial t} \in H^1(\Omega, \mathbb{R}^n)$, and $\dot{V}_2 = 0$ when $\frac{\partial \vec{R}}{\partial t}$ is equal to zero at all points. Additionally, as the initial conditions are in $H^2(\Omega, \mathbb{R}^n)$ according to Assumption 3, the system will asymptotically converge towards the largest invariant set such

that $\dot{V}_2$ is equal to zero, following from Theorem 4.3.4 of [72].

By taking a time derivative of Eq. (4.29), noting that $\vec{P}$ is constant with time, and substituting in Eq. (4.27), it is seen that

$$\frac{\partial \vec{R}}{\partial t} = \frac{\partial \vec{\xi}}{\partial t} = \vec{E} \tag{4.48}$$

Thus, if the time derivative of $\vec{R}$ is equal to zero at all points, then the temperature will be equal to T_{des} at all points. As before, it can be seen from Eq. (4.1) that the time derivative of $\vec{T}$ at a given point is equal to zero only when $\nabla^2 \vec{T}$ is equal to zero at that point, which also be said to occur once the temperature is uniform throughout the surface. Thus, the system will asymptotically converge towards the desired temperature, and the system is stable in the space $H^2(\Omega, \mathbb{R}^n)$.

4.4 Simulation Results and Discussion

Two separate scenarios are considered, with each scenario meant to mimic the conditions observed during a hot strip milling process of steel while using different methods of allocating spray to the individual nozzles. The surface has a thermal conductivity of $51.9 W/m/K$, a density of $7,858 kg/m^3$, a specific heat of $486 J/kg/K$, and a thermal diffusivity of $1.36 \times 10^{-5} m^2/s$. These properties were selected based off the values used in [73], though variations in these properties with temperature were ignored. The initial temperature distribution can be seen in Fig. 4.3. The left-most section has a uniform temperature of $910°C$, with the temperature of each subsequent section increasing by $10°C$. The sprayed surface measures $10cm$ long, and is $6mm$ thick. The surface is assumed to have a unit depth of $1cm$, and the temperature is assumed to not vary along this $z-$axis. The surface is subdivided into 10 sections, with no thermal resistance between adjacent sections. Each section has a half-height of $3mm$ and a half-width of $5mm$. It is assumed that the spray totally covers each section, with no overlap in the spray. The effects of run-off are ignored.

As stated in the Remark 4.1, no particular model is assumed for the spray cooling behavior, and it is assumed that the controllers used can accurately drive the cooling heat flux to the commanded value. In each scenario, the gains are selected in order to avoid unrealistically high heat fluxes at the initial time.

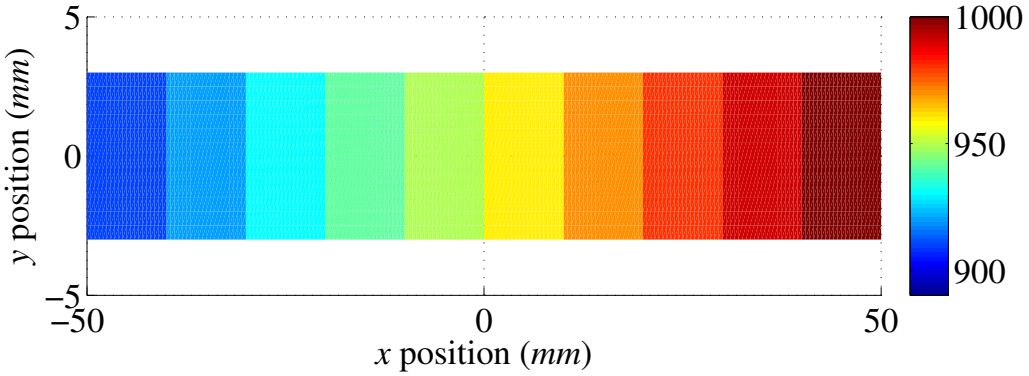

Figure 4.3: Initial temperature distribution ($^{\circ}C$) of surface

A simple Euler scheme is used to integrate the temperature, and a finite differencing method is used to update the Laplacian at every time step. In order to prevent instability when integrating, the time step dt is chosen such that the grid Fourier number, $Fo = \frac{\alpha dt}{dxdy}$ is less than 0.25 [69]. When simulating, dx and dy are set to $0.5mm$, and a dt of $1ms$ is used to satisfy this condition.

4.4.1 Cooperative Spray Cooling for S_1

In Scenario 1, the dynamics in S_1 are used. At any given time, the spray to each nozzle is allocated according to Eq. (4.11), and a value of 6×10^6 used for c_0. The final temperature distribution can be seen in Fig. 4.4, where it can be seen that the controller successfully brings the surface to a uniform temperature of $955^{\circ}C$, the average initial temperature throughout the surface. In Fig. 4.5, the temperature spread, defined as the difference between the maximum and minimum surface

temperature over time, can be seen. As expected, the graph theory approach based allocator drives the sections to exponentially converge to a common temperature, with a 90% convergence rise time of only $8.983s$.

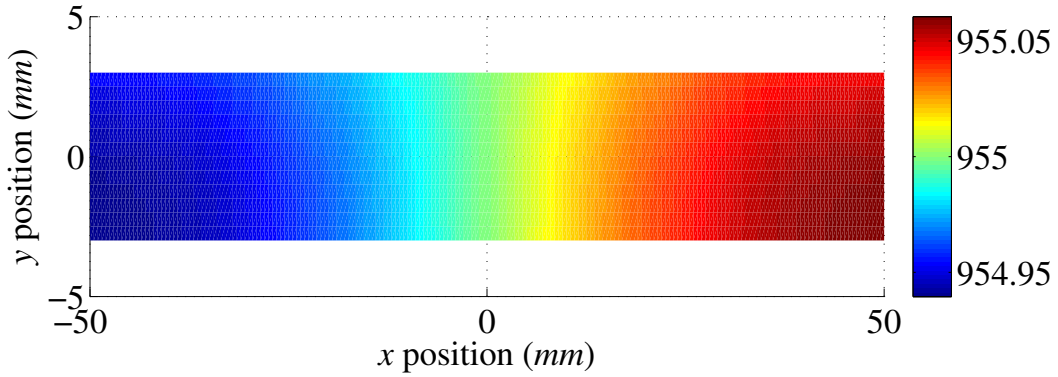

Figure 4.4: Final temperature distribution ($^{\circ}C$) of heated surface for Scenario 1

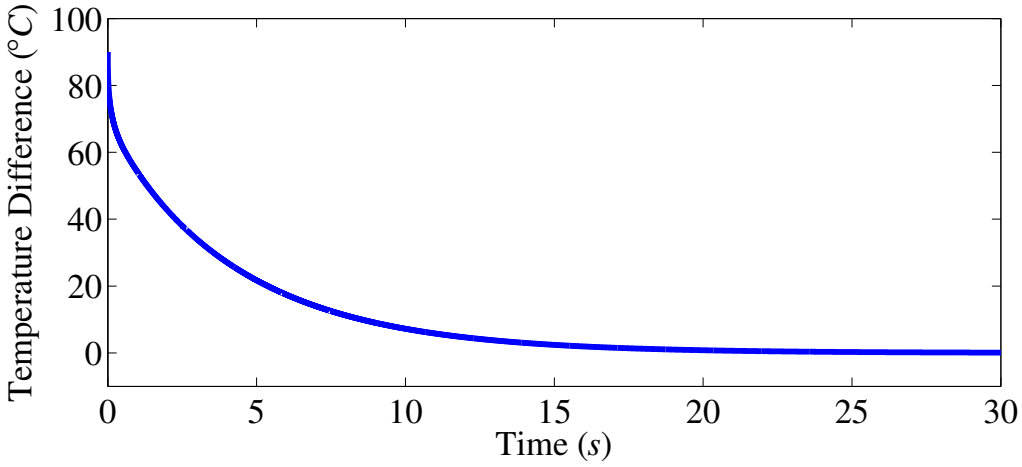

Figure 4.5: Temperature spread across surface over time for Scenario 1

The heat fluxes out of each individual section, as well as the average heat flux over the entire surface, are plotted in Fig. 4.6. The commanded heat fluxes follow the same exponential decay

seen in the temperature difference, with an initial average heat flux of $12W/cm^2$ and a maximum initial heat flux of $60W/cm^2$. The total amount of thermal energy exchanged per unit area, from the beginning of the simulation until 90% convergence is achieved, is $462.9J/cm^2$. Note that the values seen in Fig. 4.6 are the absolute values of the heat flux, as the heat flux is negative in sections whose average surface temperature is below that of the entire surface (i.e. the spray cooling is actually providing heat to these sections). This results in Fig. 4.6 appearing to only contain five unique flux histories; the remaining five are exactly equal in magnitude, but of opposing sign to those visible.

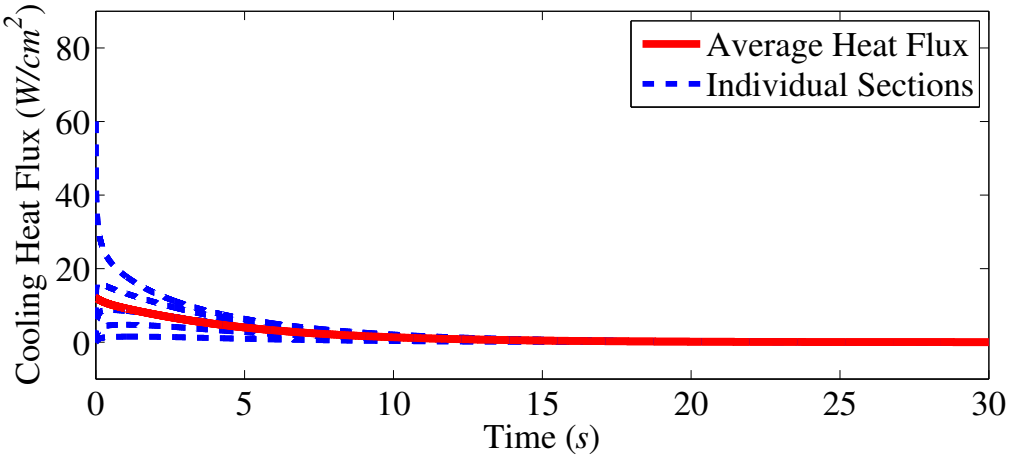

Figure 4.6: Cooling heat fluxes over time for Scenario 1

While this method is hypothetically effective at driving the surface to a uniform temperature, it is not only counter intuitive but highly impractical to use fluidic sprays to provide heat to a surface at such a high temperature. Additionally, this method cannot drive the surface to a desired temperature. Finally, this method has no way of counteracting external heating.

4.4.2 Cooperative Spray Cooling for S_2

Scenario 2 consists of three separate cases, each using the dynamics of S_2. In each case, an external heating of $20W/cm^2$ is applied uniformly to the top surface, and it is desired to drive the surface's temperature to $900°C$. A different control algorithm is used in each case, and the efficacy of each algorithm is compared. In Case 1, the spray to each nozzle is allocated according to Eq. (4.24), with values of 7×10^5, 7×10^5, and 7.5×10^4 chosen for c_0, c_1, and c_2, respectively. In Case 2, a traditional on-off spray allocator is used: while a section's average surface temperature is above $898°C$, a constant heat flux of $30W/cm^2$ is applied, but is dropped to $20W/cm^2$ once this threshold is reached. In Case 3, the gains in Case 1 are increased to 1.5×10^6, 1.2×10^6, and 1.2×10^5 for c_0, c_1, and c_2, respectively, but the heat flux within each section is constrained to $[0, 70]\, W/cm^2$ in order to prevent both negative and unrealistically high heat fluxes throughout the cooling process.

In each case, temperature uniformity is achieved, with the final temperature equal to the desired temperature in Cases 1 and 3, while the final temperature in Case 2 is slightly higher at $901.6°C$ due to the inability of the on-off controller to finely regulate the surface temperature over time. The final temperature distribution for Case 1 can be seen in Fig. 4.7. For the sake of brevity, similar figures for Cases 2 and 3 are omitted, as these three figure would be nearly indistinguishable. Performance metrics for each case can be seen in Table 4.1. It can be seen that both the proposed controller in Case 1 and its modified form in Case 3 are both superior to Case 2, in terms of rise time, overshoot, and total energy flux, while the modified controller in Case 3 is superior to both Cases 1 and 2 in all performance metrics.

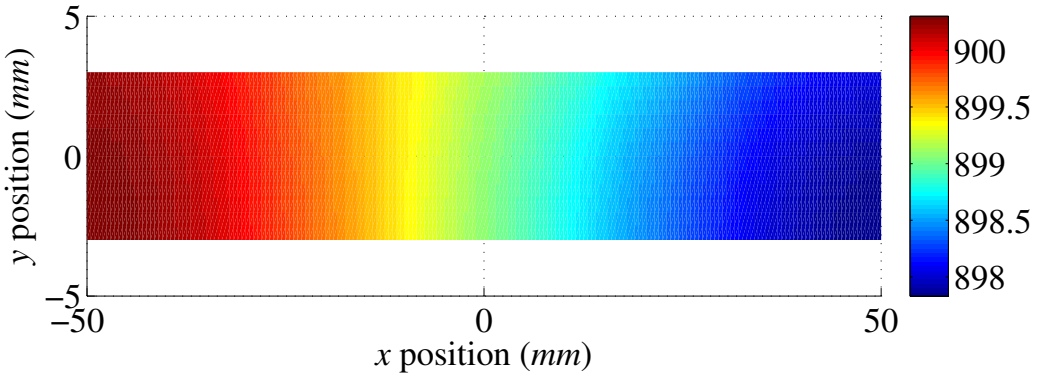

Figure 4.7: Final temperature distribution ($^{\circ}C$) of heated surface for Scenario 2 Case 1

Table 4.1: Simulation Results for S_2

	Case 1	**Case 2**	**Case 3**
Convergence Rise Time (s)	15.68	14.68	3.371
Error Rise Time (s)	12.05	15.67	6.393
Maximum Overshoot ($^{\circ}C$)	3.966	3.292	2.255
Initial Average Heat Flux (W/cm^2)	38.50	30.00	51.80
Initial Maximum Heat Flux (W/cm^2)	77.01	30.00	70.00
Total Energy Flux (kJ/cm^2)	4.352	4.720	2.305

In Fig. 4.8, the difference between the maximum and minimum surface temperatures over time can be seen for each case. Note the abrupt change that appears around $6.5s$ for Case 1. Due to the varying initial conditions of each section, it is not possible for the temperature at every point on the surface to follow an identical exponential decay; if the gains are tuned such that the center experiences a critically damped response, as is the case with the selected gains, the hotter sections will be underdamped while the cooler sections will be overdamped. This causes the temperature spread itself to be underdamped, creating the aforementioned change. Note also that the convergence to a common temperature is slower for Case 1 than in Scenario 1; the c_0 used in Case 1 is lower than that of Scenario 1, and Scenario 1 does not have to contend with any set point

tracking.

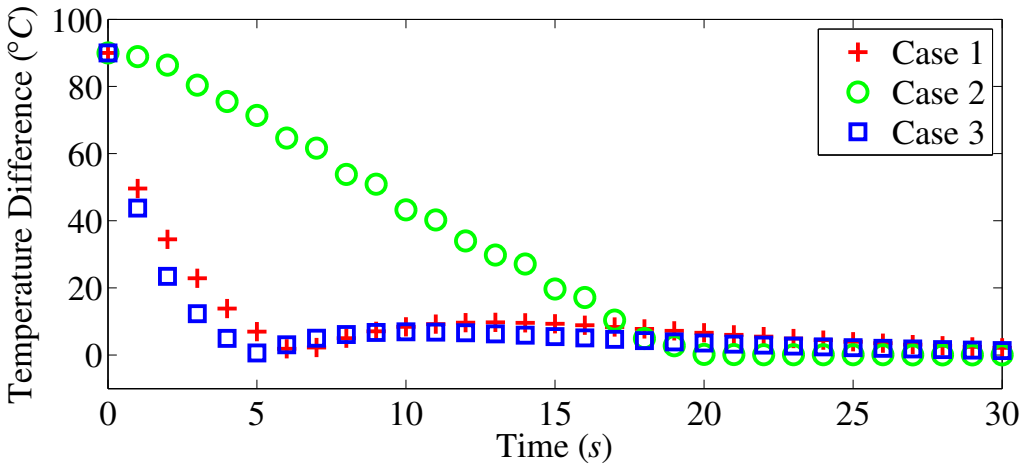

Figure 4.8: Temperature spread across surface over time for Scenario 2

In contrast, several abrupt changes appear in the temperature spread of Case 2. In order to avoid chattering, the temperature of each section is kept within $\pm 2°C$ of the desired temperature. As a result, portions of the surface are momentarily cooled below $898°C$, until conduction can take effect. This behavior produces the discontinuities seen every time a section reaches the cut-off temperature. Despite how different their responses are, Cases 1 and 2 both converge on a uniform temperature in similar times, with Case 2 being slightly faster.

As with Case 1, an abrupt change emerges in Case 3 when the sections which are initially the hottest become cooler than those which are initially the coolest, though this change is smaller in magnitude and arrives earlier. Note that convergence occurs much faster in Case 3 than in the previous cases, due to the higher gain values used.

In Fig. 4.9, the maximum error along the top surface between the current and desired temperature over time can be seen for each section. Whereas Case 1 only saw one sharp change in its tem-

perature spread, it sees two in its error response. The first change appears at the same time, and for the same reason, as the temperature spread sharp change. The second change, which emerges around $18.3s$, has a similar cause. As the temperature response is underdamped in the hotter sections, their temperatures will overshoot the desired temperature, and the temperature error in these sections will become greater than the error in the overdamped sections.

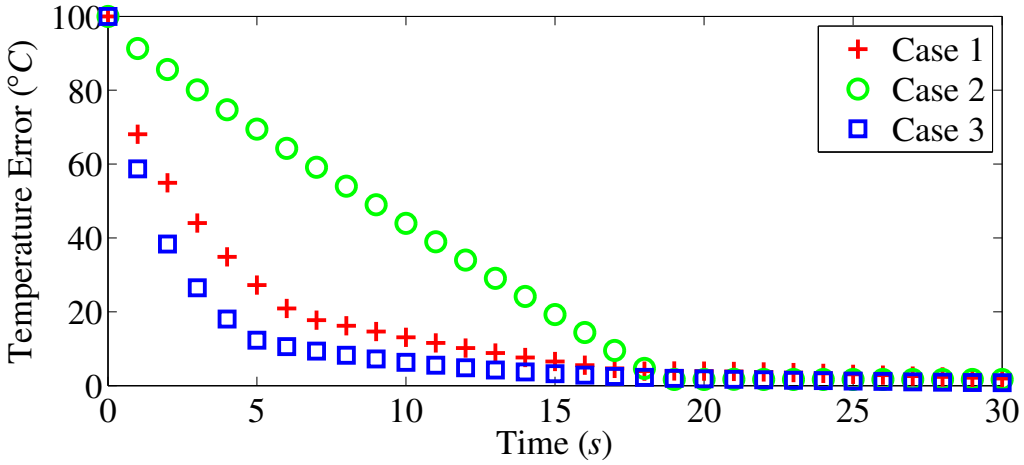

Figure 4.9: Temperature error across surface over time for Scenario 2

The temperature error exhibits a very different behavior in Case 2. Due to the constant heat flux applied to each section when it is above the cut-off temperature, the hottest point remains fixed throughout much of the simulation. This response highlights the constant decrease in temperature seen at this point, until $18.6s$ when it too approaches the cutoff temperature. Once again, despite how different their responses are, Cases 1 and 2 both reach the desired temperature in similar amounts of time, though this time Case 1's response is faster, and by a greater margin.

Unlike in Case 1, only one abrupt change emerges in the error response of Case 3. This change appears for the same reason as in the temperature spread. However, while the underdamped sections still overshoot the desired temperature, their overshoot is not as great as in Case 1, and the second

change does not emerge. The decrease in temperature is considerably faster than in Cases 1 and 2, again thanks to the higher gain values used. The maximum overshoot is also over $1°C$ lower than the overshoot in Cases 1 and 2.

The heat flux out of each individual section for Case 1, as well as the average heat flux over the entire surface, can be seen in Fig. 4.10. The commanded heat fluxes exhibit asymptotic convergence towards $20W/cm^2$, the heat flux required to cancel out the external heating. From here, the extent of the overdamped and underdamped responses exhibited in the hottest and coolest sections can be seen; in the sections which overshoot the desired temperature, the heat flux is only reduced slightly from the steady-state value, and no negative heat flux is experienced, whereas the heat flux is only slightly elevated in the overdamped sections. Note that the total amount of thermal energy exchanged per unit area, from the beginning of the simulation until convergence has occurred, is much larger than in Scenario 1. This increase is due to three factors: the increased rise time, the lower final temperature, and the presence of external heating.

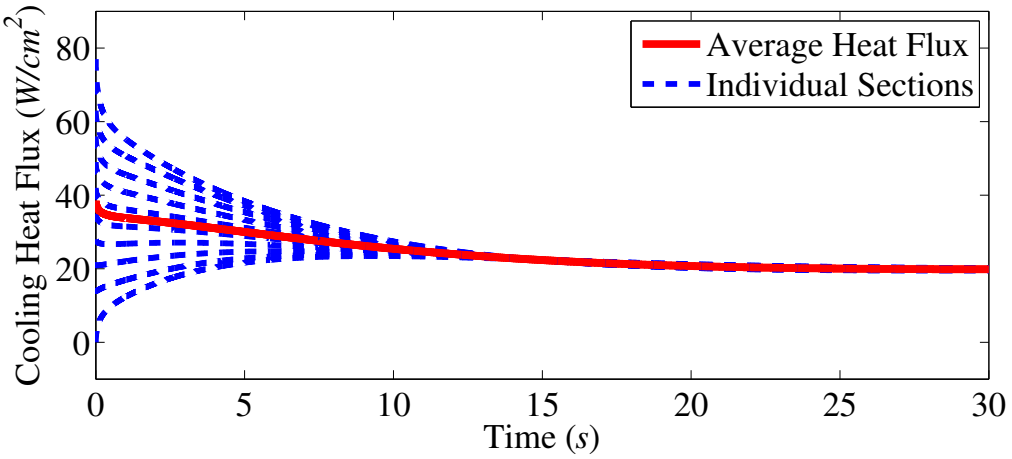

Figure 4.10: Cooling heat fluxes over time for Scenario 2 Case 1

The heat fluxes for Case 2 are plotted in Fig. 4.11. The on-off behavior of the spray allocator is

readily observed, as the individual heat fluxes promptly fall once the cut-off threshold has been reached, and the average heat flux exhibits a staggered decay towards the $20W/cm^2$ required to cancel out the external heating. The total amount of thermal energy exchanged per unit area is slightly larger than in the previous scenario, despite the 90% error rise time of the third scenario being equal to the 90% convergence rise time of the Scenario 2. This is expected, as the commanded overshoot means more heat will be extracted than is necessary, and on-off style controllers are not necessarily optimal from an energy standpoint.

Finally, the heat fluxes for Case 3 are plotted in Fig. 4.12. The upper limit imposed on the heat flux is readily apparent, as the hottest section experiences the maximum allowed heat flux for just over $1.5s$. In order to address the build up of liquid films, which reduce the cooling efficiency of sprays, the gains were selected to avoid saturating the spray in any section for too long, and $1.5s$ was considered an acceptable period of time to saturate the spray allocator. Again, the commanded heat fluxes asymptotically converge on the $20W/cm^2$ required to negate the external heating. The extent of the overdamped and underdamped responses is also reduced, with less deviation from the final heat flux. Note that while this allocator requires a 34.5% greater initial heat flux than Case 1, less energy must be extracted overall before the surface uniformly reaches the desired temperature, as the 90% error rise time seen here is almost 40% of the 90% convergence rise time seen in the Scenario 2.

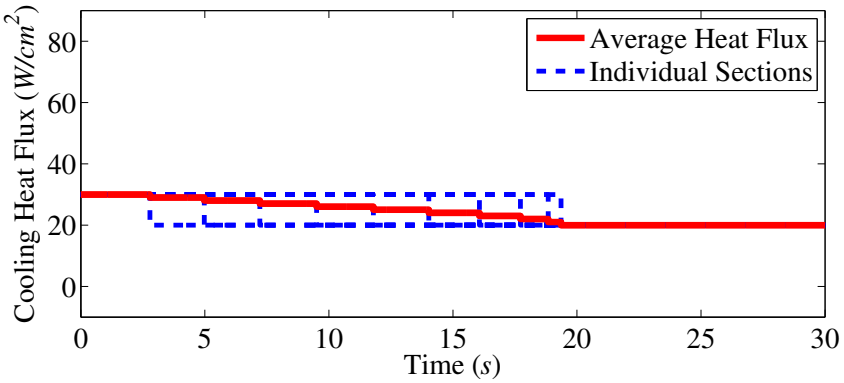

Figure 4.11: Cooling heat fluxes over time for Scenario 2 Case 2

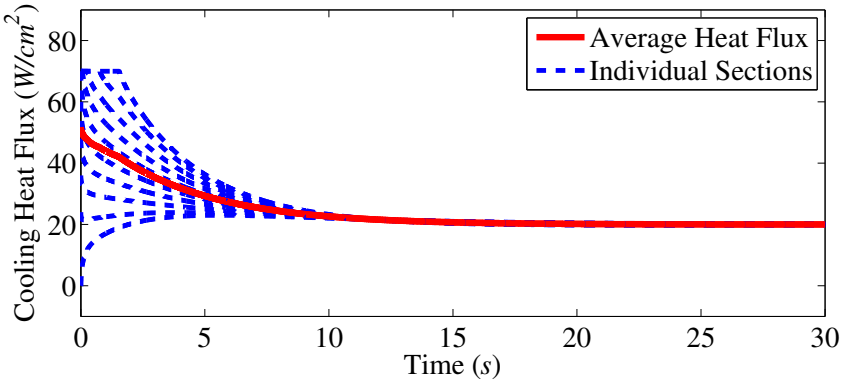

Figure 4.12: Cooling heat fluxes over time for Scenario 2 Case 3

CHAPTER 5: CONCLUSION

5.1 Concluding Remarks

In Chapter 2, a state-space model is formulated for a new SUAS control scenario. In this scenario, surface flow conditions are sensed by an array of micro-scale flow sensors, and SJAs are used to perturb the local flows around them. SJAs and control surfaces work together to generate desired aerodynamic moment for the SUAS three-axis attitude control. The state-space model describes the aerodynamic allocation dynamics among the SJAs and control surfaces. A decentralized allocator, considering constraints and with limited feedback information, is designed for this model such that the SJAs and control surfaces can achieve consensus in contributing to a desired total 3D aerodynamic moment. A Nearest Neighbor communication topology is presented for such a decentralized system. The Lyapunov method is used to prove that the closed-loop allocation dynamics is asymptotically stable. Simulation results are used to show the effectiveness of the proposed method. Note that only undirected communication topologies are considered within this study.

In Chapter 3, a decentralized consensus based control allocation scheme is presented for an SUAS equipped with synthetic jets actuators and control surfaces. Consensus is achieved among heterogeneous actuator models governing aileron, rudder, elevator, and jets, considering control limitations. These actuators collectively decide how to distribute aerodynamic moments among them and achieve a desired total moment. The allocation scheme is in a feedback form, implemented in real time. Additionally, the proposed decentralized computing approach, instead of centralized optimization approaches dominating existing control allocation schemes, has the advantage in a better center of gravity management in SUAS design. The closed-loop system is proven to be asymptotically stable, and simulation results show the effectiveness of the proposed allocator.

In Chapter 4, new approaches for allocating fluid to an array of nozzles for spray cooling heated surfaces are considered. The first approach is intended to bring the heated surface to a uniform temperature, while the second approach introduces tracking terms to drive the heated surface to a desired temperature while rejecting external heating and minimizing the temperature variation throughout the surface. These approaches were proven to be stable in the $H^1(\Omega, \mathbb{R}^n)$ space, and were later demonstrated to be effective via simulations. The tracking approach was then compared against a traditional on-off controller, and was found to be of comparable speed while requiring less heat to be rejected to achieve the desired goal. Finally, a version of the tracking controller with saturation limits was presented, and was found to be much more effective than previous approaches. In applications wherein the surface to be cooled suffers from hot spots, a graph theory approach can be very effective at both driving the system to a desired temperature and eliminating these temperature hot spots.

5.2 Future Work

The consensus-based moment allocation schemes presented in Chapters 2 and 3 could be expanded in several different directions. Currently these allocation schemes only consider undirected topologies. If they were expanded to consider directed topologies, the underlying communication complexity could be reduced, and integration onto physical hardware for either wind tunnel or flight testing could be made simpler. Currently the SJAs in the hypothetical aircraft are only located along the wing, making them ill-suited for producing pitch and yaw moments. If the SJAs could be located on the horizontal and vertical stabilizers, they would be able to make greater contributions to the overall moment along these axes. Finally, the actuator dynamics models proposed for the SJAs are very simple, and do not consider the proper Navier-Stokes equations underlying the output from physical SJAs. More rigorous models could be used, or proper robustness analysis could

be used to consider the uncertainty in the generated output.

The consensus-based spray allocation scheme presented in Chapter 4 currently only considers 2D heat conduction of an object with uniform conductivity. Most logically, this could be expanded to consider 3D heat conduction of a layered object, or with a temperature varying thermal conductivity. External heating is currently only applied to the same surface which receives spray cooling; if the heating were applied to the bottom surface, this would produce a steady-state temperature gradient throughout the object, and the nature of the spray allocator would need to be altered to reflect a new goal for this new circumstance. Finally, a more rigorous heat transfer model could be applied to the system in order to generate the necessary fluidic flux rate to meet the commanded heat flux.

APPENDIX : COPYRIGHT LETTERS

August Mark

From: Heather Brennan <HeatherB@aiaa.org>
Sent: Wednesday, February 13, 2019 5:24 PM
To: August Mark
Subject: RE: Copyright Issue for Paper Appearing in Dissertation

Dear August,

Thank you for your inquiry. Permission is granted by AIAA for you to reprint your AIAA conference paper 2019-1160 in your dissertation, as described below. Please acknowledge within the main text or a footnote that this section/chapter of your dissertation is reprinted with permission (e.g., "From [paper title and authors]; reprinted by permission of the American Institute of Aeronautics and Astronautics, Inc."). Note that the original source should be cited in full in the reference list.

For more information about author rights to their own work, see our Rights and Permissions page:
https://www.aiaa.org/rightsandpermissions/

If you have any further questions, please let me know.

Sincerely,

Heather A. Brennan
Director, Publications

American Institute of Aeronautics and Astronautics www.aiaa.org
12700 Sunrise Valley Drive, Suite 200
Reston, VA 20191-5807
800-639-AIAA (2422)
heatherb@aiaa.org 703.264.7568 (direct)

From: August Mark <augjmark@knights.ucf.edu>
Sent: Wednesday, February 13, 2019 4:24 PM
To: Heather Brennan <HeatherB@aiaa.org>
Subject: Copyright Issue for Paper Appearing in Dissertation

Heather,

I am completing a doctoral dissertation at the University of Central Florida entitled “Decentralized Consensus-Based Control Allocation for Some Dynamical Systems.” I would like your permission to reprint in my dissertation the following:

August J. Mark, Yunjun Xu, and Benjamin Dickinson. "Consensus-based Moment Allocator for Distributed Nonlinear Actuators onboard a Conceptual SUAS", AIAA Scitech 2019 Forum, AIAA SciTech Forum, (AIAA 2019-1160)

The entire document, including all text, figures, equations, tables, etc. are to be reproduced. The requested permission extends to any future revisions and editions of my dissertation, including non-exclusive world rights in all languages. These rights will in no way restrict republication of the material in any other form by you or by others authorized by you. Your response will also confirm that you own or your company owns the copyright to the above-described material.

If these arrangements meet with your approval, please let me know how best to attribute copyright of this work within the dissertation.

Thank you,
August Mark

LIST OF REFERENCES

[1] Mark, A., Xu, Y., and Dickinson, B. T., 2018. "Consensus-based decentralized aerodynamic moment allocation among synthetic jets and control surfaces". *IEEE Transactions on Control Systems Technology*, pp. 1–9.

[2] Mark, A., Xu, Y., and Dickinson, B. T., 2019. "Consensus-based moment allocator for distributed nonlinear actuators onboard a conceptual suas". In AIAA Scitech 2019 Forum.

[3] Mark, A., Xu, Y., and Dickinson, B. T. Achieving a uniform surface temperature on large surfaces via cooperative spray cooling. Submitted to IEEE Transactions on Control Systems Technology.

[4] Necker, R., 1985. "Observations on the function of a slowly-adapting mechanoreceptor associated with filoplumes in the feathered skin of pigeons". *Journal of Comparative Physiology A,* **156**(3), May, pp. 391–394.

[5] Brown, R. E., and Fedde, M. R., 1993. "Airflow sensors in the avian wing". pp. 13–30.

[6] Hörster, W., 1990. "Vibrational sensitivity of the wing of the pigeon (columba livia) — a study using heart rate conditioning". *Journal of Comparative Physiology A,* **167**(4), Sep, pp. 545–549.

[7] Sterbing-D'Angelo, S. J., and Moss, C. F. *Flow Sensing in Air and Water: Behavioural, Neural, and Engineering Principles of Operation*. ch. Air Flow Sensing in Bats, pp. 197–213.

[8] Camhi, J. M., 1969. "Locust wind receptors, I. transducer mechanics and sensory response". *Journal of Experimental Biology,* **50**(2), pp. 335–348.

[9] Shimozawa, T., Kumagai, T., and Baba, Y., 1998. "Structural scaling and functional design of the cercal wind-receptor hairs of cricket". *Journal of Comparative Physiology A,* **183**(2), July, pp. 171–186.

[10] Yu, L., Gutierrez, C. A., and Meng, E., 2016. "An electrochemical microbubble-based mems pressure sensor". *Journal of Microelectromechanical Systems,* **25**(1), February, pp. 144–152.

[11] Maschmann, M. R., Ehlert, G. J., Dickinson, B. T., Phillips, D. M., Ray, C. W., Reich, G. W., and Baur, J. W., 2014. "Bioinspired carbon nanotube fuzzy fiber hair sensor for air-flow detection". *Advanced Materials,* **26**(20), pp. 3230–3234.

[12] Magar, K. T., Reich, G. W., Kondash, C., S., K., Pankonien, A. M., Baur, J. W., and Smyers, B., 2016. "Aerodynamic parameters from distributed heterogeneous CNT hair sensors with a feedforward neural network". *Bioinspiration & Biomimetics,* **11**(6), November, p. 066006.

[13] Callegari, S., Zagnoni, M., Golfarelli, A., Tartagni, M., Talamelli, A., Proli, P., and Rossetti, A., 2006. "Experiments on aircraft flight parameter detection by on-skin sensors". *Sensors and Actuators A: Physical,* **130-131**, pp. 155 – 165.

[14] Que, R., and Zhu, R., 2012. "Aircraft aerodynamic parameter detection using micro hot-film flow sensor array and bp neural network identification". *Sensors,* **12**(8), pp. 10920–10929.

[15] Phillips, D. M., Ray, C. W., Hagen, B. J., Su, W., Baur, J. W., and Reich, G. W., 2015. "Detection of flow separation and stagnation points using artificial hair sensors". *Smart Materials and Structures,* **24**(11), "October", p. 115026.

[16] Guerreiro, N. M., and Hubbard, J. E., 2008. "Pressure port placement for lift distribution measurement on a model aircraft with optimized trailing-edge flaps". In 26th AIAA Applied Aerodynamics Conference, pp. AIAA Paper 2008–7518.

[17] Francioso, L., de Pascali, C., Pescini, E., de Giorgi, M. G., and Siciliano, P., 2016. "Modeling, fabrication and plasma actuator coupling of flexible pressure sensors for flow separation detection and control in aeronautical applications". *Journal of Physics D: Applied Physics,* **49**(23), "May", p. 235201.

[18] Thompson, K., Xu, Y., and Dickinson, B. T., 2017. "Aerodynamic moment model calibration from distributed pressure arrays". *Journal of Aircraft,* **54**(2), March, pp. 716–723.

[19] Amitay, M., Pitt, D., Kibens, V., Parekh, D., and Glezer, A., 2000. "Control of internal flow separation using synthetic jet actuators". In 38th Aerospace Sciences Meeting and Exhibit, pp. AIAA Paper 2000–0903.

[20] Rathay, N. W., Boucher, M. J., Amatay, M., and Whalen, E., 2014. "Performance enhancement of a vertical tail using synthetic jet actuators". *AIAA Journal,* **52**(4), pp. 810–820.

[21] Ramos-Pedroza, N., MacKunis, W., and Reyhanoglu, M., 2015. "Sliding mode control-based limit cycle oscillation suppression for uavs using synthetic jet actuators". pp. 1–5.

[22] Deb, D., Tao, G., Burkholder, J. O., and Smith, D. R., 2008. "Adaptive synthetic jet actuator compensation for a nonlinear aircraft model at low angles of attack". *IEEE Transactions on Control Systems Technology,* **16**(5), pp. 983–995.

[23] Ciuryla, M., Liu, Y., Farnsworth, J., Kwan, C., and Amitay, M., 2007. "Flight control using synthetic jets on a cessna 182 model". *Journal of Aircraft,* **44**(2), pp. 642–653.

[24] Liu, S., Pan, S., Xue, F., Nay, L., Miao, J., and Norford, L. K., 2015. "Optimization of hot-wire airflow sensors on an out-of-plane glass bubble for 2-d detection". *Journal of Microelectromechanical Systems,* **24**(4), pp. 940–948.

[25] Strongrich, A., Pikus, A., Sebastião, I. B., and Alexeenko, A., 2017. "Microscale in-plane knudsen radiometric actuator: Design, characterization, and performance modeling". *Journal of Microelectromechanical Systems,* **26**(3), June, pp. 528–538.

[26] Shen, H., and Xu, Y., 2013. "Pressure and shear information based three-axis attitude control for a micro air vehicle". In AIAA Atmos. Flight Mech. Conf., pp. AIAA Paper 2013–4915.

[27] Shen, H., Xu, Y., and Dickinson, B. T., 2014. "Fault tolerant attitude control for small unmanned aircraft systems equipped with an airflow sensor array". *Bioinspiration & Biomimetics,* **9**(4), November, p. 046015.

[28] Mangalam, S. M., 2003. "Phenomena-based real-time aerodynamic measurement system (prams)". In 2003 IEEE Aerospace Conference Proceedings, Vol. 7, pp. 3347–3356.

[29] Cui, Q., Chandra, S., and McCahan, S., 2003. "The effect of dissolving salts in water sprays used for quenching a hot surface: part 2—spray cooling". *Journal of Heat Transfer,* **125**(2), April, pp. 333–338.

[30] Nasr, G., Sharief, R., and Yule, A., 2006. "High pressure spray cooling of a moving surface". *Journal of Heat Transfer,* **128**(8), August, pp. 752–760.

[31] Xie, J., Gan, Z., Wong, T., Duan, F., Yu, S., and Wu, Y., 2014. "Thermal effects on a pressure swirl nozzle in spray cooling". *International Journal of Heat and Mass Transfer,* **72**, June, pp. 130–140.

[32] Lunardelli, A., and Wentz, J., 2014. "Computational fluid dynamic analysis of eccentric atomization spray cooling nozzle designs for micromachining". *Journal of Micro- and Nano-Manufacturing,* **2**(2), June, pp. 021003–1–021003–10.

[33] Bar-Cohen, A., Arik, M., and Ohadi, M., 2006. "Direct liquid cooling of high flux micro and nano electronic components". *Proceedings of the IEEE,* **94**(8), August, pp. 1549–1570.

[34] Liang, G., and Mudawar, I., 2017. "Review of spray cooling – part 1: single-phase and nucleate boiling regimes, and critical heat flux". *International Journal of Heat and Mass Transfer,* **115**, December, pp. 1174–1205.

[35] Mertens, R., Chow, L., Sundaram, K., Cregger, R., Rini, D., Turek, L., and Saarloos, B., 2007. "Spray cooling of igbt devices". *Journal of Electronic Packaging,* **129**(3), May, pp. 316–323.

[36] Mudawar, I., Bharathan, D., Kelly, K., and Narumanchi, S., 2009. "Two-phase spray cooling of hybrid vehicle electronics". *IEEE Transactions on Components and Packaging Technologies,* **32**(2), June, pp. 501–512.

[37] Gioia, A. D., Brown, I., Nie, Y., Knippel, R., Ludois, D., Dai, J., Hagen, S., and Alteheld, C., 2018. "Design and demonstration of a wound field synchronous machine for electric vehicle traction with brushless capacitive field excitation". *IEEE Transactions on Industry Applications,* **54**(2), March/April, pp. 1390–1403.

[38] Myers, S., and Smith, A., 2009. "Demonstration of combined spray and evaporative cooling of an electromagnetic railgun". *IEEE Transactions on Magnetics,* **45**(1), January, pp. 396–401.

[39] Sarkar, S., and Selvam, R., 2009. "Direct numerical simulation of heat transfer in spray cooling through 3d multiphase flow modeling using parallel computing". *Journal of Heat Transfer,* **131**(12), December, pp. 121007–1–121007–8.

[40] Wang, J., Li, Y., Li, G., Xiong, K., and Ning, X., 2017. "Investigation of a gravity-immune chip-level spray cooling for thermal protection of laser-based wireless power transmission system". *International Journal of Heat and Mass Transfer,* **114**, November, pp. 715–726.

[41] Bhunia, A., and Chen, C., 2011. "On the scalability of liquid microjet array impingement cooling for large area systems". *Journal of Heat Transfer,* **133**(6), June, pp. 064501–1–064501–7.

[42] Xie, J., Tan, Y., Wong, T., Duan, F., Toh, K., Choo, K., Chan, P., and Chua, Y., 2014. "Multi-nozzle array spray cooling for large area high power devices in a closed loop system". *International Journal of Heat and Mass Transfer,* **78**, November, pp. 1177–1186.

[43] Pautsch, A., and Shedd, T., 2005. "Spray impingement cooling with single- and multiple-nozzle arrays. part i: Heat transfer data using fc-72". *International Journal of Heat and Mass Transfer,* **48**(15), July, pp. 3167–3175.

[44] Cheng, W., Zhang, W., Jiang, L., Yang, S., Hu, L., and Chen, H., 2015. "Experimental investigation of large area spray cooling with compact chamber in the non-boiling regime". *Applied Thermal Engineering,* **80**, April, pp. 160–167.

[45] Yan, Z., Duan, F., Wong, T., Toh, K., Choo, K., Chan, P., Chua, Y., and Lee, L., 2013. "Large area impingement spray cooling from multiple normal and inclined spray nozzles". *Heat and Mass Transfer*, July.

[46] Parisio, A., Rikos, E., and Gilelmo, L., 2014. "A model predictive control approach to microgrid operation optimization". *IEEE Transactions on Control Systems Technology,* **22**(5), pp. 1813 – 1827.

[47] Johansen, T., and Fossen, T., 2013. "Control allocatio - a survey". *Automatica,* **49**(5), May, pp. 1087–1103.

[48] Petersen, J. A. M., and Bodson, M., 2005. "Interior–point algorithms for control allocation". *Journal of Guidance, Control, and Dynamics,* **28**(3), May, pp. 471–480.

[49] Cui, L., and Yang, Y., 2011. "Disturbance rejection and robust least–squares control allocation in flight control system". *Journal of Guidance, Control, and Dynamics,* **34**(6), November, pp. 1632–1643.

[50] Kishore, W. C. A., Sen, S., G.Ray, and Ghoshal, T. K., 2008. "Dynamic control allocation for tracking time–varying control demand". *Journal of Guidance, Control, and Dynamics,* **31**(4), July, pp. 1150–1157.

[51] Johansen, T. A., Fossen, T. I., and Tøndel, P., 2005. "Efficient optimal constrained control allocation via multiparametric programming". *Journal of Guidance, Control, and Dynamics,* **28**(3), pp. 506–515.

[52] Benosman, M., Liao, F., Lum, K. Y., and Wang, J. L., 2009. "Nonlinear control allocation for non-minimum phase systems". *IEEE Transactions on Control Systems Technology,* **17**(2), pp. 394 – 404.

[53] Olfati-Saber, R., Fax, J. A., and Murray, R. M., 2007. "Consensus and cooperation in networked multi-agent systems". *Proceedings of the IEEE,* **95**(1), Jan, pp. 215–233.

[54] Ren, W., 2007. "Formation keeping and attitude alignment for multiple spacecraft through local interactions". *Journal of Guidance, Control, and Dynamics,* **30**(2), March, pp. 633–638.

[55] Cao, Y., and Ren, W., 2014. "Finite-time consensus for multi-agent networks with unknown inherent nonlinear dynamics". *Automatica,* **50**(10), pp. 2648 – 2656.

[56] Wang, Y., and Wu, Q., 2015. "Distributed robust h∞ consensus for multi-agent systems with nonlinear dynamics and parameter uncertainties". *Asian Journal of Control,* **27**(1), pp. 352–361.

[57] Zhang, H., and Gurfil, P., 2016. "Distributed control for satellite cluster flight under different communication topologies". *Journal of Guidance, Control, and Dynamics,* **39**(3), pp. 617–627.

[58] Chen, C. L. P., Wen, G., Liu, Y., and Wang, F., 2014. "Adaptive consensus control for a class of nonlinear multiagent time-delay systems using neural networks". *IEEE Transactions on Neural Networks and Learning Systems,* **25**(6), June, pp. 1217–1226.

[59] Qu, Z., 2009. *Cooperative Control of Dynamical Systems: Applications to Autonomous Vehicles*. Springer-Verlag London, London.

[60] Li, C., and Qu, Z., 2014. "Distributed finite-time consensus of nonlinear systems under switching topologies". *Automatica,* **50**(6), pp. 1626 – 1631.

[61] Mesbahi, M., and Egerstedt, M., 2010. *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, Princeton, NJ.

[62] Dong, X., Zhou, Y., Ren, Z., and Zhong, Y., 2016. "Time-varying formation control for unmanned aerial vehicles with switching interaction topologies". *Control Engineering Practice,* **46**, "January", pp. 26–36.

[63] Dong, X., Zhou, Y., Ren, Z., and Zhong, Y., 2017. "Time-varying formation tracking for second-order multi-agent systems subjected to switching topologies with application to quadrotor formation flying". *IEEE Transactions on Industrial Electronics,* **64**(6), June, pp. 5014–5024.

[64] Fax, J. A., and Murray, R. M., 2004. "Information flow and cooperative control of vehicle formations". *IEEE Transactions on Automatic Control,* **49**(9), pp. 1465–1476.

[65] Deng, J., Yuan, W., Luo, J., Shen, D., and Ma, B., 2011. "Design and fabrication of a piezoelectric micro synthetic jet actuator". In 2011 6th IEEE International Conference on Nano/Micro Engineered and Molecular Systems, pp. 301–304.

[66] Kundu, P. P., and Mitra, S., 2015. "Multi-objective optimization of shared nearest neighbor similarity for feature selection". *Applied Soft Computing,* **37**, "December", pp. 751–762.

[67] Slotine, J. J. E., and Li, W., 1991. *Applied Nonlinear Control*. Prentice Hall International, Upper Saddle River, NJ.

[68] Pilloni, A., Pisano, A., Orlov, Y., and Usai, E., 2016. "Consensus-based control for a network of diffusion pdes with boundary local interaction". *IEEE Transactions on Automatic Control,* **61**(9), "September", pp. 2708–2713.

[69] Bejan, A., 1993. *Heat Transfer*. John Wiley & Sons, New York, NY.

[70] Mudawar, I., and Estes, K., 1996. "Optimizing and predicting chf in spray cooling of a square surface". *Journal of Heat Transfer,* **118**(3), pp. 672–679.

[71] Rybicki, J., and Mudawar, I., 2006. "Single-phase and two-phase cooling characteristics of upward-facing and downward-facing sprays". *International Journal of Heat and Mass Transfer,* **49**(1), pp. 5 – 16.

[72] Henry, D., 1981. *Geometric Theory of Semilinear Parabolic Equations*. Springer-Verlag, Berlin.

[73] Mohapatra, S., Ravikumar, S., Ranjan, R., Pal, S., Signh, S. B., and Chakraborty, S., 2014. "Ultra fast cooling and its effect on the mechanical properties of steel". *Journal of Heat Transfer,* **136**(3), March, pp. 032101–1–032101–9.