A STOCHASTIC DISTRIBUTED CONTROL ALLOCATION METHOD USING PROBABILITY COLLECTIVES

by

CHRISTOPHER MICHAEL ELLIOTT

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2016

To the reader...continue to explore, engineer, and improve tomorrow for our fellow human and all of life.

ACKNOWLEDGEMENTS

I have always been taught from the beginning to be grateful to God, Family, and Country. With time, I have learned why this teaching is important for these forces have provided the foundation on which all of my achievements stand. Ever enduring support, freedom to think, and above all, love have provided the fuel for my drive on this research endeavor, and for that I am humbled and grateful. My parents, Mike and Brenda Elliott, have always promoted my interests of study in aerospace engineering. My Pop and Grandma, Howard and Elnora Cordova, have backed me from birth. I can thank my Elliott grandparents, Virgil and Grace, for their strong work ethic in my blood. Rita and Charley Cherry have been instrumental in my success, as have many family friends throughout my life... Burrow, White, Hartness, Farris, Carter... to name a few. I only regret that I cannot name them all.

My gratitude is with the engineering faculty of the University of Texas at Arlington who patiently teach graduate lecture through the distance education system. From firsthand experience, thanks to Dr. Frank Lewis and Dr. Mike Niestroy on opportunities to deliver a few guest lectures myself, teaching via camera requires additional coordination beyond a conventional classroom-only delivery. Thank you to Dr. Wilson, Dr. Dogan, Dr. Shiakolas, Dr. Nomura, Dr. Liu, Dr. Lewis, Dr. Hullender, Dr. Massa, Dr. Tong, Dr. Subbarao, and Dr. Manry for the years of online learning. I was a lone student on the other side of the camera, and I appreciated each and every one of your teaching approaches. This distance learning system was my key enabler for completing graduate study throughout a demanding schedule with full time employment at Lockheed Martin Aeronautics.

I am especially grateful to Lockheed Martin for tuition support through graduate school. Lockheed Martin promotes higher learning amongst the staff and offers a remarkable benefit by providing this incentive to employees aspiring for advanced degrees. My career to date at Lockheed has been shaped by many key individuals over the years with whom I hold gratitude. The first few that I need to mention are not with the company, but their guidance was instrumental on my path. Bobby Brown, trumpet instructor and early mentor, inspired and coached a work ethic applicable for conquering any challenge. Dr. Scott Fish gave me the first engineering research opportunity at the Institute for Advanced Technology (IAT) in Austin, TX during my undergraduate years. The reward was priceless experience by allowing a young student to participate in staff meetings and contribute to simulation development and experimental findings. Dr. Fish was an inspiring first manager who would explain the most difficult solutions would at times become apparent by simply stepping out of the office, a practice I've come to appreciate. Troy Savoie was a key mentor during this time at IAT. Troy, a graduate student who later finished a PhD at MIT, patiently worked with me for an initial acclimation to MATLAB and the mechanics of the callback function for building graphical user interfaces. This was an important lesson that ended up bridging the transition in my life between graduation and career. The transition was enabled by Charles and John Phipps with RadX Corporation. John Selmarten with the United Space Alliance granted me the honor of working with his team in the OZ2 International Space Station Program Office at the NASA Johnson Space Center (JSC) in Houston, TX. When a new exciting opportunity in Robotics presented itself, thanks to a UT Austin colleague Andrew Cheang, John helped me make the transition. Andrew Cheang, Kristian Mueller, and many others in the Robotics and Automation ER5 department at JSC were indispensable examples in terms of inspiration to always continue learning. Thanks to Ryan Necessary, a col-

league and good friend at JSC, for the intellectual conversations and, most of all, the lifeboat through turbulent waters.

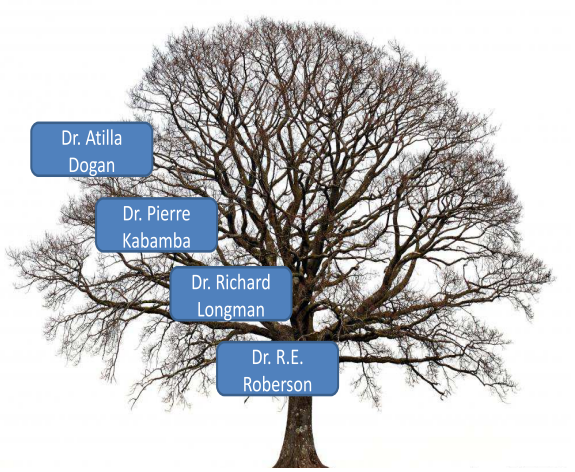
I am indebted to Dave Bodden for providing the spark to ignite my career transition from robotics and space in Houston to the atmospheric sciences at LM Aeronautics in Fort Worth, TX. Thanks to Mark Tibbs, a colleague and good friend, for fueling the transition after the ignition. Mark's world class instruction over the years in flight dynamics and control and McRuer approximate factors has been priceless as well as the fellowship. Thanks to Ken Dorsett for insight into the intuitive back-of-the-envelope nature of the stability and control discipline. I am grateful to Dr. Mark Hollingsworth for the routine exchanges we have had over the years on technical hurdles with this research and many other topics. Thanks to Jennifer Bennington, Nick Papayianis, Rick Brouse, Tom Schech, Ed Attel, and the many others with the Block 60 Flight Controls completion team, I am very proud of our accomplishments. My gratitude goes to Christopher Stewart for helping me transition from the F16 Program to the Advanced Development Programs, in line with my interests. Thanks to Dr. Bob Boyd, John Morehead, and Val Evans for the opportunity to contribute on a very interesting hybrid airship platform, remotely from Texas. I am indebted to my friend and colleague, Richard Hauffe, for the insightful research throughout this teamwork. I am thankful to Dr. Mike Niestroy for the recent opportunities in Skunk Works research, especially in Extremum Seeking Control (ESC) which catalyzed a portion of this study, and, again, for the opportunities to assist with delivering adjunct lectures. I am also thankful to Navair's Mike Olzewski and Richard Huang with the Propulsion and Power team, and Nelson Brown with NASA Armstrong for supporting recent ESC research and to Dr. Eric Charlton and Dr. Dennis Finley for the help. Thanks to Wes Zwerneman, another colleague and good friend, for the team leadership and exceptional patience. Thanks to Shawn Whitcomb and Dr.

Tim Woods for always encouraging me to weather the storm and complete graduate school. I am grateful to Ben Brown for taking the time to pick up the phone to help with my questions when he was already busy building the most advanced fighter in the world. Thanks to Neil Hall for the recent pursuit and win for research support and for multiple opportunities to work and learn with the LM Marietta experts.

My sincere appreciation goes to Greg Tallant, a Technical Fellow in Quantum Information Science and Flight Controls at LM Aero and also a good friend. I appreciate the years of research we have conducted and the continued inspiration to press the envelope, both on the career and academic front, and the library of journal articles to study. I am also very thankful for the budget support which allowed me to finish this study. Thanks to Peter Stanfill, who is a true master of ceremony beyond his expertise in flight controls. Peter's positive attitude is healthy for any team. Thank you to Dr. Kristen Pudenz for the helpful pointers in latex, and the insight into the best method to form citations. Thanks to Todd Belote, a colleague and good friend for years. Brainstorming discussions on ubiquitous control were always valued. I acknowledge the work of Dr. Federico Spedalieri, with the Information Sciences Institute, for introducing me to the Probability Collectives field, a key element in this research. Thanks to Dr. Jordan Kyriakidis, Dean Tsaltas, and Dr. Steve Adachi for the years of collaboration, best of luck on research ahead. Finally, thank you to Lockheed Martin's Dr. Ned Allen, a visionary behind many exciting projects I have had the opportunity to contribute to with Skunk Works research.

The path on my aerospace industrial career to date has been rewarding and has given me additional appreciation for the theory behind the practice. I offer a warm thank you to my committee (Dr. Atilla Dogan, Dr. Kamesh Subbarao, Dr. Ashfaq Adnan, Dr. Alan Bowling, Dr. Frank Lewis, and Dr. Michael Niestroy) for the review of this dissertation. I am especially grateful for the work I have accomplished

with my advisor, Dr. Atilla Dogan. Thank you for your patient listening, direction, and friendship over the years. I also acknowledge the contributions of our academic family, from my advisor's advisor and beyond, and welcome the opportunity to take part on another branch of this tree of progress. On this tree, I acknowledge the prior work of Dr. R.E. Roberson with Satellite Guidance and Control at Washington University in 1952; Dr. Richard W. Longman on his Generalized Approach to Gravity-Gradient Stabilization of Gyrostat Satellites at the University of California in 1969; Dr. Pierre T. Kabamba for his Contributions to Reduced Order Control Theory at Columbia University in 1981; and finally to Dr. Atilla Dogan for Guidance Strategies for Microburst Escape, University of Michigan in 2000 [1, 2, 3, 4, 5]. It is an honor to join the limb of this intellectual genealogy, and I look forward to influencing the next generation of engineers as you have influenced your students.



Finally, and most importantly, thank you to my family, Zachary, Shannon, Avery, Brandon, Kaylin, Colton, and especially my wife Kelley for enduring this decade of hard work. You made this dream possible.

November 30, 2016

ABSTRACT

A STOCHASTIC DISTRIBUTED CONTROL ALLOCATION METHOD USING PROBABILITY COLLECTIVES

Christopher Michael Elliott, Ph.D.

The University of Texas at Arlington, 2016

Supervising Professor: Atilla Dogan

The configuration of the aerospace vehicle of tomorrow marches forward in complexity in step with technological advances in computation, materials, propulsion, and beyond. One of many attributes resulting from this evolution is the multiple effector concept, replacing the traditional aircraft design approach of a dedicated four channel mixer where pitch-roll-yaw-speed is controlled by elevator-aileron-rudder-engine, respectively. Instead, for example on the propulsion front, a distributed suite of smaller engines can substitute the larger power plant and can be used in concert for multiple axis control with both simultaneous, collective thrust for force generation and differential cyclic thrust for moment generation. The effector surface is also a candidate for distribution. Rather than one large moving panel, the notion of a smart wing with many small effectors has been explored due to potential benefits from structural mode suppression, aerodynamic optimization via active flow control, improved radar and observability signature, adaptive geometrical morphing to the flight condition, and more. Beyond these gains, the distributed approach also offers an element of resiliency in tomorrow's platform by reducing the critical impact due to

the loss of a single dedicated effector. Additionally, the science of multi-agent systems is rapidly advancing and there is a growing interest in individual vehicles working together as a collective to synergistically accomplish a mission. In this work, the applicable collective mission is to jointly stabilize and control an aerospace platform, specifically a generic hovercraft where the multi-agent system is a distributed effector suite of small, electric engines.

The method of solving for the required individual effector positions in order to achieve a set of required accelerations on a vehicle is referred to as the control allocation problem (CAP) and increases in complexity as a function of the number of effectors for allocation. With the trend towards a future generation of complex distributed effector suites on the horizon, traditional centralized flight control architectures of current day may become computationally intractable, and advanced methodologies for solving the CAP are warranted. This research focuses specifically on this problem and considers the underdetermined aerospace vehicle configuration where the CAP needs to solve for a number of effectors greater than the number of desired acceleration channels.

The main contribution of this dissertation is the formulation of a novel distributed control allocation method for an aerospace vehicle using a modified formulation of Wolpert Probability Collectives stochastic optimization. The method is presented in detail and applied as a distributed flight control architecture for solving the CAP, where the multi-agent system applied is a collective of local effector controllers. Each individual agent is dedicated to a physical engine controller and is responsible for modulating the local thrust required to maintain overall vehicle stability with respect to translational and angular axes. Communication between agents shares the expected value of the current strategy in order to allow local objective evaluation and optimization. This work highlights the benefits of a distributed allocation

approach from the advantage of parallel processing; to failed effector robustness; and the ability to evaluate nonlinear control effectivity for L_2 optimization. A MATLAB Simulink® toolbox is constructed to enable future incorporation into aerospace modeling and control environments. Finally, the method is applied to a hovercraft vehicle demonstrating the viability of distributed control allocation approach for inner loop stability augmentation.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Motivation

The advancing forefront of aerospace technology continues to expand the mission effectiveness of the air vehicle. Novel invention continuously presses the bounds on size, weight, power, and cost (SWAP-C) opening the door for new mission capabilities. These attributes of an air vehicle ultimately define capability and are often linked directly to operational constraints for the flight envelope [7]. While improvements in this evolution are due to a vast domain of cross-disciplinary contributions from materials and structures, propulsion, aerodynamics, computer science, estimation, and beyond, a recent key enabler has been the advancement of the small electric engine for distributed propulsion [8].

The premise with the distributed propulsion application for air vehicles is a heavy, primary power plant can be alleviated by a number of smaller engines which can offer reduced noise levels; improved terminal phase aerodynamics for shorter landing and take-off requirements; and better overall fuel consumption characteristics [9]. Taking a step further, recently investigators at the NASA Langley Research Center are compounding the benefits of distributed propulsion with the benefits of all-electric propulsion, coining the phrase Distributed Electric Propulsion (DEP). Electric propulsion offers extremely quiet thrust at much higher efficiencies with respect to conventional mixed flow turbofans and combustion-driven rotary piston engines. Also, electric motors can be scaled to small, compact sizes without a dramatic impact on the power-to-weight ratio, effectively opening the door for distributing a large

number of engines on an airframe [10]. In brief, becoming more apparent is the need for a flight control system to manage many smaller effectors rather than a few primary devices. This motivation can be described as a push from the larger and more expensive components to the domain of smaller-simpler-cheaper. Cutting edge actuation is also opening the door for a future aerospace vehicle generation of distributed effector suites. The application domain is vast and, to name a few, ranges from aeroelastic structural mode suppression; to miniature plasma jets for refined control at hypersonic speeds; to morphing wing concepts to augment the operational flight envelope [11, 12, 13, 14].

The common denominator with these potential distributed effector suites of tomorrow is the need for advanced control allocation methods which accommodate a non-traditional number of engines and surfaces. Especially important is the need to decentralize the current day centralized flight control architecture. Clearly, computational burden will increase for a centralized flight control computer (FCC) managing a large number of effector channels, and real-time solutions may even become intractable. Communication bandwidth with a central FCC will also pose a potential bottleneck as every input and output data stream across the vehicle is routed to a single point. One single safety-critical hardware component managing a full suite of distributed effectors also illuminates the issue of resiliency, where the loss of the primary FCC results in the loss of the vehicle. On the other hand, the loss of a single effector amongst a distribution of many would not be expected as critical. This balance in robustness must be addressed where the future distribution of effectors requires the future distribution of the controller to some degree.

In summary, advanced methodologies for control allocation may be required in response to technological trends driving tomorrow's complex effector suites. These trends are enabled by the continuation of processor miniaturization; electric power

and propulsion advancement; and the evolution of materials and structures technology. Future generation platforms could involve a significant number of smaller actuators distributed at multiple locations on a rigid body, and reveal that traditional centralized flight control architectures are intractable, if not risky solutions.

1.2 Research Objective

The ultimate goal of this research is to innovate a novel approach to control allocation for future aerospace vehicles with distributed effector suites, where the control allocation problem is generally stated as "determine the required configuration for each individual effector (engine or actuator) in order to achieve a set of desired control or virtual commands, given the control effectiveness mapping" [15, 16, 17, 18]. These hypothetical aerospace platforms are forecast to exhibit a large number, m, of distributed effectors, and therefore, the "underdetermined" control allocation problem is the focus objective of this study. Three research objectives are delineated in order to achieve the primary goal. First, the novel approach should identify a decentralized and distributed control methodology in order to (i) garner the advantage of parallel processing to address computational and network burdens expected with a centralized architecture; and (ii) promote balance in robustness with respect to a distributed effector suite. Traditional approaches today to the control allocation problem (CAP) employ a priori mixing; direct matrix inversion; linear programming and quadratic optimization techniques such as weighted least squares with constraints. The second objective of this research is to explore and leverage stochastic optimization techniques to reduce the need for a priori design considerations for various reconfiguration control modes with effector failures; and garner the advantage of covariance estimation in solution. Rather than solely providing a solution, the quantification of uncertainty in the solution is important. Finally, the third objective in this research is to explore a

hypothetical hovercraft as an application to demonstrate this novel control allocation technique. In brief, the following list enumerates the objectives for this research effort to innovate a novel approach to the control allocation problem for future aerospace vehicles with distributed effector suites:

1. Identify, design, and enable decentralized, distributed control allocation
2. Leverage the benefits of stochastic optimization techniques in the distributed control allocation approach
3. Demonstrate the distributed control allocation approach with the simulation of a hypothetical hovering platform

The expected significance for achieving the research objectives is to establish a building block for an infrastructure for next generation distributed inner loop stability augmentation systems (SAS) for the flight control domain. While the objectives in this research focus on the control allocation element of the SAS, continued progress is expected with further distribution of closed loop functionality beyond this primary element as discussed at the conclusion of this dissertation in Ch. 7. The feedback loop closure consists of control laws and actions, allocation, sensors, signal conditioning, and estimation and are all considered viable candidates for distribution to a local domain, residing closer to the vehicle effector. This research and the ultimate complete decentralization of the aerospace flight control system will significantly impact future vehicle designs and considerations.

1.3 Literature Survey

The uniqueness of this work was accomplished by performing a survey over three subject fields: (i) Distributed Effectors for Aerospace Vehicles; (ii) Control Allocation; and (iii) Distributed Optimization with Probability Collectives. Due to the broad nature between these three immense topics, the body of this dissertation

includes additional literature survey results throughout relevant discussions. This section summarizes significant works with respect to each field as a entry point to the original contribution of this research effort described in Section 1.4.

1.3.1 Distributed Effectors

Motivation for research in distributed control allocation is immediately witnessed with the current storm of new flying robots, where centralized control mechanisms are becoming more and more intractable due to the sharp rise in number of agents and effectors to control [8, 19, 20, 21, 22]. One key enabler for this wave is the advancement of the electric engine and continued miniaturization of technology opening the door for new complex aerospace concepts with non-traditional effector suites [7, 10, 23]. Beyond the power of automation with new intelligent, small vehicles, many benefits have been identified by employing smaller, more efficient effectors on the aerospace vehicle from noise reduction, lower fuel emissions in [23] to aerodynamic drag reduction and improved energy efficiency [6, 24, 25, 26]. The all electric aircraft poses new challenges as described in [9], especially when the primary propulsive device is replaced with a distribution of many, smaller thrusters.

Beyond small electric engines, advancement in structural materials is opening the door for smarter wings, which may optimally morph to a flight condition [11, 12, 27]. This capability will require an ability to control many independent variables accordingly and will most likely become quickly intractable with centralized control architectures, especially with the emergence of nanotechnology and nanostructures [28]. Aeroelastic suppression is a current interest, where structural rigidity of a wing is reduced to improve endurance, at the expense of more control effectors to actively attenuate the bending and torsional modes [29, 30, 31]. Modeling for these type of vehicles will be a challenge [14] and on-line system identification may be an

important application beyond control allocation for future distributed effector suites [32]. Additionally, distributed systems invite the requirement for the consideration of communication topologies and consensus dynamics [33, 34, 35, 36, 37], especially as the number of effectors increases as sizes decrease. Smaller, non-traditional actuators are becoming more and more common and can be used to actively control the flow over the wing [13].

1.3.2 Control Allocation

The control allocation problem (CAP) is the challenge of real time optimization for actuator coordination as referred by [38] and hosts a wealth of literature as a critical centerpiece in the field of air vehicle control. Due to the motivation aforementioned where the number of effectors continues to rise due to technological advance, the primary focus of this work is on the over-actuated control allocation problem as described in [39]. In this case, often the problem is formed as an optimization problem in order to find the best solution to minimize a cost consisting of virtual commands (e.g. desired accelerations) and desired effector positions [15, 16, 17, 40]. While some authors consider the nonlinear control allocation problem (CAP) [41, 42, 43], the majority of available literature focuses on the linearized CAP [18, 44, 45]. In both linear and nonlinear forms, control allocation must account for the limits of the effector devices in that a solution which requires magnitudes beyond position or rate limits will not be achievable.

Control allocation has a rich history in the field of robotics and manipulators, and usually takes the name of the inverse kinematics problem for determining the required joint angular positions in order to achieve the desired end effector position [46]. In some approaches, the mathematical groundwork for solving such problems makes use of a generalized Jacobian inverse matrix, that may be non-square in the case

of highly redundant manipulators with a significant number of degrees of freedom. In [47], the problem is partitioned into a macro and mini concept to improve the understanding of the overall mechanism as a control design aid.

Bordignon describes a range of techniques for solving the CAP as (i) generalized (Jacobian) inverse solutions; (ii) daisy chaining where effectors are sub-grouped by establishing priority channels with respect to the task (e.g. thrust vectoring an engine nozzle may be a last resort for pitch control and would only be used if the first group of effectors fail to accomplish the pitching task); (iii) cascading generalized inverse solutions, where if a solution saturates an effector, the solution is iterated with the effector fixed at the saturation point (also commonly referred to as a redistributed pseudo-inversion approach); and (iv) null-space intersection methods which determine a control policy to satisfy both the null-space of the control effectivity subspace and a higher dimensional polytope of desired accelerations (often called direct control allocation) [48]. Bodson also provides a helpful survey of the known control allocation techniques for the flight control application, and details the control allocation optimization problem where an exact solution may not exist and an error or control effort is minimized as the best choice [16].

The inclusion of position and rate limits on the effector solution creates the constrained control allocation problem. A graphical geometrical depiction is provided in [49] where the valid solution space and constraints are presented on phase portraits. The authors in [50] extend a further step beyond these nonlinearities and include actuator dynamics in a direct control allocation approach where lower frequency desired dynamics are allocated to the slower effector channels. In a technique described as Structured Adaptive Model Inversion (SAMI) in [51], the ability to estimate the pseudo-inversion of a linearized control effectivity mapping online is improved by avoiding the actuator saturation regime with a command reference

hedging approach. In [52], a weighted pseudo-inversion control allocation method is designed with weights that incorporate the effector constraints. The method is shown to improve control of a hovering hexacopter with respect to classical pseudo-inversion [52].

Nonlinear dynamic inversion is employed (with nonlinear control allocation) in [53], and the nonlinear control effectivity mapping is approximated using a multivariate simplex spline. The approach is shown to accomplish the control of an F-16 [53]. Approximating the nonlinear control effectivity mapping on-board an air vehicle is a challenge in and of itself. In [54], a series of feedforward neural networks fits are used to capture force and moment data for use in a dynamic inversion scheme. Additionally, the work presents a novel dynamic control allocation synthesis method which trades control redundancy for closed loop stability margin [54]. In [43], an approach to nonlinear control allocation is presented where nonlinearities in multi-dimensional force and moment lookup tables are treated as piecewise linear segments, and the authors cast the entire problem as a mixed integer linear program enabling an exact allocation solution.

Generalized inverse allocation techniques tend to offer speed advantage over computationally expensive direct control allocation approaches [55], however even generalized inversion can be costly especially in cases where effector saturation is encountered and the cascading inversion approach requires iteration [56]. In [38], two control allocation optimization algorithms are presented using an active set method and an improved primal-dual interior point method for up to one hundred actuators. The active set shows a linearly increasing relation between computational time and number of effectors while the primal-dual interior point method exhibited a decreased slope [38], although all of the approaches consist of a centralized architecture.

1.3.3 Probability Collectives

The Wolpert Probability Collectives (PC) optimization method emerged from the science of collective intelligence (COIN) in circa 1999 from Dr. David Wolpert with Stanford University and the NASA Ames Research Center. Analogous to the ingenuity of the Laplace transform to solve a differential equation algebraically in a new domain, the PC framework transforms from the variable space to the probability space and accomplishes optimization with a distributed reinforcement learning approach. The distributed nature of the algorithm is apparent with each variable in the optimization problem acting as an agent (i.e. player) working to maximize an individual reward, effectively driving the total system to an optimal equilibrium [57, 58, 59, 60, 61].

One beauty of the transformation to the probability domain is the fundamental idea that the agent works to maximize local knowledge or information of the optimizing argument to the cost. Initially, the lack of information for the agent is modeled as a uniform probability distribution where only the search lower and upper bounds are specified as the feasible search area[1]. As the PC algorithm recursively learns, each agent refines a probability distribution governing the strategy of best possible actions. The refinement occurs in a manner in which syntactic information contained in an empirical probability distribution is maximized by using the concept of Shannon entropy [60]. Fig. 1.1 depicts this general notion in which the PC algorithm works to maximize information at the agent level. Clearly the final distribution with the lowest covariance is the most informative. Convergence of the PC algorithm is declared when the agents detect that no additional reward is available with con-

[1]Here the probability functions are described as discrete distributions which are used primarily throughout this work. The Wolpert PC method is also applicable to the continuous domain with probability densities.

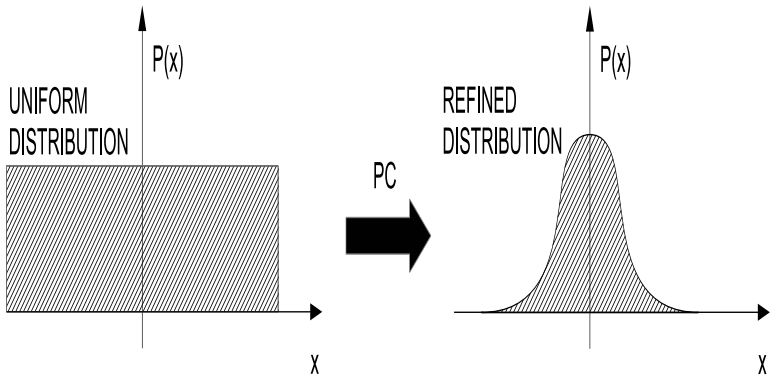


Figure 1.1: The Probability Collectives Method Iteratively Minimizes Uncertainty by Refining an Empirical Uniform Distribution with Information from Drawn Samples

tinued refinement (i.e. the expected value of the local cost reaches a steady state). At this point, a stochastic inversion is performed by sampling the final distribution, effectively transforming the solution back to the variable domain[2]. The key point is with high probability, the final samples represent an approximation of the optimizing argument to the total system. Fig. 1.2 illustrates the PC optimization method where the original problem is to determine the minimizing value of x, denoted as x^*, in the unconstrained, but not necessarily smooth, objective function $G(x)$ [62]. The Wolpert transformation, $\mathcal{W}$, casts the problem into a smooth optimization landscape, removes local extrema, and establishes a framework that is robust despite potential poor structural properties in the original objective [62, 63]. In the probability domain, the problem becomes to determine the parameters θ governing the probability distribution, $\mathbf{q}_\theta(\mathbf{x})$, which minimize the expected value of the objective, $E_{\mathbf{q}_\theta(\mathbf{x})}G(\mathbf{x})$,

[2]An alternative inversion method is to compute the expected value of the final probability distribution.

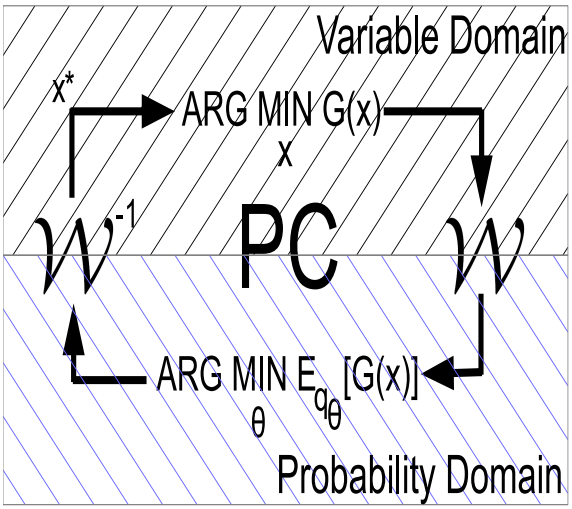


Figure 1.2: The Probability Collectives Method Invokes a Wolpert Transformation of the Objective Function from the Variable Domain to the Probability Domain, Searches for the Best Distribution, and Samples the Optimizing Variable Argument with a Stochastic Inversion

evaluated with respect to the distribution.[3] Finally, at the conclusion of the PC method, x^*, is sampled from $\mathbf{q}_\theta(\mathbf{x})$ as the minimizing argument.

PC has strong relations to parametric machine learning, information theory, game theory, and distributed multi-agent systems, and is similar to other optimization techniques such as Response Surface Models (RSM) [64, 65], simulated annealing [66, 67, 68, 69], cross-entropy method [70, 71, 72], estimation of distribution algorithms (EDA) [73, 74, 75], genetic algorithms [76, 77], and particle swarm optimization [78, 79, 80]. However, PC is unique with respect to many of these other approaches

[3]Typically the θ parameters of the probability distribution are more relevant in the method of immediate sampling PC where a Gaussian shape is assumed for the solution, and θ represents the optimal first and second moments of the distribution (mean μ and covariance Σ). In general for delayed sampling PC, the search is for an arbitrary, empirical distribution $\mathbf{q}(\mathbf{x})$ and the θ subscript is dropped.

with optimization performed directly on the underlying probability distribution [60, 63]. PC promotes distributed architectures for optimization due to inspiration from bounded rational agent game theory, where agents (i.e. players) compete or cooperate by minimizing a local cost. The joint strategy the agents converge to resembles a Nash equilibrium, with the exception of the case where full rationality is not available due to bounded local information. The joint strategy of the multi-agent system is coupled over time with individual strategies dependent on the prior actions of other agents. PC captures this coupling aspect indirectly by evolving the distributions accordingly [59]. However, at a specific instance in time, PC models the joint strategy of agents as a product distribution, which paves the road for a distributed optimization approach, offering additional advantage of a decentralized computational structure with reduced latency between agents [61]. Extensive literature by Wolpert et al has been published on this idea, described as the theory of probability Lagrangians and product distribution theory, which constitutes a significant portion of the PC mathematical framework [59, 81].

Advantages	**Disadvantages**
Nonlinear, Discontinuous Objectives	Many Objective Evaluations
Insight Into Estimation Covariance	Bounded Rational Agents
Decentralized Optimization	Complex Parameter Configurations
Reduced Network Topology	
Failed Agent Robustness	

Table 1.1: Positives and Negatives of PC

Table 1.1 summarizes the advantages and disadvantages of employing the PC method to an optimization problem from the perspective of use in a real time flight

controls environment. As previously outlined, the transformation to the probability domain provides robustness to noisy and even discontinuous objective functions as detailed in [62]. The final distribution for the minimizing argument, $q_\theta(x)$, also provides key insight into the uncertainty or covariance of the solution [58]. Additionally, many benefits could be potentially gained from the PC-enabling distributed architecture where a centralized flight control computer is eliminated. Instead of a primary hub of communication relaying information to all vehicle effectors, a distributed approach can simplify network topologies where agents pass information amongst required neighbors. In this environment, the PC approach may provide increased resiliency due to a failed agent (i.e. erroneous actuator or engine) as discussed by Kulkarni, Tai, and Abraham in [61].

Numerous successful applications of PC have been published. A convenient overview summarizing many of these examples is provided in [61, 82], as well as a demonstration of multiple applications such as the Traveling Salesman Problem (TSP), Vehicle Routing Problem (VRP), and various mechanical design problems (10 to 45 bar structural trusses, concrete load bearing beams, and steel cantilever beams among others). Bieniawski, Kroo, and Wolpert originally applied PC to the structural truss problem with a 10 bar example in [60], and extensions have been applied in [83, 84, 85]. An example of a Sensor Network Coverage Problem (SNCP) was also demonstrated with PC in [61, 82]. Zhang et al in [86] applied PC in a Coordinated Multiple Target Assignment Problem (CMTAP) for air-combat decision making. Wireless network routing and resource allocation was optimized with PC by Mehr and Khalaj in [87] and Ryder and Ross in [88]. An application of the airplane fleet assignment problem was solved with PC by Wolpert et al in [89] demonstrating the benefit of a decentralized, parallel optimization in the face of a complex combi-

natorial optimization problem [61, 82]. In [90, 91, 92], Huang et al modify PC for dynamic optimization where the objective is time variant.

Bieniawski in [57] implements PC in a flight control architecture to suppress aeroelastic structural modes and improve gust alleviation on a flying wing planform UAV with a distributed effector suite, referred to as MiTEs (Miniature Trailing Edge Effectors), marking the original exploration of PC for control of an aerospace vehicle. In the aeroelastic suppression experiment, the PC method was compared to two baseline control techniques, Linear Quadratic Gaussian (LQG) and a Direct Policy Reinforcement Learning (DPRL) method. All three controllers were successful in stabilizing the unstable open loop system and increased the maximum flutter airspeed by approximately 25%. While the centralized LQG controller exhibited the best overall performance, the more appropriate comparison was PC with respect to DPRL due to the fact that the PC and DPRL implementations employed the equivalent optimization objective as a finite time horizon quadratic reward function with a future discount factor. The LQG synthesis employed a standard quadratic performance index (PI) as a control case in the experiment. In summary, the PC method was shown to successfully stabilize the experimental platform with distributed optimization and even indicated a slight advantage in robustness with off design airspeeds tested after gain calculation at a nominal condition [57].

While the first exploration of PC for control of an aerospace vehicle was accomplished by Bieniawski in [57], the implementation focused on control via a bang-bang type solution with the MiTEs. PC was employed to discretely set the surface positions to full trailing edge up or down based on the cost function, described as the DPRL objective function [57]. Here in this work, PC is employed as a primary component in an inner loop SAS controller, where distributed control allocation optimization determines the local effector position with respect to a range of possible values bound by

position and rate limits, numbering to thousands of numerically discrete possibilities. Additionally, this work addresses the nonlinear control allocation problem.

1.4 Original Contribution

As a result of accomplishing the objectives delineated for this study as described in Section 1.2, the original contribution of this work is the formulation of a novel distributed control allocation method for an aerospace vehicle using a modified formulation of Wolpert Probability Collectives stochastic optimization[4]. This distributed approach works at the local agent scope, accounts for effector constraints, and works to minimize a desired optimization objective consisting of anticipated acceleration error and preferred solution error, based on knowledge of the control effectiveness mapping from a set of virtual commands to each individual effector. Additionally, the distributed architecture of the approach promotes futuristic decentralized flight control architectures by migrating more computation to the local effector. This decentralization also brings an inherent robustness to effector failure, where other remaining effectors continue to operate individually despite the fault, alleviating the need for a priori design of flight control reconfiguration modes. This research employs PC as distributed inner loop components in a closed loop flight control stability augmentation system, where the distributed control allocation optimization determines the required local effector position to minimize the mixed L_2 objective consisting of acceleration error and total effector position. A modified form of the delayed sampling method of PC is built as a MATLAB Simulink® toolbox and includes provisions for bound-

[4]Modifications to the method include an adaptive technique to refine the sampling domain; a mechanism to iteratively latch the best local agent strategy for the L_2 control allocation optimization application; and the broadcasting of the expected value of the strategy over the distributed network.

ing position and rate limits[5]. The method is employed to determine the numerical solution between these position and rate bounds over a continuous range, by searching through an admissible discretized space with a specified number of finite points. Additionally, this work addresses the control allocation problem when the issue of nonlinear and non-affine control effectivity mappings become a factor, and failure scenarios are studied emphasizing the benefit of the distributed approach. Multiple variants of PC are explored, dating beyond the original work in [57], and a modified implementation is applied in this work and integrated into a hypothetical hovercraft control system. In brief, the original contributions resulting from this research effort are tabulated below in Table 1.2.

Original Contributions Resulting from Research Effort
First Implementation of Wolpert PC for the Nonlinear Control Allocation Problem
Modifications to the Wolpert PC Algorithm from Original Inception
Novel Distributed Control Allocation Technique with Benefits of Parallel Processing and Failed Agent Resiliency
New Simulink toolbox implementation for deployment to applications
Hypothetical generic vehicle environment based on prior work [93, 94]

Table 1.2: Original Contributions Resulting from Research Effort

1.5 Overview of Dissertation Layout

The layout of this dissertation is as follows. First, as a motivation to investigate a novel control method, Distributed Electric Propulsion (DEP) and other distributed effector applications are discussed in detail in Ch. 2. Second in Ch. 3, the control

[5]The distributed control allocation MATLAB Simulink® toolbox created as a result of this work is conducive to future modeling, simulation, and control design efforts, and is employed at the conclusion of this dissertation on a hypothetical hovercraft vehicle.

allocation problem (CAP) is defined and common solution methods are summarized for the underdetermined system where m effectors is greater than k desired control channels [16]. Next, in Ch. 4, a stochastic optimization technique is explored, referred to by the authors as Wolpert Probability Collectives (PC) [57, 95]. Ch. 5 then applies the PC approach to the CAP and presents simulation results for a simple example, followed by a more complex hypothetical hovercraft in Ch. 6. Finally, Ch. 7 summarizes research findings and outlines a road map for future work. Additional detailed write ups are referenced within this dissertation and provided as appendices at the conclusion of this document.

CHAPTER 2

DISTRIBUTED CONTROL

This chapter introduces the concept of a distributed effector suite and further explores electric propulsion as the example category of interest and application in this dissertation. Following the introductory discussion, the equations of motion for a distributed propulsion suite are presented.

2.1 Distributed Effectors

Since the dawn of flight, scientists have utilized the concept of altering the lift distribution on the airfoil as a primary method of vehicle control. While the conventional trailing edge moving surface was believed to be invented in 1870 by Richard Harte, Wilbur and Orville Wright pioneered an early form of distributed control, in a continuous sense, by warping the wing box for lateral steering. The asymmetric contortion of the structure resulted in a change in the spanwise lift distribution, as depicted in Fig. 2.1, effectively imparting a lateral-directional moment on the vehicle [96].

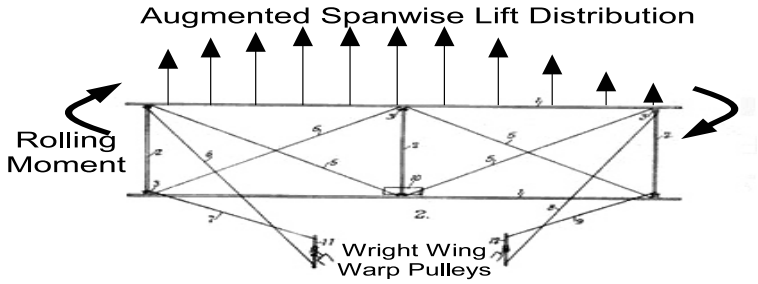


Figure 2.1: The Wright Brothers Wing Warp Cable-Pulley System Augmented the Spanwise Lift Distribution for Roll Control [96]

Advanced wing morphing remains a research thrust today and is recognized as a powerful technique for adapting to the current mission priority, with multiple benefits ranging from reduced distance requirements for landing and takeoff, improved endurance and cruise characteristics, greater maneuverability, and beyond. In essence, next generation reconfigurable systems refine the lift profile to the optimal distribution for the specific task at the current flight condition. Lift profile refinement via geometrical morphing, or variable wing geometry, has been operationally successful to a limited degree historically in that only a small number of effectors are enabled for the reconfiguration. A famous example is the U.S. Navy F-14 Tomcat with the actuated sweep wing, scheduled as a function of Mach number in order to optimize the lift to drag ratio [97]. This example is essentially employing a single effector, the symmetric sweep angle, and represents a simplified form of adaptation. Another commonly encountered example in operation is the F-16 fighter, where two effectors, the leading and trailing edge flaps, are used to optimize the chord-wise lift distribution at the terminal takeoff and landing phase [27].

These examples, however, only employ conventional control surfaces and are not considered by the current day research community as morphing technologies with a smart wing created by a significant number of effectors [11]. The study of utilizing a larger number of effectors to promote a more adaptive wing offers a wealth of literature. The authors in [12] present a smart wing using piezoelectric actuation on both span and camber shape control in order to attenuate vibration and wake-induced buffet. A critically coupled control system is presented for an unmanned air vehicle with a total of 10 control surfaces in [29] and [30] by Holm-Hansen et al, in which rigid body and flexible modes coincide in the frequency domain, a notable design challenge. The wing of an F-18 fighter was modified in the AFRL and NASA Armstrong (formerly Dryden) Active Aeroelastic Wing technology program, demonstrating improved per-

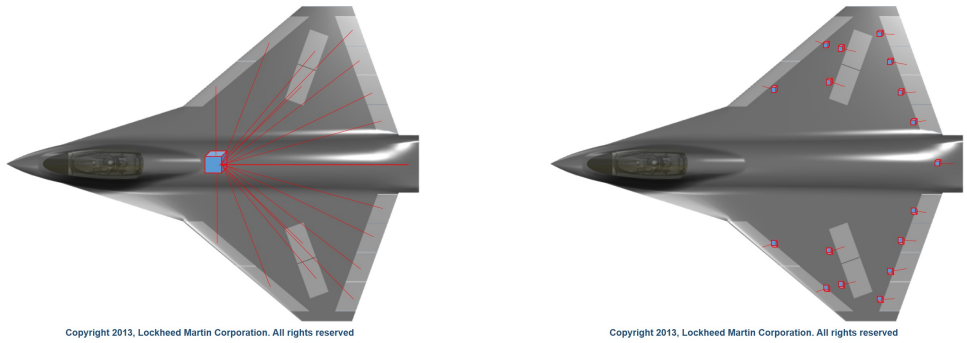
formance at transonic mate numbers. In this application, the number of effectors for the base platform was increased by partitioning the leading edge flap surface into an inboard and outboard element [31]. Bieniawski in [57] successfully demonstrates the use of Miniature-Trailing Edge Effectors (MiTEs) for increasing the flutter speed and for improved vehicle stability. Cattafesta and Sheplak in [13] survey the spectrum of common actuator technologies for active flow control with a classification summary of fluidic, piezoelectric and electroactive moving surfaces, and plasma discharging effectors. In brief, the pattern is clear. The advancement of actuation and the smart wing will increase the number of effectors beyond the conventional approach of one to two primary hinged surfaces per lifting body.

2.1.1 Challenges

Advancing the capability of the air vehicle with more actuation presents a number of opportunistic roads for engineering contribution. One challenge of modeling and simulation for such a configuration is illustrated by Obradovic in [14], where employing an increased number of effectors is required for a gull-wing aircraft that can shape to three significantly different modes. Architecturally, centralized control methods present a computational and network communication challenge as systems expand in complexity. Bandwidth requirements, processor floating point operations per second, transport lag, power demand, and the need for global synchronization are a number of factors pointing engineers to the distributed approach, where job loads can be shared across a team [35].

Figure 2.2a depicts a hypothetical vehicle with 17 effectors and a centralized flight control computer. Other than being positioned optimally with respect to the planform geometry (near the center of mass), the critical flight control computer element is strategically located just aft of the pilot station to maximize protection from

the environment. Alternatively, a hypothetical distributed architecture is depicted in Fig. 2.2b. In this case, each effector is controlled by a local module effectively increasing resiliency to failure as well as bringing the benefits of parallel processing to the control allocation of the vehicle. In essence, each effector in this approach operates a simultaneous real time process to compute the best local command.



(a) Centralized (b) Decentralized

Figure 2.2: Hypothetical Vehicle Flight Control Computer Architectures

2.2 Distributed Electric Propulsion

Recently the remote control hobbyist industry for small air vehicles has been inundated with the advancement of the Quad-rotor UAV, and the more general term, multi-rotor flying robot is now appearing in anticipation of the next wave of designs with an increased number of effectors [19]. Octo-rotors and Mesicopters are also beginning to appear in the press, and naturally, the expectation is the evolution of technology will continue to expand the envelope to the Centi-rotor or even the Milli-rotor [20, 21, 98]. The implication of distributing smaller electric engines reaches beyond toys. In a NASA Langley research program called Leading Edge Asynchronous Propellers Technology (LEAPTech) for a general aviation platform, Moore et al explore

the concept for a blown wing with the strategic placement of many small, low weight electric motors, effectively increasing the lift coefficient across the wing at low speeds [24]. The benefit is a reduction in the takeoff and landing sizing requirement for the wing, which typically drives the overall structure design. In [6], Zeune and Hooker et al present a detailed parametric study of an integrated distributed propulsion system for a next generation transport aircraft, and identify an 8% aerodynamic improvement in transonic airspeed conditions compared to the conventional beneath-the-wing large propulsive engines. Fig. 2.3 provides an illustration of the distributed propulsion concept where a number of smaller propulsive effectors are placed at the trailing edge of the wing.

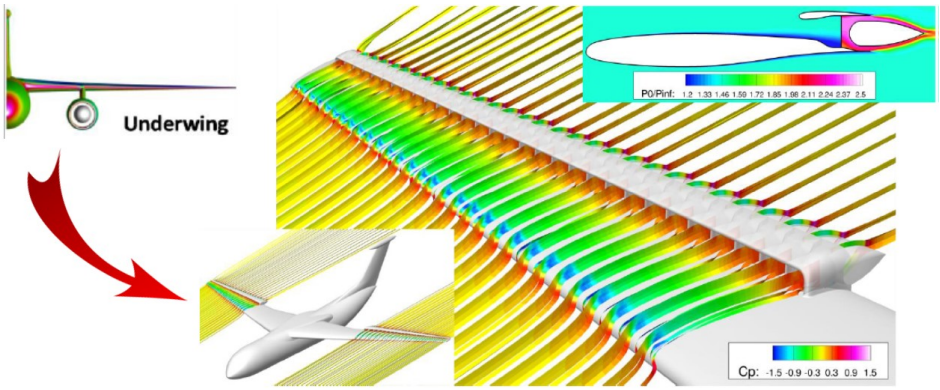


Figure 2.3: Distributed Propulsion Concepts Identified to Yield Substantial Aerodynamic Benefits by Zeune and Hooker et al in [6]

Additionally, other benefits from distributing a propulsion system have been researched from improvement to the powered lift short takeoff profile; better foreign object debris (FOD) robustness with decreased engine inlet surface area across multiple fans; decreased acoustics and noise levels; reduced fuel emissions; and an increased efficiency with in line replacement unit (LRU) maintenance costs due to the modularity of the approach [6, 23].

Significant research has been accomplished in the propulsion domain regarding these benefits for developing and integrating distributed electric approach, however there is a need for further study on the problem of controlling these distributed suites [25]. In this work, the distributed propulsion application is considered, specifically for a hovering aerospace platform. The next section presents the detailed equations of motion for modeling this type of dynamical vehicle.

2.3 Equations of Motion for Distributed Effector Suites

The equations of motion for a distributed propulsion vehicle are derived from first principles for a rigid body and presented in detail in App. A.1 with references to [93, 99, 100]. Four groups of equations are formulated to describe the dynamics and kinematics for both the translation and rotation of the vehicle and are given by Dogan et al in [93]. In the dynamics equations, the external force and moment (control input) vectors are expanded to accommodate a summation of individual effector contributions. This expansion for a distributed effector suite is presented below.

2.3.1 Translational Dynamics

The generalized Translational Dynamics equations are expressed in the vehicle body frame as

$$\dot{\mathbf{V}}_B = \mathbf{S}(\omega_B)\mathbf{V}_B + \frac{1}{m}\mathbf{f}_B \tag{2.1}$$

where $\mathbf{V}_B = [u\ v\ w]^T \in \Re^3$, $\omega_B = [p\ q\ r]^T \in \Re^3$, $\mathbf{f}_B = [f_{x_B}\ f_{y_B}\ f_{z_B}]^T \in \Re^3$, and $\mathbf{S}(\omega_B) \in \Re^{3\times 3}$ is the skew symmetric matrix of the angular velocity vector[1]. The force vector, $\mathbf{f}_B$, describes the total external input onto the system. To isolate the

[1]The skew symmetric matrix enables expressing vector cross product operations in matrix multiplication format as $\mathbf{a} \times \mathbf{b} = -\mathbf{S}(\mathbf{a})\mathbf{b}$ and is discussed in App. A.1.

control input, the inertial gravity $\mathbf{g} = [0\ 0\ g]^T$ is extracted using the rotation matrix from inertial frame to body frame to express with respect to the vehicle body frame as

$$\dot{\mathbf{V}}_B = \mathbf{R}\mathbf{g} + \mathbf{S}(\omega_B)\mathbf{V}_B + \frac{1}{m}\mathbf{F}_B \tag{2.2}$$

where

$$\mathbf{R} = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & \sin\phi\cos\theta \\ \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi & \cos\phi\cos\theta \end{bmatrix}$$

and ϕ, θ, and ψ represent the orientation of the vehicle as Euler angle roll, pitch, and yaw, respectively. The $\mathbf{F}_B$ term represents the total control input for N number of effectors[2]. Writing the individual contribution of the j^{th} effector as $\mathbf{F}j_B$, the generalized Translational Dynamics expressed in the vehicle body frame are

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \mathbf{R}\mathbf{g} + \begin{bmatrix} 0 & r & -q \\ -r & 0 & p \\ q & -p & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} + \frac{1}{m}\sum_{j=1}^{N} \mathbf{F}j_B \tag{2.3}$$

2.3.2 Rotational Dynamics

The general form of the Rotational Dynamics equations expressed in the vehicle body frame are given in [93] as

$$\dot{\omega}_B = \mathbf{I}_B^{-1}\left[\mathbf{S}(\omega_B)\mathbf{I}_B\omega_B + \tau_B\right] \tag{2.4}$$

where $\mathbf{I}_B \in \Re^{3\times3}$ denotes the vehicle mass inertia matrix, and $\tau_B = [\tau_{x_B}\ \tau_{y_B}\ \tau_{z_B}]^T \in \Re^3$ is the total external moment on the body. Due to writing this equation with

[2]In this research, the focus application is for a hovercraft in calm atmospheric conditions where the total external forces on the vehicle are gravity and thrust due to propulsive effectors. Therefore, the $\mathbf{F}_B$ expansion in this case does not include additional aerodynamics due to negligible dynamic pressure.

respect to the vehicle center of mass, the external moment due to inertial gravity is not required to be expanded in the total external moment vector. Therefore, τ_B consists of (i) the individual effector moments created by each thruster, calculated as the lever arm $\mathbf{d}_j$ with respect to the center of mass cross multiplied with the thruster force vector, $\mathbf{d}_j \times \mathbf{F}j_B$; and (ii) the inertial torque of each engine, τj_B. Writing $\mathbf{d}_j \times \mathbf{F}j_B = -\mathbf{S}(\mathbf{d}_j)\mathbf{F}j_B$ and substituting,

$$\dot{\omega}_B = \mathbf{I}_B^{-1}\left\{\mathbf{S}(\omega_B)\mathbf{I}_B\omega_B + \sum_{j=1}^{N}\left[-\mathbf{S}(\mathbf{d}_j)\mathbf{F}j_B + \tau j_B\right]\right\} \tag{2.5}$$

and expanding, the hovercraft Rotational Dynamics are governed by

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \mathbf{I}_B^{-1}\left(\begin{bmatrix} 0 & r & -q \\ -r & 0 & p \\ q & -p & 0 \end{bmatrix}\mathbf{I}_B\begin{bmatrix} p \\ q \\ r \end{bmatrix} + \sum_{j=1}^{N}\left[-\mathbf{S}(\mathbf{d}_j)\mathbf{F}j_B + \tau j_B\right]\right) \tag{2.6}$$

2.3.3 Translational Kinematics

The translational kinematics equations are written as the body frame represented velocity vector transformed by the Euler 321 rotation matrix transpose from body frame to inertial frame as

$$\mathbf{V} = \mathbf{R}^T\mathbf{V}_B \tag{2.7}$$

where $\mathbf{V} = [\dot{x}\ \dot{y}\ \dot{z}]^T \in \Re^3$ describes the velocity components of the hovercraft in the inertial fixed frame.

2.3.4 Rotational Kinematics

The rotational kinematics of the hovercraft vehicle are defined with respect to the inertial frame as

$$\dot{\mathbf{R}} = \mathbf{S}(\omega_B)\mathbf{R} \tag{2.8}$$

The left hand side of this equation is the derivative with respect to time of the Euler 321 Rotation matrix, $\dot{\mathbf{R}} \in \Re^{3\times 3}$, and yields a set of nine scalar equations which can be algebraically solved for the inertial angular velocity vector (Euler attitude rates).

$$\begin{aligned}
\dot{\phi} &= p + \tan\theta(q\sin\theta + r\cos\phi) \\
\dot{\theta} &= q\cos\phi - r\sin\phi \\
\dot{\psi} &= \frac{1}{\cos\phi}(r\cos\phi + q\sin\phi)
\end{aligned} \tag{2.9}$$

CHAPTER 3

CONTROL ALLOCATION PROBLEM

This chapter defines the control allocation problem and presents a simplified example of a linear closed loop flight control architecture referred to as dynamic inversion, where the control allocation problem is commonly encountered. Finally, the chapter concludes with a description of the control allocation problem for a nonlinear system.

3.1 Definition

The general form of a n^{th} order dynamic system to be controlled can be written as a set of n first order ordinary differential equations of the form,

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \\ \mathbf{y}(t) &= \mathbf{h}(\mathbf{x}, \mathbf{u}, t) \end{aligned} \tag{3.1}$$

where the time variant nonlinear mapping $\mathbf{f}(\mathbf{x}, \mathbf{u}, t) : \Re^{n+m+1} \mapsto \Re^n$ describes the state rate dependency on both the states $\mathbf{x}(t) \in \Re^n$ and inputs $\mathbf{u}(t) \in \Re^m$. The outputs of interest $\mathbf{y}(t) \in \Re^k$ of the system may also be a function of the states and inputs and are described by the nonlinear mapping $\mathbf{h}(\mathbf{x}, \mathbf{u}, t) : \Re^{n+m+1} \mapsto \Re^k$. A special case of this general system allows simplification where the state rate equation is decomposed into a nonlinear mapping of the states and a nonlinear mapping of the control effectivity as

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}, t) + \mathbf{G}(\mathbf{x}, \mathbf{u}, t) \\ \mathbf{y}(t) &= \mathbf{h}(\mathbf{x}, \mathbf{u}, t) \end{aligned} \tag{3.2}$$

where the nonlinear mapping $\mathbf{G}(\mathbf{x}, \mathbf{u}, t) : \Re^{n+m+1} \mapsto \Re^n$ is referred to as the control effectiveness. In this class of systems, the control effectiveness is a function of the vehicle dynamical states, $\mathbf{x} \in \Re^n$, however the pure open loop mapping $\mathbf{f}(\mathbf{x}, t) : \Re^{n+1} \mapsto \Re^n$ is now decoupled from the control input [18].

For the purpose of introducing the control allocation problem, consider a further simplified time invariant system with an output mapping without a feedforward $\mathbf{u}$ dependency written as

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x}, \mathbf{u}) \\ \mathbf{y}(t) &= \mathbf{h}(\mathbf{x}) \end{aligned} \tag{3.3}$$

with $\mathbf{f}(\mathbf{x}) : \Re^n \mapsto \Re^n$, $\mathbf{G}(\mathbf{x}, \mathbf{u}) : \Re^{n+m} \mapsto \Re^n$, and $\mathbf{h}(\mathbf{x}) : \Re^n \mapsto \Re^k$. Defining $\mathbf{g} \equiv \nabla_x \mathbf{h}\mathbf{G}(\mathbf{x}, \mathbf{u}) : \Re^{n+m} \mapsto \Re^k$ where $\nabla_x \mathbf{h} \equiv \frac{\partial \mathbf{h}}{\partial \mathbf{x}}(\mathbf{x}_o) \equiv \mathbf{C} \in \Re^{k \times n}$, the control allocation problem (CAP) can be generally stated as given k desired virtual control inputs $\mathbf{v}_d(t) \in \Re^k$, determine the required vehicle effector position $\mathbf{u}(t) \in \Re^m$ from the equation

$$\mathbf{v}_d(t) = \mathbf{g}[\mathbf{x}(t), \mathbf{u}(t)] \tag{3.4}$$

In this work, the focus is on the case where the vehicle has a significant number of effectors, yielding an underdetermined control allocation problem with significantly more true effectors than virtual control inputs $m >> k$. This is a commonly encountered case where typically many surfaces are ganged together to create a combination conducive to the desired overall effect on the vehicle. For example, in a conventional air vehicle, the left elevator and right elevators are ganged identically to create a virtual symmetric elevator input for pitching moment control. Likewise, the left and right ailerons are mirrored with a -1 gain opposition between each to create a virtual differential aileron for rolling moment control. In the conventional setting, ganging surfaces together in a technique known as effector mixing is usually intuitive. However, the evolution of technology continues to introduce complex effector suites and

air vehicle planforms, and identifying the appropriate mixing technique may not be readily apparent. Additionally, the drive for adaptive systems that reallocate control effectors after an in-flight emergency such as an actuator failure emphasizes the need for advanced methods in control allocation [15, 16, 17, 18].

The linear case of the control allocation problem is typically the focus in practice, due to the computational challenge that solving Eq. (3.4) often involves constrained nonlinear programming. Härkegård in [18] provides a detailed derivation of the linearized control allocation problem for a nonlinear system. The derivation linearizes Eq. (3.4) at a given starting control position, $\mathbf{u}_0$, and allows the state $\mathbf{x}$ to vary, creating a time variable approximation. Neglecting higher order terms, the nonlinear control effectivity mapping is expanded about $\mathbf{u}_0$ via Taylor series as [18]

$$\begin{aligned} \mathbf{g}(\mathbf{x},\mathbf{u}) &\approx \mathbf{g}(\mathbf{x},\mathbf{u}_0) + \frac{\partial \mathbf{g}}{\partial \mathbf{u}}(\mathbf{x},\mathbf{u}_0)\cdot(\mathbf{u}-\mathbf{u}_0) \\ &\approx \mathbf{g}(\mathbf{x},\mathbf{u}_0) + \mathbf{CB}(\mathbf{x})\cdot(\mathbf{u}-\mathbf{u}_0) \end{aligned} \tag{3.5}$$

where clearly $\mathbf{CB}(\mathbf{x}) \equiv \nabla_x \mathbf{h}\frac{\partial \mathbf{g}}{\partial \mathbf{u}}(\mathbf{x},\mathbf{u}_0)$. Expanding and equating to the desired virtual control input,

$$\begin{aligned} \mathbf{g}(\mathbf{x},\mathbf{u}) &\approx \mathbf{g}(\mathbf{x},\mathbf{u}_0) + \mathbf{CB}(\mathbf{x})\cdot\mathbf{u} - \mathbf{CB}(\mathbf{x})\cdot\mathbf{u}_0 \\ \mathbf{v}_d(t) &\approx \mathbf{g}(\mathbf{x},\mathbf{u}_0) + \mathbf{CB}(\mathbf{x})\cdot\mathbf{u} - \mathbf{CB}(\mathbf{x})\cdot\mathbf{u}_0 \end{aligned} \tag{3.6}$$

an approximated linear desired virtual control input with respect to a nominal control position $\mathbf{u}_0$ is defined as $\mathbf{v} = \mathbf{v}_d - \mathbf{g}(\mathbf{x},\mathbf{u}_0) + \mathbf{CB}(\mathbf{x})\mathbf{u}_0$. Substitution of $\mathbf{v}_d(t)$ from Eq. (3.6) into this definition for linear $\mathbf{v}$ yields,

$$\begin{aligned} \mathbf{v}(t) &= \mathbf{g}(\mathbf{x},\mathbf{u}_0) + \mathbf{CB}(\mathbf{x})\cdot\mathbf{u} - \mathbf{CB}(\mathbf{x})\cdot\mathbf{u}_0 - \mathbf{g}(\mathbf{x},\mathbf{u}_0) + \mathbf{CB}(\mathbf{x})\cdot\mathbf{u}_0 \\ \mathbf{v}(t) &= \mathbf{CB}(\mathbf{x})\cdot\mathbf{u} \end{aligned} \tag{3.7}$$

Härkegård in [18] defines $\mathbf{u}_0$ in a continuous time sense as the initial condition delayed by a fixed time step window of duration T. Therefore, $\mathbf{u}_0 = \mathbf{u}(t-T)$ in a discrete

control allocation process with deterministic updates. Finally, Eq. (3.4) is rewritten as the linear control allocation problem as [18]

$$\begin{aligned} \mathbf{CBu}(t) = \mathbf{v}(t) \mid \mathbf{u}_{min} \leq \mathbf{u}(t) \leq \mathbf{u}_{max} \\ \mathbf{p}_{min} \leq \dot{\mathbf{u}}(t) \leq \mathbf{p}_{max} \end{aligned} \tag{3.8}$$

with $\mathbf{u}_{min}, \mathbf{u}_{max}$ and $\mathbf{p}_{min}, \mathbf{p}_{max}$ are defined as effector position bounds and rate limits, respectively. The rate of the effector is approximated with a finite difference as $\dot{\mathbf{u}}(t) \approx \frac{[\mathbf{u}(t)-\mathbf{u}(t-T)]}{T}$ to allow combining the constraints into a compact form as,

$$\begin{aligned} \underline{\mathbf{u}} &= max\left[\mathbf{u}(t-T) + T\mathbf{p}_{min}, \mathbf{u}_{min}\right] \\ \overline{\mathbf{u}} &= min\left[\mathbf{u}(t-T) + T\mathbf{p}_{max}, \mathbf{u}_{max}\right] \end{aligned} \tag{3.9}$$

For example, Fig. 3.1 depicts when $\overline{\mathbf{u}}$ would equal the rate limited response of the effector while Fig. 3.2 depicts $\overline{\mathbf{u}}$ equal to $\mathbf{u}_{max}$, the case where the position limit is encountered.

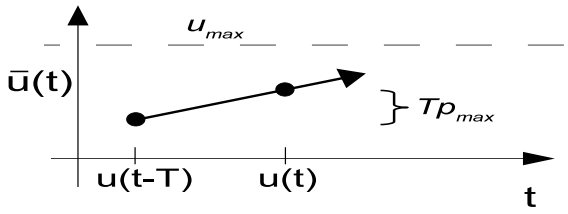


Figure 3.1: Rate Limited Constraint when $\overline{\mathbf{u}} = \mathbf{u}(t-T) + T\mathbf{p}_{max}$

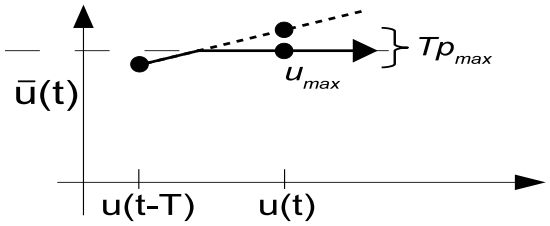


Figure 3.2: Position Limited Constraint when $\overline{\mathbf{u}} = \mathbf{u}_{max}$

The final compact form of the linear control allocation problem is then written as [18]

$$\begin{aligned} \mathbf{CBu}(t) &= \mathbf{v}(t) \\ |\ \underline{\mathbf{u}} &\leq \mathbf{u}(t) \leq \overline{\mathbf{u}} \end{aligned} \tag{3.10}$$

with $\mathbf{CB} \in \Re^{k \times m}$ with rank k. The control allocation problem is typically solved using a centralized flight control architecture. In this work, the interest is on the underdetermined setting where the number of effectors is much larger than the number of virtual control variables ($m >> k$) as detailed by Oppenheimer, Doman, and Bolender in [39] as an over-actuated system. In this case, explicit surface ganging, and the definition of pseudo-surfaces is not readily intuitive, a challenge that is becoming more apparent with complex next generation actuation.

Bodson in [16] provides a concise overview of common mathematical methods for solving the control allocation problem and categorizes into four approaches: (i) direct allocation where the goal is to determine the maximum gain[1], ρ, on $\mathbf{v}(t)$ and determine the vector $\mathbf{u}(t)$ such that $\mathbf{CBu}(t) = \rho\mathbf{v}(t)$ and $\underline{\mathbf{u}} \leq \mathbf{u}(t) \leq \overline{\mathbf{u}}$; (ii) virtual command error minimization where given $\mathbf{CB}$, determine $\mathbf{u}(t)$ such that $||(\mathbf{CBu}-\mathbf{v})||$ is minimized subject to $\underline{\mathbf{u}} \leq \mathbf{u}(t) \leq \overline{\mathbf{u}}$; (iii) control minimization where given $\mathbf{CB}$ and a desired solution $\mathbf{u}_d$, determine $\mathbf{u}(t)$ such that $||\mathbf{u} - \mathbf{u}_d||$ is minimized subject to $\mathbf{CBu}(t) = \mathbf{v}(t)$ and $\underline{\mathbf{u}} \leq \mathbf{u}(t) \leq \overline{\mathbf{u}}$; and finally, (iv) mixed minimization. In this work, the focus is on category (iv) where a mixed optimization problem is formed from both (ii) and (iii), in order to determine the best solution to minimize both the error with respect to the requested virtual command, and the magnitude of effector position required with respect to a desired position, $\mathbf{u}_d$.

[1] If the maximizing gain is determined to be $\rho > 1$, the direct control allocation method normalizes the effector solution to $\mathbf{u} \leftarrow \frac{1}{\rho}\mathbf{u}$ as described in detail in [16].

The mixed optimization problem can be defined using a variety of error norm definitions, with the most common being the Euclidean L_2 norm. Employing the L_1 norm is an active field of research and enables the use of linear-programming techniques such as the Simplex algorithm [16, 39, 45, 101]. In [44], the authors investigate the L_∞ or *sup* norm to improve robustness in the solution in the presence of effector failures and nonlinearities. In this research, the focus is on the Euclidean L_2 optimization based control allocation method. L_2 optimization control allocation appends to Eq. (3.10) a penalty for deviating from a desired control effector solution, $\mathbf{u}_d \in \Re^m$. For example, $\mathbf{u}_d$ may represent an a priori schedule of effector solutions with respect to flight conditions, and the mixed optimization solution determines the closest answer to the a priori schedule. Another case could be where a particular effector is more expensive to change with respect to a trimmed position. For instance, $\mathbf{u}_d$ could be set accordingly to penalize deviation from the current engine or throttle setting while favoring use of the surfaces on a vehicle. In this research, $\mathbf{u}_d$ is simply treated as a vector of zeros, essentially casting the optimization problem into a minimum position search. i.e. Position the effectors the least in order to achieve the desired accelerations[2].

Defining a control penalty matrix $\mathbf{W_u} \in \Re^{m \times m}$, acceleration penalty gain $\gamma \in \Re^1$, and acceleration error penalty matrix $\mathbf{W_v} \in \Re^{k \times k}$, Härkegård in [18] formally states this mixed optimization or combination as a pseudoinverse class of CAP where

$$\begin{aligned} \min_{\mathbf{u}} \; & ||\mathbf{W_u}(\mathbf{u} - \mathbf{u}_d)||_2 \\ & | \; \mathbf{CBu} = \mathbf{v} \end{aligned} \tag{3.11}$$

[2]In the hovercraft application in Ch. 6, $\mathbf{u}_d = \mathbf{0} \in \Re^m$ equates to a minimum energy solution where the engines are set to the lowest value of thrust possible to maintain the acceleration constraints. The hovercraft application also explored employing $\mathbf{u}_d$ to balance each engine to the average thrust as $\mathbf{u}_d = \left(\frac{1}{m}\sum_{j=1}^{m} u_j\right)\mathbf{1}$ with $\mathbf{1} \in \Re^m$.

yields an analytic solution when actuator constraints are not encountered. For the unconstrained optimization problem, without consideration of effector limits, a global cost is formed as the 2-norm of the desired (preferred) control effector solution and the virtual control error as

$$L_2 = ||\mathbf{W_u}(\mathbf{u} - \mathbf{u}_d)||_2 + \gamma||\mathbf{W_v}(\mathbf{CBu} - \mathbf{v})||_2 \tag{3.12}$$

Finally, when effector constraints are accounted, an unconstrained optimization problem is formed by augmenting the L_2 objective with an additional penalty, $\lambda \in \Re^1$, on position and rate limit violations,

$$L_{2C} = ||\mathbf{W_u}(\mathbf{u} - \mathbf{u}_d)||_2 + \gamma||\mathbf{W_v}(\mathbf{CBu} - \mathbf{v})||_2 + \lambda||\mathbf{u} - \underline{\mathbf{u}}||_2 + \lambda||\mathbf{u} - \overline{\mathbf{u}}||_2 \tag{3.13}$$

For cases where the control allocation solution is not bounded by effector limits ($\underline{\mathbf{u}} \leq \mathbf{u}(t) \leq \overline{\mathbf{u}}$), the λ penalty terms are set to zero, and Eq. (3.13) and Eq. (3.12) yield identical costs.

3.2 Example Closed Loop Application

The solution to Eq. (3.13) yields the optimal, achievable commanded values for the vehicle effector position $\mathbf{u}(t)$ in order to accomplish the demands of the desired virtual control inputs $\mathbf{v}_d(t)$. This functionality is a primary element in an inner loop control problem for an aerospace vehicle. For demonstration, a simplified input-output feedback linearization system, depicted as a dynamic inversion controller in Fig. 3.3 is provided.

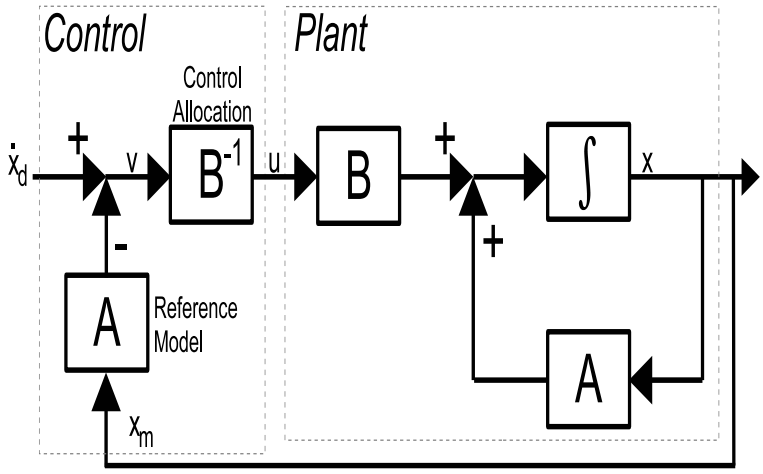


Figure 3.3: Illustration of Simplified Dynamic Inversion Control Scheme with Control Allocation and On Board Reference Model

This example system has been vastly simplified for the purpose of brevity to illustrate the placement of the CAP functionality into a closed loop system. Many alternative closed loop control methodologies exist with the control allocation centerpiece providing the real time mixing to the individual effectors. Here, this architecture is chosen simply to depict an example setting with the control allocator in the loop.

Consider a linear state space system described as

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{3.14}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} \tag{3.15}$$

with system state vector $\mathbf{x} \in \Re^n$, control input vector $\mathbf{u} \in \Re^m$, and output vector $\mathbf{y} \in \Re^k$. The dynamic inversion methodology requires relating the control variable of interest in the output $\mathbf{y}$ to the input vector $\mathbf{u}$ [15, 102]. To accomplish this relation,

the output Eq. (3.15) is differentiated until the control input vector $\mathbf{u}$ appears in the expression.

$$\mathbf{y} = \mathbf{C}\mathbf{x} \tag{3.16}$$

$$\dot{\mathbf{y}} = \mathbf{C}\dot{\mathbf{x}} \tag{3.17}$$

$$\dot{\mathbf{y}} = \mathbf{C}(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}) \tag{3.18}$$

In this simple example, one differentiation is required before the input vector $\mathbf{u}$ appears in Eq. (3.18) and therefore the dynamic inversion approach is of the first relative degree [15]. The next step is to solve for the control input vector $\mathbf{u}$,

$$\dot{\mathbf{y}} = \mathbf{C}\mathbf{A}\mathbf{x} + \mathbf{C}\mathbf{B}\mathbf{u} \tag{3.19}$$

$$\mathbf{C}\mathbf{B}\mathbf{u} = \dot{\mathbf{y}} - \mathbf{C}\mathbf{A}\mathbf{x} \tag{3.20}$$

$$\mathbf{u} = (\mathbf{C}\mathbf{B})^{-1}(\dot{\mathbf{y}} - \mathbf{C}\mathbf{A}\mathbf{x}) \tag{3.21}$$

The $\dot{\mathbf{y}}$ term in Eq. (3.21) is the specified output dynamics or desired dynamics of the system and provides a command hook for the inner loop controller, while the $\mathbf{C}\mathbf{A}\mathbf{x}$ term is typically provided by an on-board model estimation (denoted as estimated linear state space matrices $\hat{\mathbf{A}}$, $\hat{\mathbf{B}}$, and $\hat{\mathbf{C}}$). Rewriting to reflect these descriptions as the desired output of the system $\dot{\mathbf{y}}_d$ and the predicted response of the vehicle $\hat{\mathbf{C}}\hat{\mathbf{A}}\mathbf{x}_m$ with $\mathbf{x}_m$ as the measured states of the vehicle, the dynamic inversion control law is

$$\mathbf{u} = (\hat{\mathbf{C}}\hat{\mathbf{B}})^{-1}(\dot{\mathbf{y}}_d - \hat{\mathbf{C}}\hat{\mathbf{A}}\mathbf{x}_m) \tag{3.22}$$

Finally, substituting the control law back into the linear system in Eq. (3.18),

$$\dot{\mathbf{y}} = \mathbf{C}\mathbf{A}\mathbf{x} + \mathbf{C}\mathbf{B}\left[(\hat{\mathbf{C}}\hat{\mathbf{B}})^{-1}(\dot{\mathbf{y}}_d - \hat{\mathbf{C}}\hat{\mathbf{A}}\mathbf{x}_m)\right] \tag{3.23}$$

Clearly, this equation illustrates the importance of accurate on board modeling in the control law, or residual dynamics occur. This issue is beyond the scope of this

research and the assumption is $\hat{\mathbf{A}} = \mathbf{A}$, $\hat{\mathbf{B}} = \mathbf{B}$, $\hat{\mathbf{C}} = \mathbf{C}$, and $\mathbf{x} = \mathbf{x}_m$. The closed loop equation then simplifies to

$$\dot{\mathbf{y}} \;=\; \dot{\mathbf{y}}_d \tag{3.24}$$

Therefore, the dynamics of the chosen output can be set as desired given this approach. Fig. 3.3 further simplifies this example with perfect state feedback, where $\mathbf{C} = \mathbf{1}^{k\times n}$ and ones populate the state elements chosen as virtual control variables[3]. The control allocation is depicted as the $\mathbf{B}^{-1}$ element mapping the linear commands into the control vector $\mathbf{u}$.

Many important details remain in successfully building the dynamic inversion control scheme outside of the control allocation problem. In brief, the relative order of the system must be considered in order to verify internal dynamics stability. Residual dynamics due to error between the truth and the on board models are important factors as well as modeling errors that impact the control allocation problem where a valid solution is not obtained. Additionally, open loop systems which exhibit non-minimum phase (NMP) characteristics pose a challenge in the dynamic inversion approach. Lee et al in [103] present decoupling strategies to improve the NMP dynamics. This issue becomes prevalent for highly flexible structures and for hypersonic flight conditions [104, 105].

3.3 Nonlinear Control Allocation

While the linear control allocation problem is most commonly studied in literature, there have been notable efforts solving the full nonlinear form [16, 41, 42, 43]. A sequential quadratic programming technique was constructed in [41] by Johansen to avoid effector constraints as well as inversion singularities for a marine vessel. Jo-

[3]In the special case of full and perfect state feedback, $k = n$ and $\mathbf{C} = \mathbf{I}^{n\times n}$

hansen also in [42] develops a Lyapunov dynamic update law to solve the nonlinear CAP for over-actuated mechanical systems. In [43], Bolender and Doman use piecewise linear functions to model the nonlinear control effector moment data and solve the nonlinear CAP with a mixed integer linear program.

In general, the nonlinear control allocation problem can be solved again as a mixed L_2 optimization problem using the full nonlinear control effectivity matrix of the system,

$$L_{2C} = ||\mathbf{W_u}(\mathbf{u} - \mathbf{u}_d)||_2 + \gamma||\mathbf{W_v}[\mathbf{g}(\mathbf{x}, \mathbf{u}) - \mathbf{v}_d]||_2 + \lambda||\mathbf{u} - \underline{\mathbf{u}}||_2 + \lambda||\mathbf{u} - \overline{\mathbf{u}}||_2 \quad (3.25)$$

where the governing nonlinear (possibly time variant) state equation is given in Eq. (3.2). In nonlinear dynamic inversion, which typically employs nonlinear control allocation along with a nonlinear reference model, the control problem must be cast as one affine in the inputs, as described by Subbarao et al in [106]. Wedershoven in [107] presents a transformation procedure to accomplish this step, however this work investigates solving the non-affine form in Eqs. (3.2, 3.3, 3.25) directly using distributed optimization.

CHAPTER 4

PROBABILITY COLLECTIVES

This chapter outlines the probability collectives optimization method in detail and presents two forms: (i) delayed sampling PC and (ii) immediate sampling PC. After the description of each form, a simplified two dimensional optimization example is demonstrated. Finally, the chapter concludes with the presentation of the distributed control allocation optimization algorithm based on delayed sampling PC and implemented in a MATLAB Simulink® toolbox.

4.1 The Distributed Optimization Method of Probability Collectives

Transformation of the optimization problem, $argmin_{\mathbf{x}} G(\mathbf{x})$ from the variable domain to the probability domain, over a sample region $\mathbf{D}$, requires use of the expectation operation,

$$E_{\mathbf{p}}(G) \equiv \int_{\mathbf{D}} \mathbf{p}(\mathbf{x}) G(\mathbf{x}) d\mathbf{x} \tag{4.1}$$

casting the goal into searching for the optimal probability distribution, $\mathbf{p}(\mathbf{x})$, to minimize the expected value of the original objective, $argmin_{\mathbf{p}} E_{\mathbf{p}}[G(\mathbf{x})]$ [108]. The continuous domain Lebesgue integral is interchangeable with a discrete domain point sum over the sample region $\mathbf{D}$ as described in [81]. Substitution of the expectation operation from Eq. (4.1) yields the problem, $argmin_{\mathbf{p}} \int_{\mathbf{D}} \mathbf{p}(\mathbf{x}) G(\mathbf{x}) d\mathbf{x}$. Denote the true joint strategy of a multi-agent system as $\mathbf{p}(\mathbf{x})$ with n_{agent} as the total number of agents and n_{moves_i} as the number of potential actions or moves that each i^{th} agent can choose from a pure set, $\mathbf{x_i} \in \Re^{n_{moves_i}}$. In general, the agents are not independent and the true strategy is coupled as $\mathbf{p}(\mathbf{x}) = \mathbf{p}(\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_{n_{agent}}})$. In PC, a product

distribution, $\mathbf{q}(\mathbf{x}) = \mathbf{q_1}(\mathbf{x_1})\mathbf{q_2}(\mathbf{x_2})...\mathbf{q_{n_{agent}}}(\mathbf{x_{n_{agent}}})$ is used to approximate this true strategy, $\mathbf{q}(\mathbf{x}) \approx \mathbf{p}(\mathbf{x})$, but this assumption is not made crudely without consideration. Provisions are implemented to effectively drive the approximation to the truth, by using the Kullback-Leibler distance measure between two probability distributions, to be described further in detail ahead. Substituting the approximating joint product distribution for the true strategy into the expectation operation yields the following constrained optimization problem,

$$\begin{aligned} \min_{\mathbf{q}} E_{\mathbf{q}}(G) = \int_{\mathbf{D}} d\mathbf{x} G(\mathbf{x}) \prod_{i=1}^{n_{agent}} \mathbf{q_i}(\mathbf{x_i}) \\ | \; S(\mathbf{q}) = s \\ \int_{\mathbf{D}} \mathbf{q_i}(\mathbf{x_i}) d\mathbf{x_i} = 1, \; \forall i \\ \mathbf{q_i}(\mathbf{x_i}) \geq 0, \; \forall \mathbf{x_i} \end{aligned} \tag{4.2}$$

where the constraint to maintain $S(\mathbf{q})$ at a value of s is a key step in forming the maximum-entropy (maxent) problem. $S(\mathbf{q})$ is the Shannon entropy of the probability distribution, essentially quantifying the amount of syntactic information provided by the distribution. The remaining constraints in the maxent, Eq. (4.2), enforce the solution $\mathbf{q}$ to remain in the valid probability domain with the area under the curve always equal to 1 as a unit simplex with non-negative values [57, 60]. The Shannon entropy of a probability distribution, $\mathbf{q}(\mathbf{x})$, is defined as

$$S(\mathbf{q}) = -\int_{\mathbf{D}} \mathbf{q}(\mathbf{x}) ln \left[\frac{\mathbf{q}(\mathbf{x})}{\mu(\mathbf{x})} \right] d\mathbf{x} \tag{4.3}$$

where $\mu(\mathbf{x})$ is typically designated as a uniform distribution of $\mathbf{x}$, but as a constant value in the integrand over a finite sample region $\mathbf{D}$, is often omitted. Typically, this omission is considered as the general form for differential entropy of a continuous random variable [109]. Eq. (4.3) represents a scalar quantification of the amount of

given uncertainty, or lack of syntactic information provided by the probability distribution, where the measure of syntactic content is proportional to negative $S(\mathbf{q})$.[1] The syntactic information content in this case is measured in natural units (nats) due to the chosen form of the entropy equation with a logarithmic base e, as described by Shannon in [111, 112]. Forming the unconstrained optimization problem, the Maxent Lagrangian represents the adjoined objective function and entropy with T representing a temperature penalty term analogous to simulated annealing optimization and a Lagrangian multiplier. The augmented entropy constraint with s represents a given level of uncertainty [63, 66].

$$L(\mathbf{q}, T) = E_{\mathbf{q}}(G) - T\left[S(\mathbf{q}) - s\right] \tag{4.4}$$

The Maxent Lagrangian optimization problem is to minimize the expected value of the objective $G(\mathbf{x})$ by determining the optimal joint mixed strategy, $\mathbf{q}(\mathbf{x})$, such that the negative Shannon entropy, $-S(\mathbf{q})$, is minimized (uncertainty is maximized). Intuitively, one would expect the Lagrangian to be formed with $+S(\mathbf{q})$ to minimize uncertainty, however this forms a biased assignment. Dating to the original work of Bernoulli, Laplace, and Poincaré, maximizing entropy and the Wolpert mathematical framework extends the Principle of Insufficient Reason with the realization that in order to form a valid statistical inference on partial information, the probability distribution must represent only the known measurements while fully accounting for uncertainty. Jaynes in [110] provides a detailed explanation for this theory as well as insight into the deep connections behind PC with statistical mechanics and ex-

[1] Negative Shannon entropy as positive syntactic content promotes an insightful discussion on the extremum optimization problem where considering the quantities as interchangeable leverages the idea that maximizing a function is identical to minimizing -1 times the same function. e.g. Minimizing negative Shannon entropy $\implies$ maximizing Shannon entropy $\implies$ minimizing information beyond that which is already measured [110].

perimental physics to explain macroscopic quantities[2]. In general, the solution to Eq. (4.4) is non-trivial to determine and the true joint strategy, $\mathbf{p}$, of the collective is of a set of coupled Boltzmann equations in the canonical ensemble of the form,[3,4]

$$\mathbf{p}^{\beta}(\mathbf{x}) \propto exp^{-\beta G(\mathbf{x})} \tag{4.5}$$

where $\beta = 1/T$. The PC method solves Eq. (4.4) in an iterative, distributed manner where each individual agent optimizes a local Maxent Lagrangian. Let $G[\mathbf{x_i}, \mathbf{x}^{-}_{(\mathbf{i})}]$ denote the i^{th} agent's view of the global objective, as the value of G that results from a local action $\mathbf{x_i}$, given the prior actions (measured or estimated) of all other agents, $\mathbf{x}^{-}_{(\mathbf{i})}$. The local Maxent Lagrangian for agent i is written as [57, 60]

$$L_i(\mathbf{q_i}) = E_{\mathbf{q}}\{G[\mathbf{x_i}, \mathbf{x}^{-}_{(\mathbf{i})}]\} - T[S(\mathbf{q_i})] \tag{4.6}$$

Expanding to the discrete form to reflect the approaches constructed by Bieniawski in [57, 60] and implemented in this work,

$$\begin{aligned} L_i(\mathbf{q_i}) = \sum_{j=1}^{n_{moves_i}} q_i[x_i(j)] E_{\mathbf{q_{(i)}}} \left\{ G[x_i(j), \mathbf{x}^{-}_{(\mathbf{i})}] | x_i(j) \right\} - \\ T \left(- \sum_{j=1}^{n_{moves_i}} q[x_i(j)] ln\{q[x_i(j)]\} \right) \end{aligned} \tag{4.7}$$

In the distributed optimization scheme, each agent in the total system optimizes the local Maxent Lagrangian in Eq. (4.7) while enforcing a valid probability domain on

[2]Jaynes in [110] also reveals thermodynamic entropy is identical to the information Shannon entropy discussed herein with the exception of a Boltzmann's constant when describing units of temperature.

[3]See discussion beneath Eq. (4.1) regarding true strategy $\mathbf{p}$ vice the approximated strategy $\mathbf{q}$ (a product distribution) and in [57, 60, 62, 63]. Here, naturally, the true canonical ensemble solution given in Eq. (4.5) is for the true Maxent Lagrangian with $\mathbf{p}$ substituted for $\mathbf{q}$ in Eq. (4.4).

[4]Jaynes also refers to the general solution to the Maxent Lagrangian in Eq. (4.5) as the quantum-mechanical grand canonical ensemble [110].

the updated mixed strategy, $\mathbf{q_i}(\mathbf{x_i})$. These constraints to maintain a unit simplex and non-negative distribution are not augmented to the cost of the Lagrangian and are performed as in-line updates in the algorithm. The local discrete optimization problem for each agent is then written in final form as [57, 60]

$$\min_{\mathbf{q_i}} L_i(\mathbf{q_i}) \mid \sum_{j=1}^{n_{moves_i}} q_i[x_i(j)] = 1, q_i[x_i(j)] \geq 0, \forall j \tag{4.8}$$

4.2 Delayed Sampling PC

The canonical ensemble solution to the true Maxent Lagrangian in Eq. (4.4) is generally not a product distribution and is a full joint Boltzmann distribution as described by Wolpert and Bieniawski in [59]. In Delayed Sampling PC (DSPC), the Kullback-Leibler distance, $KL(\mathbf{p}||\mathbf{q})$, is employed to iteratively drive a product distribution approximation $\mathbf{q}$ to the truth $\mathbf{p}$ in a second order simplex-constrained gradient descent (SOSCGD) method described as Nearest Newton. The KL distance measure is a combination of the Shannon cross-entropy between two probability distributions, $S(\mathbf{p}||\mathbf{q})$, and the Shannon entropy of the target $S(\mathbf{p})$, and is positive definite and minimum if and only if the two distributions are identical[5]. The scalar value resulting from the measure describes the "infinite limit log likelihood of data being generated by one distribution but misattributed to have come from the other" and is given as [59]

$$\begin{aligned} KL(\mathbf{p}||\mathbf{q}) =& S(\mathbf{p}||\mathbf{q}) - S(\mathbf{p}) \\ =& -\sum_{i=1}^{n_{agent}} \int_{\mathbf{D}} \mathbf{p}(\mathbf{x}) ln\left[\mathbf{q_i}(\mathbf{x_i})\right] d\mathbf{x} \end{aligned} \tag{4.9}$$

[5]The Shannon cross-entropy is $S(\mathbf{p}||\mathbf{q}) = -\int_{\mathbf{D}} \mathbf{p}(\mathbf{x}) ln\left[\frac{\mathbf{q}(\mathbf{x})}{\mu(\mathbf{x})}\right] d\mathbf{x}$ and is also the general form for the Shannon entropy of a single PDF as $S(\mathbf{q}) = S(\mathbf{q}||\mathbf{q})$. Typically the notation considering a single PDF is abbreviated as shown in Eq. (4.3).

The KL distance is not symmetrical in that $KL(\mathbf{p}||\mathbf{q}) \neq KL(\mathbf{q}||\mathbf{p})$ as demonstrated below in Fig. 4.1 with two discrete Gaussian probability distributions.

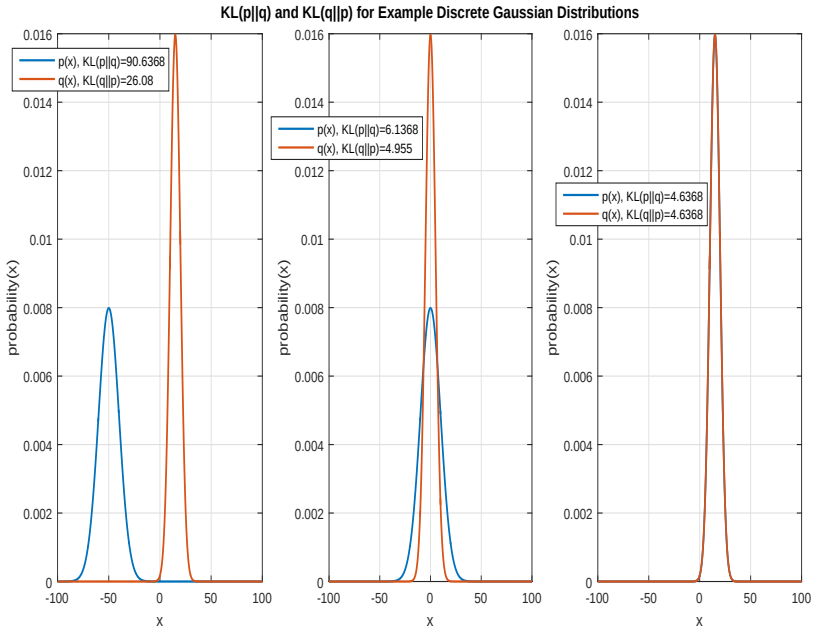


Figure 4.1: Example Kullback-Leibler Distance Measures of $KL(\mathbf{p}||\mathbf{q})$ and $KL(\mathbf{q}||\mathbf{p})$ Between Probability Distributions $\mathbf{p}(\mathbf{x})$ and $\mathbf{q}(\mathbf{x})$

A detailed discussion comparing both metrics, $KL(\mathbf{p}||\mathbf{q})$ and $KL(\mathbf{q}||\mathbf{p})$, for use in two second order descent schemes to optimize the true Maxent Lagrangian is given in [59]. In summary, one method considered is to approximate the canonical ensemble solution in Eq. (4.5) by recursively minimizing the qp-distance, $KL(\mathbf{q}||\mathbf{p})$, which effectively yields an approximating approach on both the distribution, $\mathbf{q}$, and the Lagrangian, $L(\mathbf{q})$. The gradient descent update refining the approximate $\mathbf{q}$ traverses an approximation to the true Lagrangian. Wolpert et al conclude a second method is superior, and refer to the approach as Nearest Newton descent where, instead, an analytic form of SOSCGD on a quadratic approximation to $L(\mathbf{p})$ is found. Then, using

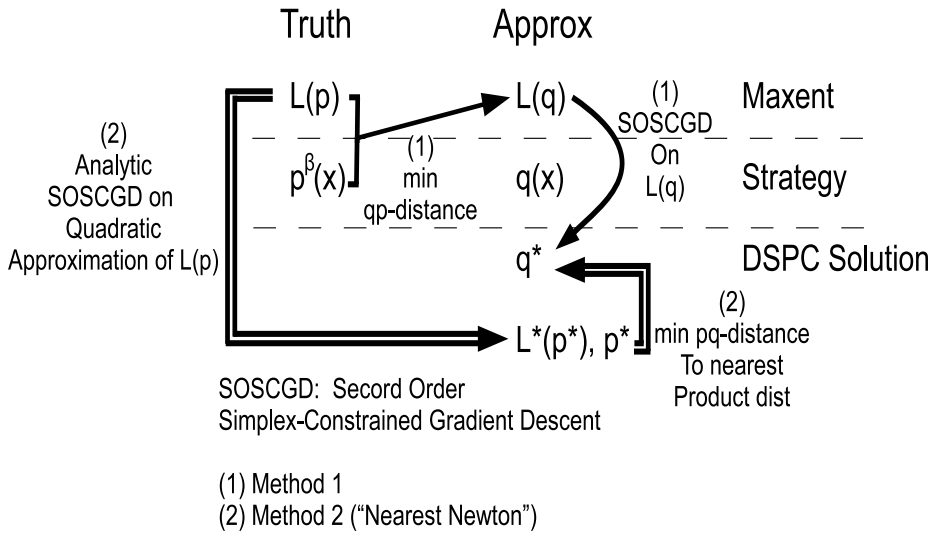


Figure 4.2: Two Methods to Determining the Optimal Mixed Strategy, $\mathbf{q}^*$, where Method (2) is Denoted as the Nearest Newton Method in Delayed Sampling Probability Collectives as Derived by Wolpert et al

the pq-distance, $KL(\mathbf{p}||\mathbf{q})$ in Eq. (4.9), an approximation to the analytic solution is enabled by substituting the minimizing solution to the pq-distance as $\mathbf{q_i} = \mathbf{p_i}, \forall i$. These two methods are depicted in Fig. 4.2. In detail, Nearest Newton (NN) makes use of the analytic form of the SOSCGD scheme on $L(\mathbf{p})$, defined as [59]

$$\mathbf{p}^*(\mathbf{x}) = \mathbf{p^0}(\mathbf{x}) \left\{1 - S(\mathbf{p^0}) - ln[\mathbf{p^0}(\mathbf{x})] - \beta[G(\mathbf{x}) - E(G)]\right\} \tag{4.10}$$

where $\mathbf{p}^*(\mathbf{x})$ denotes the updated probability distribution on a quadratic approximation to $L(\mathbf{p})$. Next, in order to determine a product distribution, $\mathbf{q}$, that is closest to $\mathbf{p}^*$, the minimizing solution, $\mathbf{q_i} = \mathbf{p_i}, \forall i$, to the pq-distance in Eq. (4.9) is substituted yielding the NN method as defined by Wolpert et al [59].

$$\mathbf{q_i}(\mathbf{x_i}) \leftarrow \mathbf{q_i}(\mathbf{x_i}) - \alpha \mathbf{q_i}(\mathbf{x_i}) \times \left\{ S(\mathbf{q_i}) + \ln[\mathbf{q_i}(\mathbf{x_i})] + \frac{E(G|\mathbf{x_i}) - E(G)}{T} \right\} \tag{4.11}$$

Eq. (4.11) is rewritten as an update law with a scalar gain, α, as a step size in order to prevent the iteration to the probability distribution from occurring too rapidly. Bieniawski in [57] discusses an important note in the NN method in that α should be equivalent for all agents[6]. Additionally, after each update to $\mathbf{q}$, the DSPC algorithm renormalizes the PDF (probability distribution function) and ensures a valid empirical distribution is maintained [60]. Given the scope of the local agent, the expected value of the global objective is the next hurdle in the NN update law. In DSPC, these expectations are approximated by Bieniawski et al in [60] with two "private utilities" as either Team Game (TG) or Wonderful Life Utility (WLU) functions. In this work, the TG private utility has been elected where the global objective (i.e. world utility) is estimated at the local scope as[7]

$$G[\mathbf{x_i}, \mathbf{x_{(i)}}] \approx g_{TG}[\mathbf{x_i}, \mathbf{x^-_{(i)}}] \tag{4.12}$$

The expected value of impact (cost) on the global objective for the i^{th} agent's new action, $\mathbf{x_i}$, is estimated by evaluating the global objective given the sampled prior actions of the other agents, $\mathbf{x^-_{(i)}}$. Also, at this stage of development, an assumption is drawn that an analytic form of the optimization problem G is known and this information is available at the local agent level, and therefore, g_{TG} is evaluated by each agent. Additionally, a network topology is in place in that each agent receives a single sampled value of the other agents' prior actions, $\mathbf{x^-_{(i)}}$. Future research on

[6]Additionally, Bieniawski concludes that method (1) and method (2) depicted in Fig. 4.2 result in the equivalent update law for NN, however the method (2) derivation emphasizes the need for all agents to employ an equivalent step size, α.

[7]The Wonderful Life Utility (WLU) was not investigated for distributed control allocation in this work and uses the global objective minus the global objective with a fixed argument as the lowest probability action for the local agent, $g_{WLU}[\mathbf{x_i}, \mathbf{x^-_{(i)}}] \approx G[\mathbf{x_i}, \mathbf{x_{(i)}}] - G[\min_{\mathbf{x_i}} \mathbf{q_i}(\mathbf{x_i}), \mathbf{x_{(i)}}]$ and is described to yield lower variance in [57].

this assumption may relax this constraint further where only local measurements are conducted on the system G (or on an estimate of the system), to create a truly decentralized, distributed architecture. In this initial study, distributed optimization is in place with DSPC, however a centralized network router, or bus is required for minimal communication between all agents.

With the TG private utility in Eq. (4.10), the NN update law is performed on the probability value for each j^{th} action of the total n_{moves_i} available in the mixed strategy $q_i[x_i(j)]$ using a data aging technique for m samples. A weighted average of costs for the global objective is computed over the k of n_{step} total iterations of the DSPC routine where γ is a data aging coefficient [57].

$$\begin{aligned} E[g_i|x_i(j)] &= \frac{N_{ij}^{(k)}}{D_{ij}^{(k)}} \\ &= \frac{\sum_m g_i[x_i(j), \mathbf{x}_{(\mathbf{i})}^-]\mathbf{1}[x_i(j)] + \gamma N_{ij}^{(k-1)}}{\sum_m \mathbf{1}[x_i(j)] + \gamma D_{ij}^{(k-1)}} \end{aligned} \tag{4.13}$$

where

$$\mathbf{1}[x_i(j)] = \begin{cases} 1 \text{ when } \mathbf{x_i} = j, \\ 0 \text{ otherwise} \end{cases}$$

Combining the data aging technique and agent private utility, the final NN update law for the j^{th} probability bin, where the bin represents the probability of the i^{th} agent's strategy selecting the j^{th} action, is given as [57, 59, 60]

$$\begin{aligned} q_i[x_i(j)] \leftarrow q_i[x_i(j)] - \\ \alpha q_i[x_i(j)] \times \Big(S(\mathbf{q_i}) + \ln\{q_i[x_i(j)]\} + \\ \frac{1}{T}[E\{g_{TG}[\mathbf{x_i}, \mathbf{x}_{(\mathbf{i})}^-]|\mathbf{x_i}\} - E\{g_{TG}[\mathbf{x_i}^-, \mathbf{x}_{(\mathbf{i})}^-]\}] \Big) \end{aligned} \tag{4.14}$$

Note the $E\{g_{TG}[\mathbf{x_i}^-, \mathbf{x}_{(\mathbf{i})}^-]\}$ term represents the expected value of the objective in the case where agent i does not make a change in strategy and maintains the last action, $\mathbf{x_i}^-$.

DSPC also includes provisions for constraints in the optimization problem by a traditional augmentation of the global objective with a Lagrange multiplier [59]. Given a constraint function on the i^{th} agent, $\mathbf{c_i}(\mathbf{x_i})$, and a Lagrange multiplier, λ_i, the calculation in Eq. (4.12) is summed with $\lambda_i \mathbf{c_i}(\mathbf{x_i})$. In the case where constraints are satisfied, which in this work represents when effectors are not saturating on position or rate limits as detailed in Ch. 5 and Ch. 6, $\mathbf{c_i}(\mathbf{x_i})$ is simply zero and does not impact the global cost. [59] also provides an update rule for λ_i with step size η for constrained DSPC as

$$\lambda_i \leftarrow \lambda_i + \eta E[\mathbf{c_i}(\mathbf{x_i})] \tag{4.15}$$

DSPC is summarized below in Alg. (1) with additional details available in [59] by Bieniawski et al.

Algorithm 1 Delayed Sampling Probability Collectives

procedure DSPC

init:

Initialize PC Parameters (T, α, γ, η)

Initialize Stopping Tolerances or Maximum Steps

Initialize Agent Strategy to Uniform Distributions

loop until stop tolerance or maximum steps reached:

Draw m Samples from each agent strategy

Evaluate private utility Eq. (4.12) + constraint costs

Perform data averaging in Eq. (4.13)

SOSCGD NN Update on PDF Strategy Eq. (4.14)

Update Lagrange Multipliers per Eq. (4.15)

Constrain PDF to Valid Domain

exit:

Draw Final Sample and Depict PDF

4.2.1 Optimization Example

Consider a simple two dimensional quadric optimization problem,

$$\min_{\mathbf{x},\mathbf{y}} G(\mathbf{x},\mathbf{y}) = (\mathbf{x}-50)^2 + (\mathbf{y}+50)^2 \tag{4.16}$$

Clearly, the minimizing argument, x^* and y^* is equal to 50 and -50, respectively. DSPC is employed with $T = 0.1$, $\alpha = 0.01$, $\gamma = 0.02$, and with two agents, $n_{agent} = 2$, with $\mathbf{x_1} \equiv \mathbf{x} \in \Re^{n_{moves_i}} \in [-100, 100]$, $\mathbf{x_2} \equiv \mathbf{y} \in \Re^{n_{moves_i}} \in [-100, 100]$ where $n_{moves_i} = 5000$. The stopping condition is set to a maximum step size of 10 iterations to optimize the agent mixed strategy, $\mathbf{q_i}(\mathbf{x_i}) \in \Re^{n_{moves_i}}$. 100 samples ($m = 100$) are drawn from the current strategy per iteration. Starting with the initial condition in Fig. 4.3, the DSPC iterations for $k = 1, 3$ are depicted below in Figs. 4.4,4.5 to illus-

trate the update process. As each sample is drawn, the strategy is evolved by binning the sample to the corresponding agent action $x_i(j)$ on the PDF independent axis. Each bin height then updates with the Nearest Newton law in Eq. (4.14) essentially refining the probability of the action as a function of the impact on the system cost. Note the decreasing value of the Shannon Entropy (SE) at each update as given on the vertical axis of the figure. The conclusion of this process at $k = 10$ is depicted in Fig. 4.6 with the DSPC final converged solution, where the top row of subplots from left to right are the final PDF and CDF (cumulative distribution function), respectively. The CDF is computed after each PDF update to facilitate the inverse transform method employed for drawing samples from the empirical agent strategy [113]. The bottom row of subplots in Fig. 4.6 show on the left the decreasing value of the objective $G(x_1^*, x_2^*)$ as a function of DSPC iteration, k, and x_1^* and x_2^* on the right as the most recent sampled values from the optimized strategy. Additionally, the two dimensional trajectory of x_1^* and x_2^* as a function of k is provided on a truth contour plot in Fig. 4.7. The contours are generated on the exact values of $G(\mathbf{x}, \mathbf{y})$ over the specified domain of interest.

Due to the stochastic nature of PC with transformations to the probability domain and inversions via random drawn samples, multiple tests were then performed to characterize performance on the simple two dimensional example. Fig. 4.8 presents the summary of DSPC performance for 100 tests with the left and right subplots depicting the final x_1^* and x_2^* values, respectively, as a function of test number. While the means of the converged solutions are nearly identical to the truth, the dispersion presented here is regarded as the best observed after a series of trial and error parameter optimizations for the given values of $T = 0.1$, $\alpha = 0.01$, $m = 100$, and $\gamma = 0.02$. These variances are reflected by the 1σ bounds and were achieved in this multiple

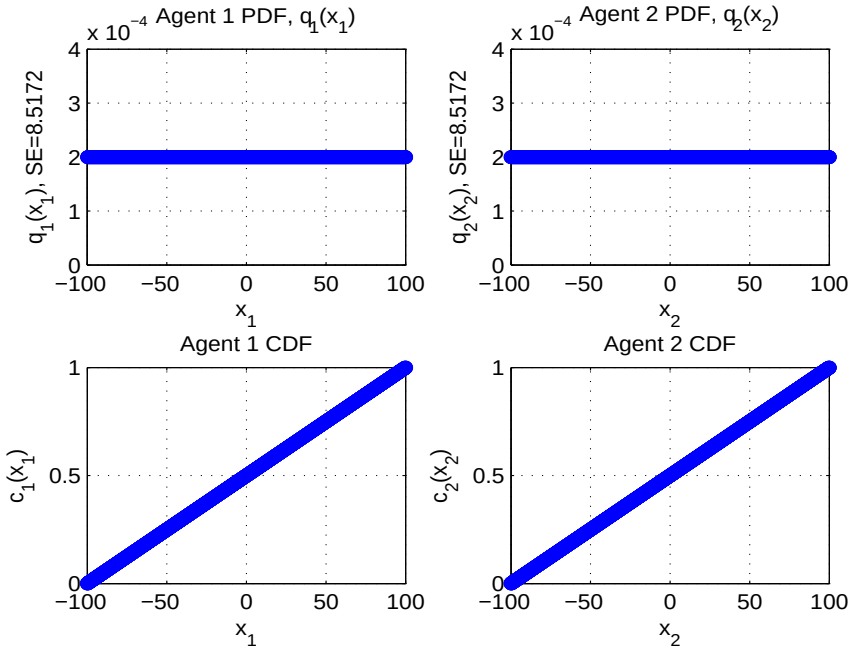


Figure 4.3: DSPC Initial Condition $k = 0$, Uniform Distribution Given Search Bounds for $T = 0.1$, $\alpha = 0.01$, $m = 100$, and $\gamma = 0.02$

test sweep with these identical parameter values with a deterministic 10 maximum step stopping criteria enabled to promote comparison [60, 57][8,9].

[8]One method currently explored to improve the variance of the minimizing argument is for the agent to compute and report the mean of the last round of drawn samples, rather than a single sample from the current strategy.

[9]Alternatively, using the expected value of the final strategy, $\mathbf{q}(\mathbf{x})$, is in research for further improvement.

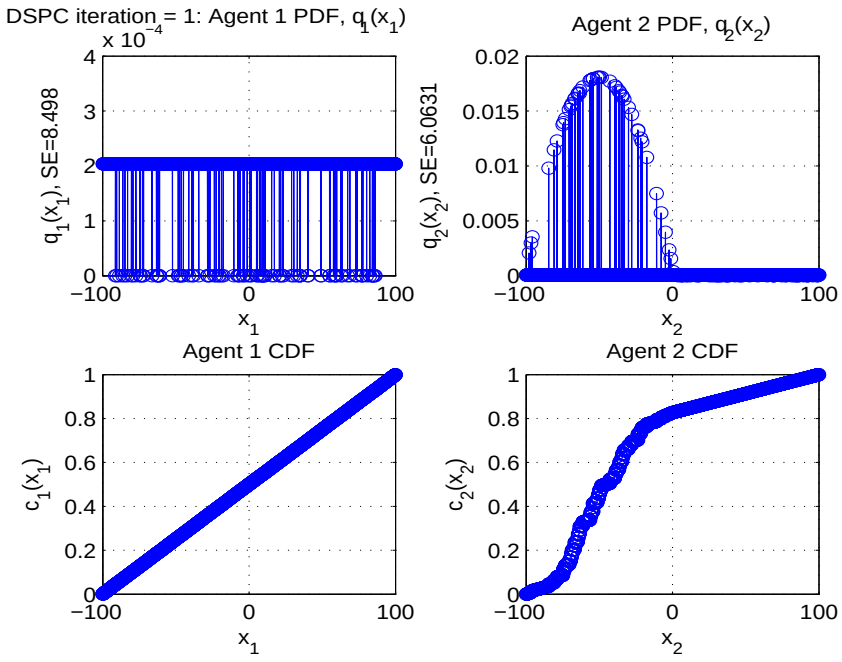


Figure 4.4: DSPC Update at $k = 1$

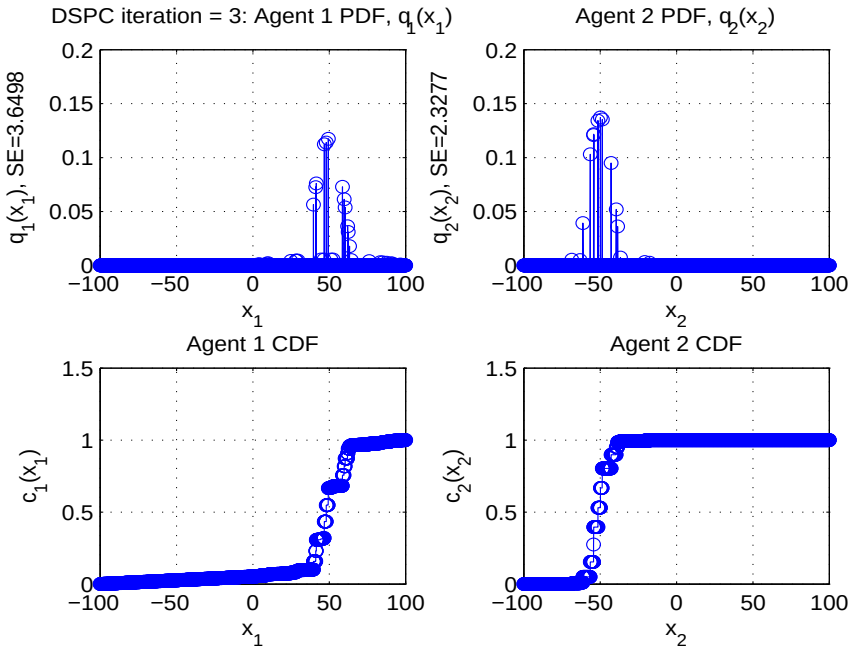


Figure 4.5: DSPC Update at $k = 3$

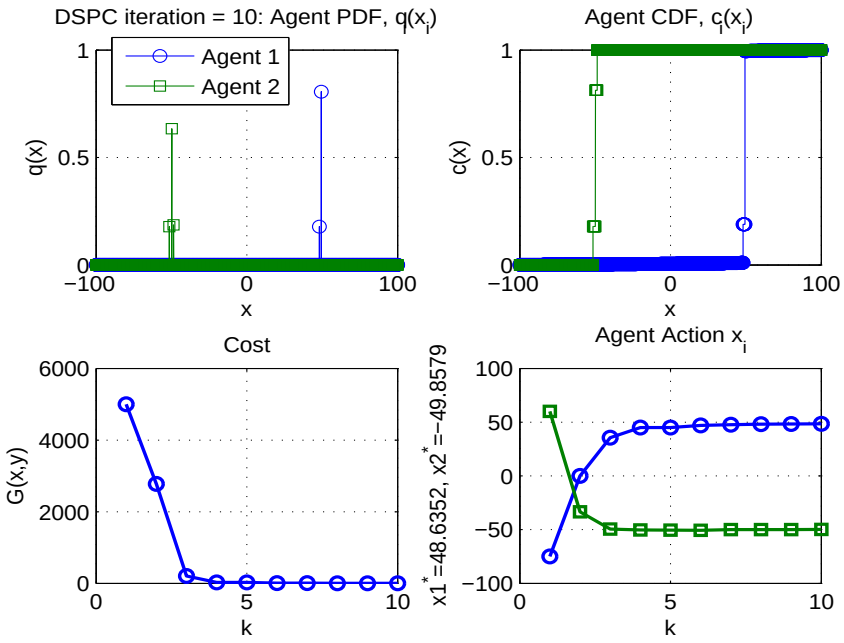


Figure 4.6: DSPC Update at $k = 10$

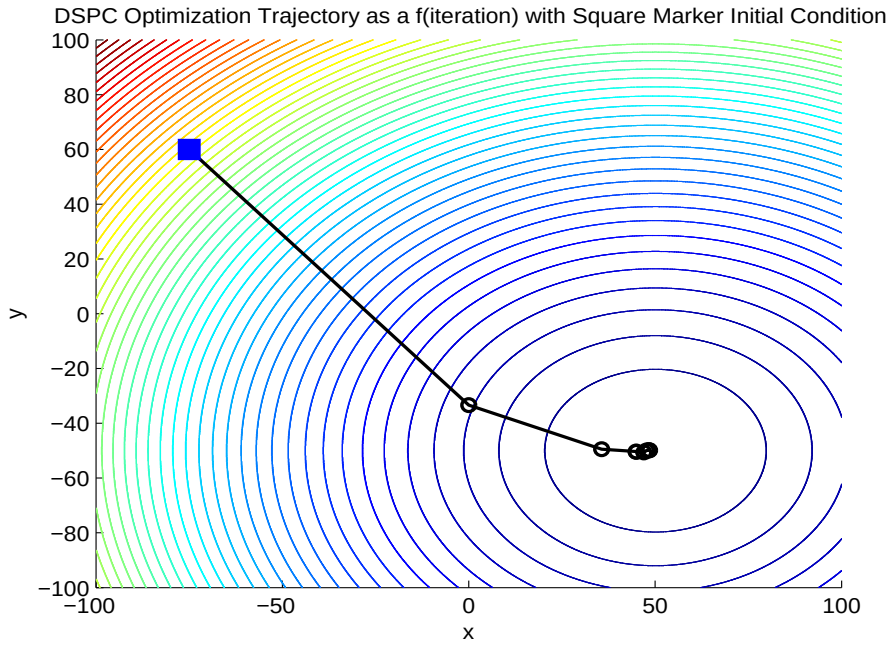


Figure 4.7: DSPC Update Trajectory

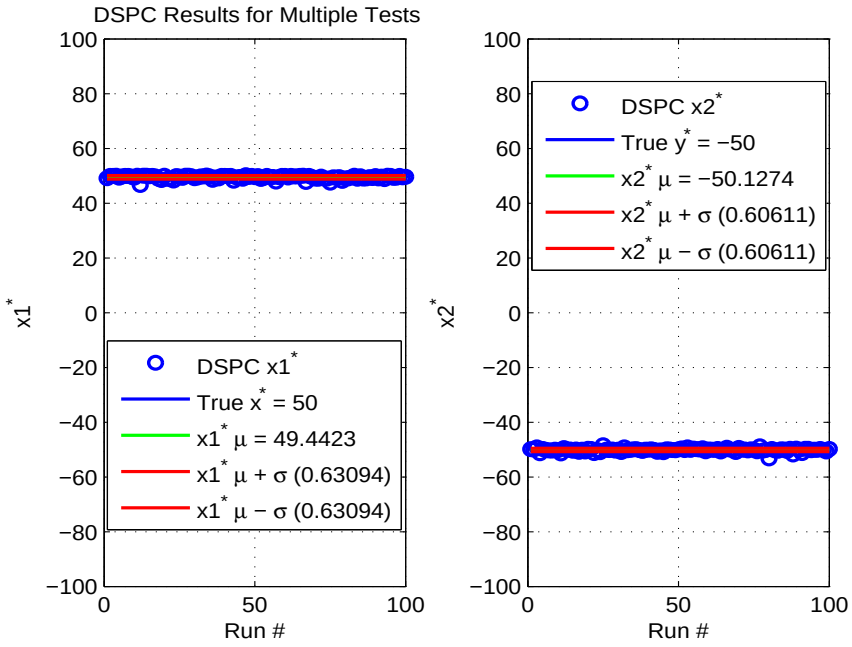


Figure 4.8: DSPC Results on Simple Two Dimensional Example for 100 Tests

4.3 Immediate Sampling PC

In [114], Rajnarayan, Kroo, and Wolpert in more recent work circa 2007 refer to Immediate Sampling PC (ISPC) as a simplified form of DSPC, with strong ties to the cross-entropy method, response surface methods, and parametric machine learning. In essence ISPC no longer treats the strategy of the agent, $\mathbf{q_i}$, as an empirical, arbitrary probability distribution. Instead, a Gaussian normal form is assumed, yielding analytic update laws for directly computing the first and second moments for the mean μ and covariance Σ, respectively. These direct analytic update laws are derived to minimize $KL(\mathbf{p}||\mathbf{q}_\theta)$ where $\theta = [\mu, \Sigma]$. The benefits of PC are still preserved by employing the transformation, however, the calculations are now streamlined. The general form of a univariate Gaussian normal distribution is used for the probability framework for each agent,

$$\mathbf{q}_{\mu_i,\sigma_i}(\mathbf{x_i}) = \frac{1}{\sigma_i\sqrt{2\pi}} exp\left[-\frac{(\mathbf{x_i}-\mu_i)^2}{2\sigma_i^2}\right] \tag{4.17}$$

although a discrete distribution is formed due to the permitted set of x_i actions with $\mathbf{q}_{\mu_i,\sigma_i}(\mathbf{x_i}) \in \Re^{n_{moves_i}}$. σ_i^2 denotes the variance of the i^{th} agent's corresponding diagonal element in the full covariance matrix, Σ. The closed form equations for the optimal mean and variance, resulting in $KL(\mathbf{p}||\mathbf{q}_\theta) = 0$, are the traditional definitions of

$$\begin{aligned} \mu_i^* &= \int_{-\infty}^{\infty} [\mathbf{x_i}\mathbf{p_i}(\mathbf{x_i})] d\mathbf{x_i} \\ \sigma_i^* &= \int_{-\infty}^{\infty} [(\mathbf{x_i}-\mu_i)^2\mathbf{p_i}(\mathbf{x_i})] d\mathbf{x_i}, \forall i \end{aligned} \tag{4.18}$$

and are problematic due to the need for full knowledge of the canonical ensemble solution, $\mathbf{p}(\mathbf{x})$, in Eq. (4.5). Instead, ISPC uses importance sampling to minimize the pq-distance by defining a sampling distribution, $\mathbf{h_i}(\mathbf{x_i^m})$, with $\mathbf{x_i^m} \in \Re^m$ as the most

recent set of m samples drawn from Eq. (4.17) [115]. Next, a scaled likelihood ratio is defined as [63, 114]

$$\mathbf{r_i^m} = \frac{exp\{-\beta g_{TG}[\mathbf{x_i^m}, \mathbf{x}_{(\mathbf{i})}^-]\}}{\mathbf{h_i}(\mathbf{x_i^m})} \tag{4.19}$$

The pq-distance optimization problem is then written as

$$\min_{\theta_\mathbf{i}} \left\{ -\sum_m \mathbf{r_i^m} ln\left[\mathbf{q}_{\theta_\mathbf{i}}(\mathbf{x_i^m})\right] \right\} \tag{4.20}$$

which is equivalent to a cross-entropy method [63, 114]. Finally, the critical points for Eq. (4.20) are provided to yield the optimal mean μ_i^* and variance σ_i^* as detailed by Rajnarayan et al in [114].

$$\mu_i^* = \frac{\sum_m \mathbf{r_i^m x_i^m}}{\sum_m \mathbf{r_i^m}} \tag{4.21}$$

$$\sigma_i^* = \frac{\sum_m \mathbf{r_i^m}(\mathbf{x_i^m} - \mu_i^*)(\mathbf{x_i^m} - \mu_i^*)^T}{\sum_m \mathbf{r_i^m}} \tag{4.22}$$

The likelihood ratio, $\mathbf{r_i^m}$, in Eq. (4.19) is computed using a Boltzmann approximation with parameter, β. In this work, β is held constant throughout the iterative process updating the agent strategy while in [63], various update laws are explored, including a simplified multiplicative recursion such as $\beta(k+1) = 1.1\beta(k)$. As ISPC converges, the higher values of β drive the Boltzmann distribution to the minimizing value of the global objective. A summary of the ISPC algorithm is outlined in Alg. (2) with details provided by Rajnarayan et al in [114, 63]. The final step in ISPC to constrain the numerical PDF to the valid probability domain is included for completeness, but is less of a factor in this simplified approach, now that the agent strategy is no longer an arbitrary, empirical distribution. Additionally, while the approach is simplified, the method of combining multiple Gaussian distributions in a mixture model is a powerful technique for capturing higher order problems with non-normal forms [63, 116].

Algorithm 2 Immediate Sampling Probability Collectives

procedure ISPC

init:

Initialize PC Parameters (β, η)

Initialize Stopping Tolerances or Maximum Steps

Initialize Agent Strategy to Uniform Distributions

loop until stop tolerance or maximum steps reached:

Draw m Samples from each agent strategy

Evaluate private utility Eq. (4.12) + constraint costs

Compute Scaled Likelihood Ratios per Eq. (4.19)

Compute Optimal θ parameters per Eqs. (4.21,4.22)

Update Lagrange Multipliers per Eq. (4.15)

Constrain PDF to Valid Domain

exit:

Draw Final Sample and Depict PDF

4.3.1 Optimization Example

The ISPC algorithm is demonstrated on the equivalent simple two dimensional optimization problem previously given in Eq. (4.16) with $\beta = 1.0$, again with two agents, $n_{agent} = 2$, with $\mathbf{x_1} \equiv \mathbf{x} \in \Re^{n_{moves_i}} \in [-100, 100]$, $\mathbf{x_2} \equiv \mathbf{y} \in \Re^{n_{moves_i}} \in [-100, 100]$. Other parameters for the example were matched with respect to the DSPC test with the same stopping condition at a maximum step size of 10 iterations, and 5000 bins, $n_{moves_i} = 5000$, defining the agent mixed strategy, $\mathbf{q_i}(\mathbf{x_i})$. Finally, 100 samples per iteration ($m = 100$) are drawn from the current strategy. The initial condition is shown in Fig. 4.9, followed by the ISPC iterations for $k = 1, 4$

in Figs. 4.10,4.11[10]. In ISPC, the strategy is evolved by evaluating the objective with the set of samples, computing the likelihood ratio in Eq. (4.19), and solving for the optimal Gaussian mean and variance in Eqs. (4.21, 4.22). The conclusion of this process at $k = 10$ is depicted in Fig. 4.12 and a two dimensional trajectory is provided in Fig. 4.13.

Figure 4.14 presents the summary of the ISPC algorithm on the simple two dimensional optimization problem previously given in Eq. (4.16) for a sweep of 100 tests, with constant $\beta = 1.0$ (update factor of 1.0) over a fixed 50 maximum iterations per test. The ISPC mean solution is nearly identical to the truth over 100 tests, and the variance is similar in performance to DSPC for this simple example[11,12].

[10]The initial condition for the ISPC optimization was set to $x_1^* = -75$ and $x_2^* = 60$ to enable trajectory comparison to the DSPC optimization example. This initialization is not required in general and both algorithms begin with a stochastic sample drawn from a uniform distribution.

[11]In this study, the ISPC algorithm required less parameter optimization for the indicated performance on this simple example. Future work investigating methods to determine the optimal parameters in the current implementation of the DSPC method is required. Intuitively, the DSPC method is expected to yield superior results to ISPC, where the former is a more generalized approach to an arbitrarily structured optimal strategy as opposed to assuming a Gaussian normal distribution, although ISPC removes the need for data aging as discussed in [63].

[12]Additionally, at the time of the sweep data for 100 tests, ISPC did not yet meet a design criterion that the number of samples times the maximum number of steps must be much less than the number of agent actions, $mn_{step} << n_{moves_i}$, enforced to minimize the number of required objective evaluations. DSPC met this goal and was further explored for control allocation.

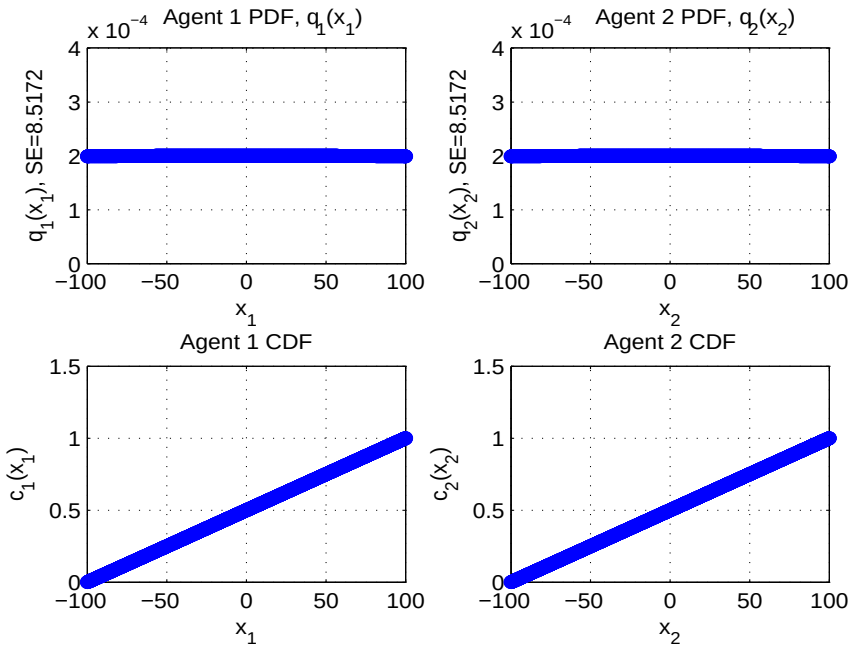


Figure 4.9: ISPC Initial Condition $k = 0$, Uniform Distribution Given Search Bounds for $\beta = 1.0$ and $m = 100$

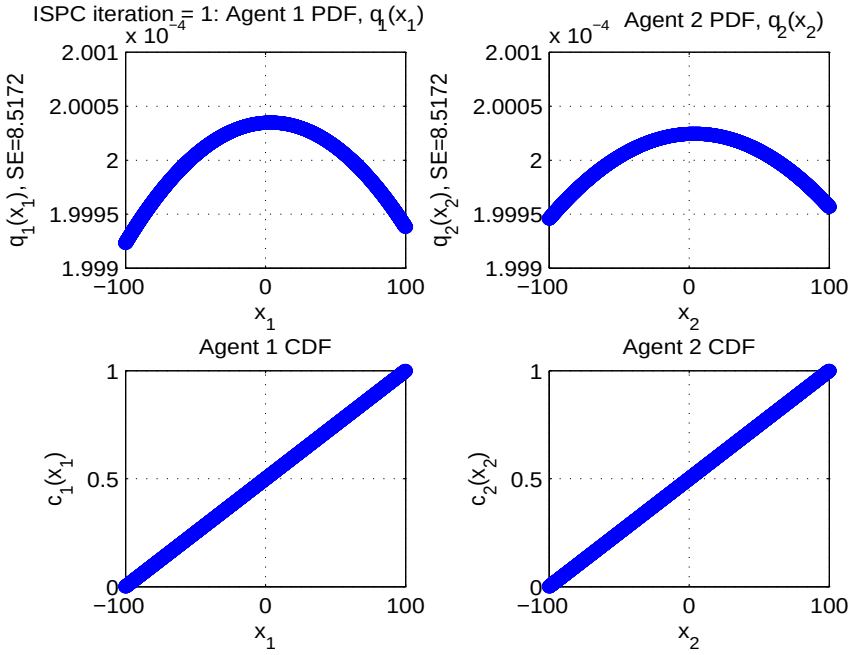


Figure 4.10: ISPC Update at $k = 1$

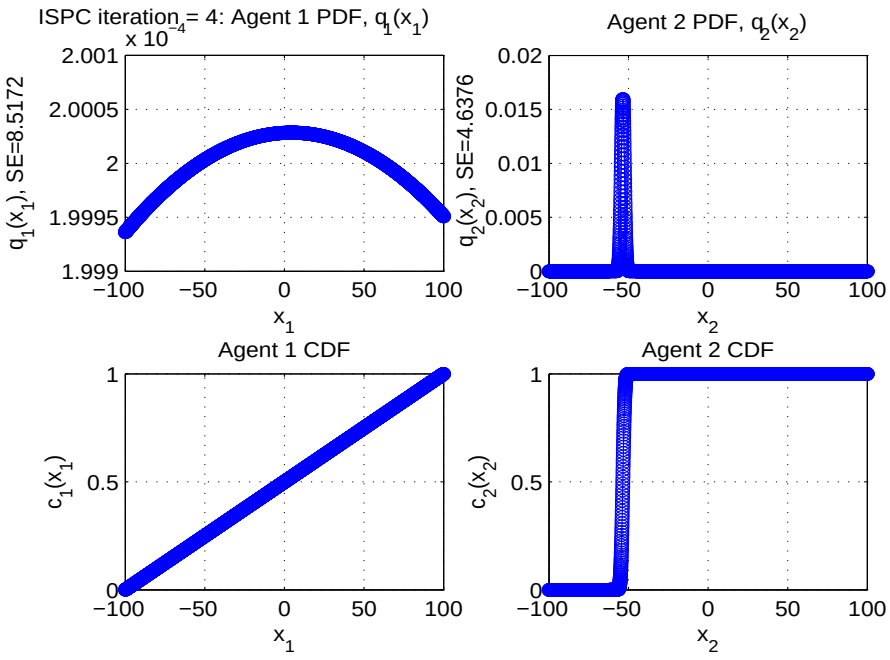


Figure 4.11: ISPC Update at $k = 4$

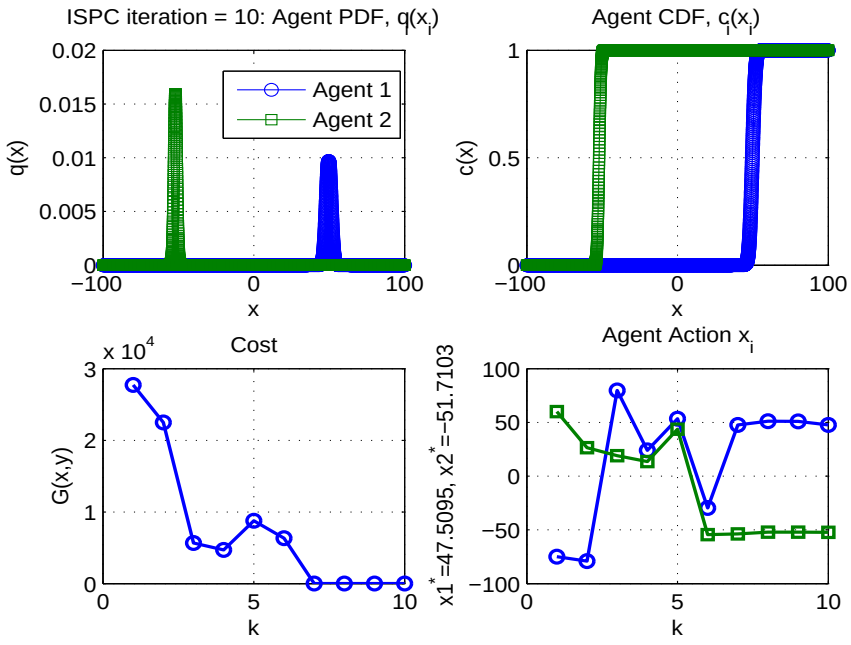


Figure 4.12: ISPC Update at $k = 10$

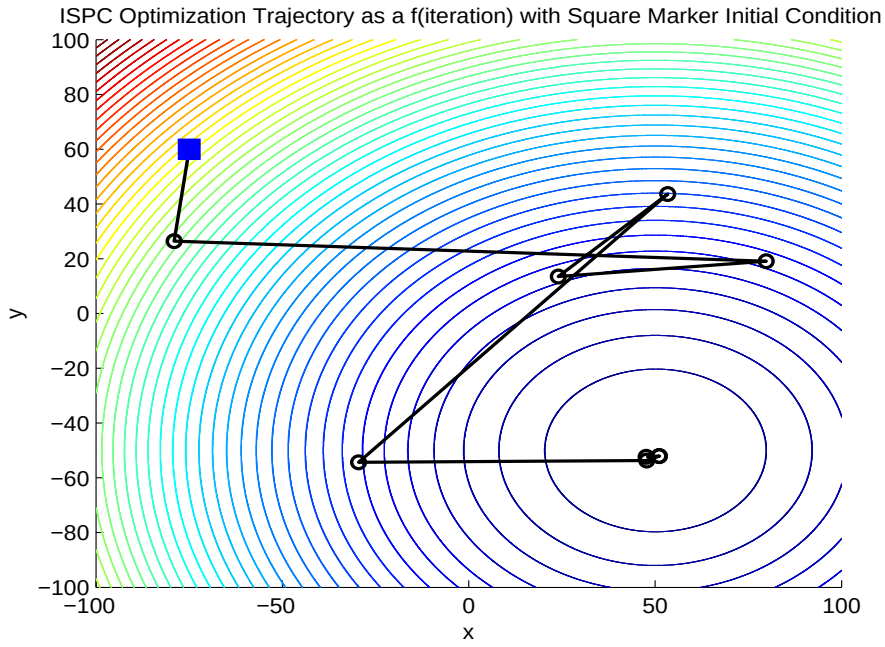


Figure 4.13: ISPC Update Trajectory

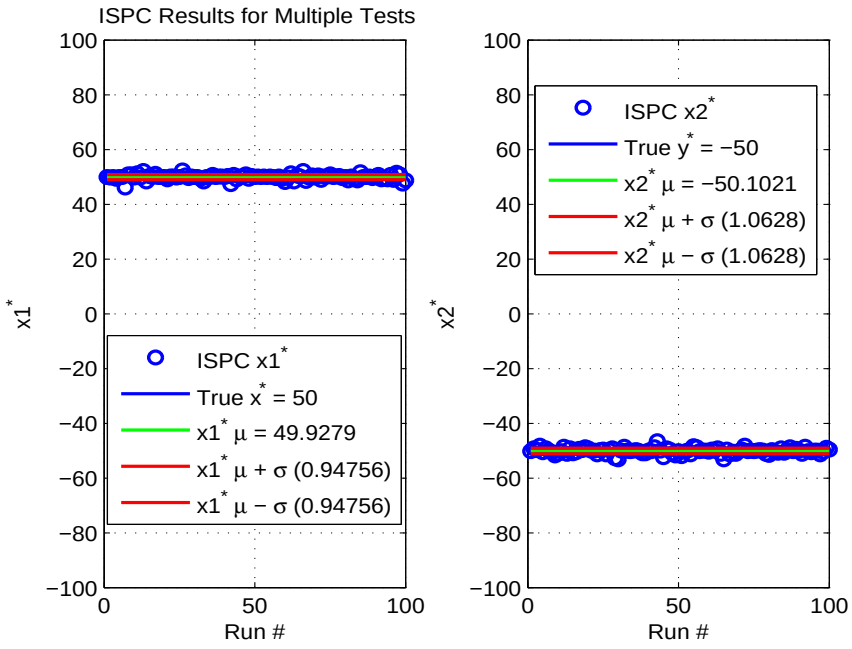


Figure 4.14: ISPC Results on Simple Two Dimensional Example for 100 Tests

4.4 Distributed Control Allocation with DSPC

The DSPC algorithm was implemented in a MATLAB Simulink® toolbox as a distributed control allocation optimization algorithm with a modular approach for each agent to allow flexibility for a range of aerospace applications with a different number of effectors. Fig. 4.15 depicts a block diagram of major components of the algorithm and expands the j^{th} agent for insight into the employed form of DSPC.

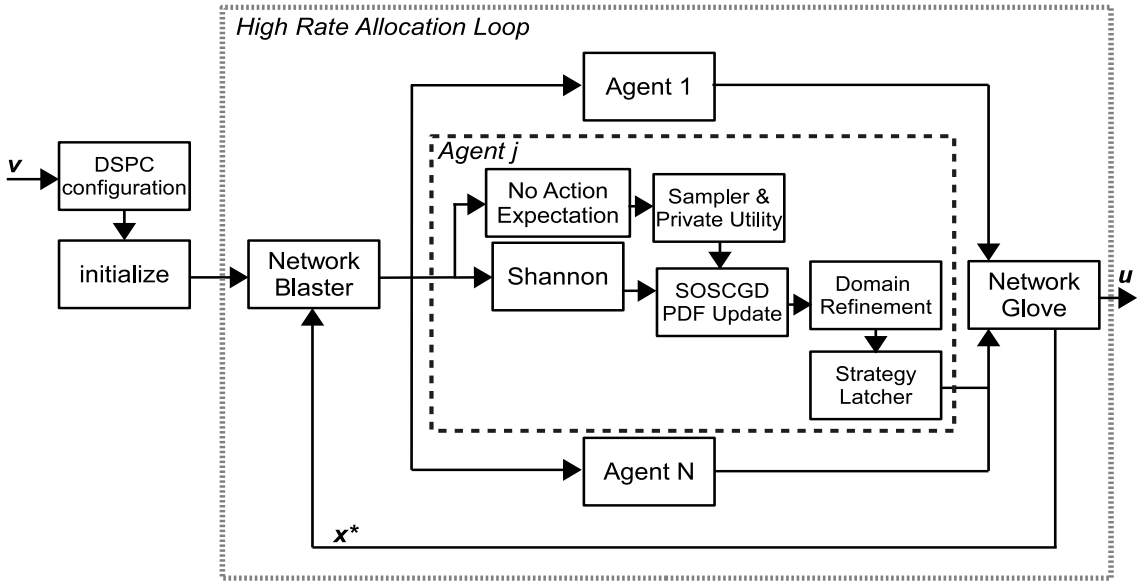


Figure 4.15: Distributed Control Allocation with DSPC

Beginning with the entry and exit points of the algorithm, the primary interface to the toolbox is the input of virtual control command $\mathbf{v}$ and the output of the effector solution $\mathbf{u}$. The first block labeled DSPC configuration reads in all parameters required to define the L_2 control allocation problem (see Eq. 3.25) and to define the DSPC algorithm described earlier in Alg. (1). Table 4.1 lists both of these sets of parameters in a combined setting here for convenient reference.

Table 4.1: Distributed Control Allocation Parameters for L_2 Optimization.

Parameter	**Variable**
Number of Agents	n_{agent}
Maxent Lagrangian Temperature	T
Probability Update Gain	α
Data Aging Gain	γ
Samples Drawn Per Iteration	m
Iterations Per Frame	n_{step}
Moves Per Agent	n_{moves_i}
Agent Effector Limit Constraint Gain	η
Agent Effector Upper Limit	$\overline{\mathbf{u}} \in \Re^{n_{agent}}$
Agent Effector Lower Limit	$\underline{\mathbf{u}} \in \Re^{n_{agent}}$
Acceleration Error Penalty	$\mathbf{W_v} \in \Re^{k \times k}$
Effector Use Penalty	$\mathbf{W_u} \in \Re^{n_{agent} \times n_{agent}}$
Effector Desired Values	$\mathbf{u}_d \in \Re^{n_{agent}}$
Effector Constraint Penalty	λ_{L_2}
L_2 Acceleration Error Penalty	γ_{L_2}

The initialization block shown in Fig. 4.15 sets all probability distributions (agent strategies) to uniform, given the specified number of moves per agent, n_{moves_i}. Next, the initialized distributions and parameter settings are sent to the network blaster which emulates a high rate distributed bus on an air vehicle. The purpose of the network blaster is to broadcast the expected value of each agent strategy for use in the local optimizations. This process occurs in a high rate allocation loop that is required to complete n_{step} iterations per execution frame from the parent caller (e.g. the inner loop stability augmentation system (SAS))[13].

At the agent scope, the first steps are to compute the Shannon entropy of the local strategy and to evaluate the L_2 objective given all expected values of the collec-

[13]For real time implementation, if the SAS inner loop system executes at 100 hz, the DSPC high rate network bus must complete the maximum number of optimization iterations (or converge to a stopping tolerance) within .01 seconds. This work did not implement the toolbox in a real time application.

tive strategy. This step is referred to in Fig. 4.15 as No Action Expectation. Next, the Sampler & Private Utility subsystem generates m from the local strategy, and evaluates the L_2 cost given these samples and the expected values of the collective. Both the No Action Expectation and the Private Utility subsystems employ a MATLAB feature referred to as model referencing for the evaluation of the nonlinear control effectivity mapping $\mathbf{g}$. This feature allows the designer to create a single model instance (Simulink diagram) of $\mathbf{g}$ in one location which is then loaded and evaluated by all agents.

The SOSCGD PDF Update block sorts through these sampled actions and cost pairs to apply the second order simplex constrained update on the probability bin associated with the action as detailed in Eq. (4.14). The updated strategy is renormalized to ensure the probability distribution is numerically valid and within limits, and is passed to the Domain Refinement block for adaptation of the search domain (see App. D.1).

The final step for the agent occurs in the Strategy Latcher block in Fig. 4.15 where the prior cost of the L_2 objective is compared with respect to the updated cost, given the expected value of the updated strategy. If the cost increases, the strategy update is not applied, and the prior strategy is restored. This latching mechanism improved performance for the L_2 control allocation examples studied in this work, however is not always effective for other optimization problems, especially with multiple extrema (refer to App. E.1).

The conclusion of the iteration in the high rate allocation loop collects all updated agent strategies into the Network Glove subsystem. The purpose of this block is to return all strategies to the Network Blaster for subsequent iterations. After completing n_{step} iterations, the Network Glove down-samples the final expected values as the effector solution, $\mathbf{u}$.

CHAPTER 5

CONTROL ALLOCATION EXAMPLES

This chapter applies the delayed sampling form of probability collectives (DSPC) to both open loop and closed loop linear examples as a distributed control allocation method, previously described at the conclusion of Ch.4. Effector saturation and effector failure examples are included. Finally, examples are presented with nonlinear control effectivity mappings. Ch. 6 expands this simple example to a larger application with higher dimensions and more effectors.

5.1 Open Loop Linear Example

A simplified L_2 linear CAO problem with $\mathbf{CB} \in \Re^{k\times m}$, $\mathbf{u}(t) \in \Re^m$, and $\mathbf{v}(t) \in \Re^k$ in Eq. (5.1) for a five-dimensional case (with $m = 5$ effectors, $k = 1$ desired virtual controls) is solved using (i) a weighted least squares (WLS) centralized method provided by Härkegård in a Quadratic Control Allocation Toolbox (QCAT)[1] detailed in [18]; and (ii) the delayed sampling PC (DSPC) distributed method.

$$\begin{aligned} \mathbf{CBu}(t) = \mathbf{v}(t) \mid \mathbf{u}_{min} \leq \mathbf{u}(t) \leq \mathbf{u}_{max} \\ \mathbf{p}_{min} \leq \dot{\mathbf{u}}(t) \leq \mathbf{p}_{max} \end{aligned} \tag{5.1}$$

[1]The QCAT WLS toolbox by Härkegård employs an active set method to solve the linear control allocation problem where multiple subproblems with a reduced set of constraints are optimized sequentially. In an iterative manner, the method regards a subset of the inequality constraints as a working set of equality constraints while disregarding the other inequality constraints, finds a feasible solution, and continues to re-introduce constraints until all are satisfied. The algorithm concludes when the working set has been optimized, referred to as an active set [18, 117, 118].

This initial open loop example is considered simplified where the allocation solution is intentionally simulated with arbitrarily large actuator limits to satisfy $[\underline{\mathbf{u}} \leq \mathbf{u}(t) \leq \overline{\mathbf{u}}]$ as previously defined in Eq. (3.9). Subsequent examples introduce increased complexity with a closed loop feedback system where control allocation is demonstrated in the presence of effector limits. Additionally, in this chapter a single virtual control variable is selected ($k = 1$) with a one to one mapping to the state and therefore $\mathbf{C} = 1$ is omitted for brevity. The objective of this open loop linear control allocation optimization problem is to determine the minimizing argument, $\mathbf{u}(t)$ for

$$L_{2C} = ||\mathbf{W_u}(\mathbf{u} - \mathbf{u}_d)||_2 + \gamma_{L_2}||\mathbf{W_v}(\mathbf{Bu} - \mathbf{v})||_2 + \lambda_{L_2}||\mathbf{u} - \underline{\mathbf{u}}||_2 + \lambda_{L_2}||\mathbf{u} - \overline{\mathbf{u}}||_2 \quad (5.2)$$

with a linearized control matrix of $\mathbf{B} = \begin{bmatrix} 1 & .8 & -.1 & -1 & 2.8 \end{bmatrix}$ with $\mathbf{W_u} = \mathbf{I} \in \Re^{5x5}$, $\mathbf{W_v} = 1$, $\mathbf{u}_d = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \end{bmatrix}'$, $\gamma_{L_2} = 1$, and $\lambda_{L_2} = 0$ nominally or $1e9$ during effector saturation[2]. The effector limits for this first case are set arbitrarily large to avoid saturation at $\overline{\mathbf{u}} = \begin{bmatrix} 5 & 5 & 5 & 5 & 5 \end{bmatrix}'$ and $\underline{\mathbf{u}} = -\overline{\mathbf{u}}$. The DSPC configuration employed for this example was $T = 0.15$, $\alpha = 0.3$, $\gamma = 0.3$, $n_{agent} = 5$, $\eta = 0.1$, with a stopping condition at 15 maximum iterations. The i^{th} agent mixed strategy $\mathbf{q_i}(\mathbf{x_i})$ used $n_{moves_i} = 3000$ bins with $m = 100$ samples drawn per iteration.

Figure 5.1 depicts the $\mathbf{u_i}$ solutions for both the QCAT WLS and DSPC methods which are in agreement for a sinusoidal driving $v(t) = 15\sin(5t)$ command. The control allocation error and total cost, $L_2(t)$ in Eq. (3.12), is depicted in Fig. 5.2. In brief, DSPC nearly meets the performance of the QCAT WLS centralized allocator but offers the benefit of the distributed setting as discussed in Ch. 4. In this data, the sinusoidal driving virtual command is depicted in the first subplot, and the error between the actual achieved value with respect to the command is shown in the second subplot. The most significant deviations in virtual command tracking for

[2] $\lambda_{L_2} = 0$ when $\underline{\mathbf{u}} \leq \mathbf{u}(t) \leq \overline{\mathbf{u}}$, otherwise $\lambda_{L_2} = 1e9$ in all control allocation examples in this work.

DSPC occur at the largest magnitude $v(t)$ commands per the second subplot. This response may be indicative of the effector #5 strategy approaching the bounds of the probability distribution. Finally, the third subplot of Fig. 5.2 shows the L_2 cost time history. In the cases where $L_2 \approx 0$, both control allocation routines achieve a near perfect solution, where the virtual command error is small (and the $v(t)$ command is zero) and the control effector solution is near desired (at zero). As the $v(t)$ command oscillates, the $L_2(t)$ responds in kind due to the deviation in tracking error (second subplot) and the effector positions (previously shown in Figure 5.1).

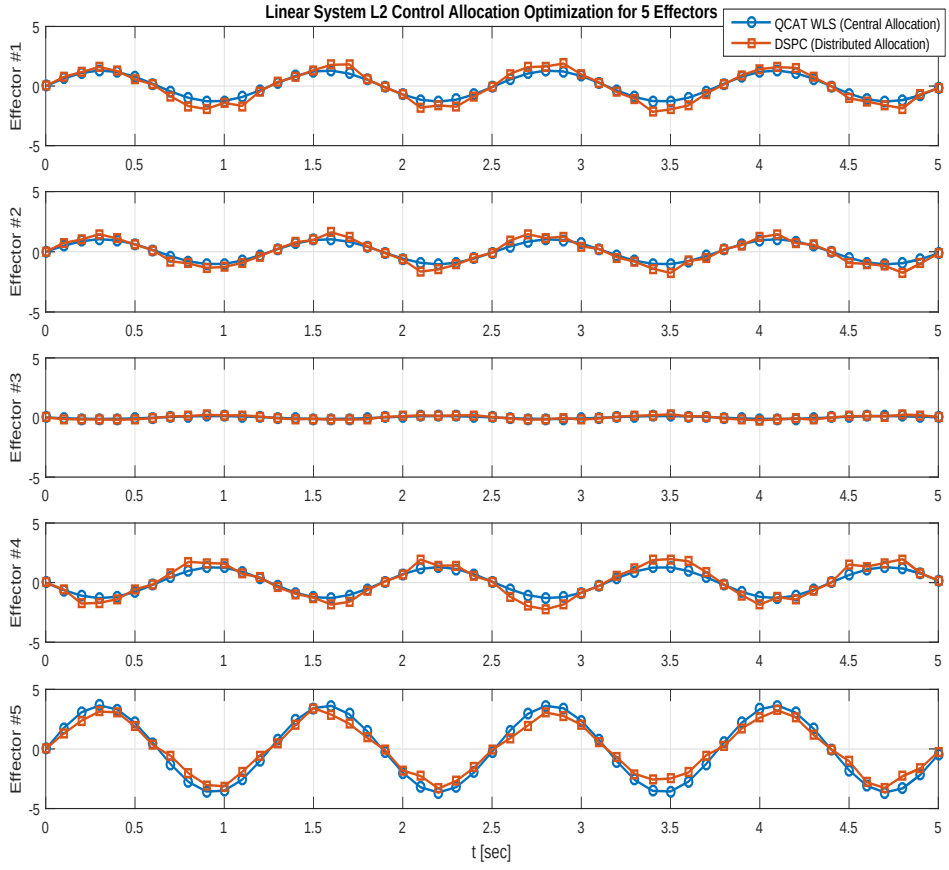


Figure 5.1: Solution to 5 Effector Open Loop Linear System L_2 Control Allocation Problem

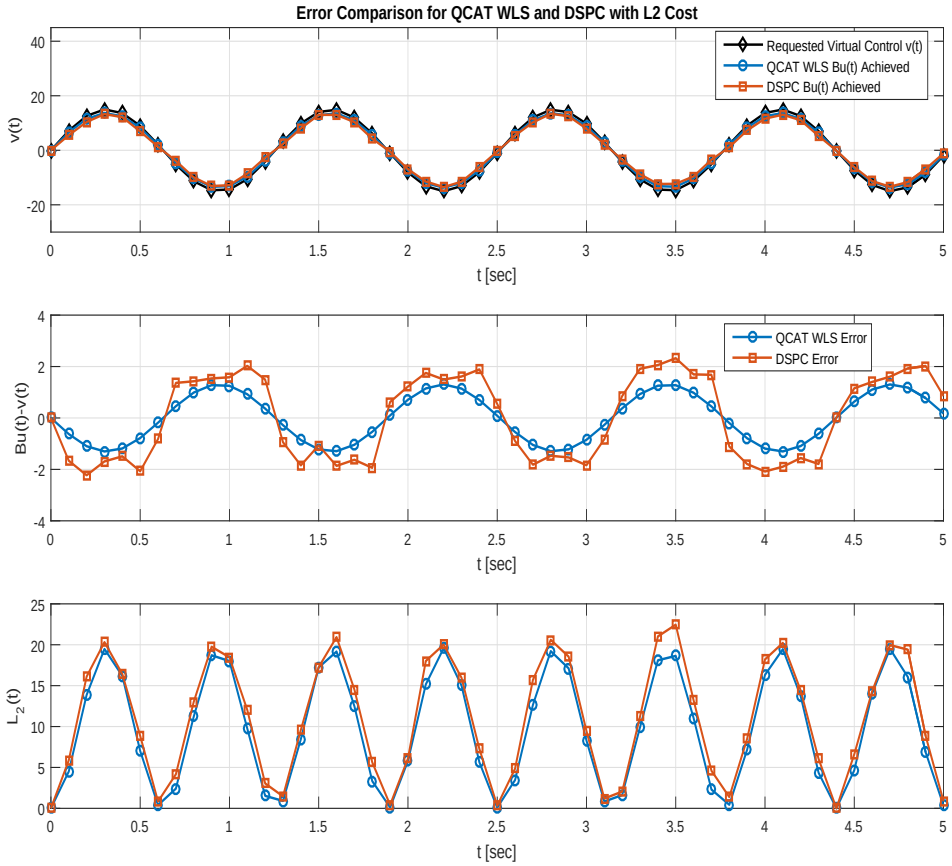


Figure 5.2: Error Comparison and L_2 Cost for QCAT WLS and DSPC for 5 Effector Open Loop Linear System L_2 Control Allocation Problem

5.2 Closed Loop Linear Examples

In this section, the control objective is to track one desired output ($k = 1$) $y_d \in \Re^k$ given the linear dynamical system in Eq. (5.3) with full state feedback, $\mathbf{C} \in \Re^{k \times n}$. In this case, $n = k = 1$, and the open loop dynamics matrix is a scalar as

$A \in \Re^1$. The linear control effectivity matrix is $\mathbf{B} \in \Re^{1 \times m}$ for $m = 5$ control effectors constituting the required input signal $\mathbf{u} \in \Re^5$.

$$\begin{aligned} \dot{x} &= Ax + \mathbf{B}\mathbf{u} \\ y &= Cx \end{aligned} \tag{5.3}$$

5.2.1 Nominal Case

For a simplified closed loop linear example, an architecture was constructed to mimic the dynamic inversion example previously presented in Fig. 3.3, with linearized control matrix of $\mathbf{B} = \begin{bmatrix} 1 & .8 & -.1 & -1 & 2.8 \end{bmatrix}$ as used in the open loop example, $A = \begin{bmatrix} 0.2 \end{bmatrix}$, and $C = \begin{bmatrix} 1.0 \end{bmatrix}$. Also, $\mathbf{W_u} = \mathbf{I} \in \Re^{5x5}$, $W_v = 1$, $\mathbf{u}_d = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \end{bmatrix}'$, and $\gamma_{L_2} = 1$, however effector limits are now set to $\overline{\mathbf{u}} = \begin{bmatrix} 3 & 3 & 3 & 3 & 3 \end{bmatrix}'$ and $\underline{\mathbf{u}} = -\overline{\mathbf{u}}$. The DSPC configuration employed for this example was $T = 0.1$, $\alpha = 0.1$, $\gamma = 0.3$, $n_{agent} = 5$, $\eta = 0.1$, with a stopping condition at 10 maximum iterations. The i^{th} agent mixed strategy $\mathbf{q_i}(\mathbf{x_i})$ used $n_{moves_i} = 1000$ bins with $m = 90$ samples drawn per iteration.

Figure 5.3 depicts the closed loop system response using both QCAT WLS and DSPC distributed control allocation. The desired state position tracks to the command with the specified first order response, with a τ of 2.0 seconds, in general for both methods. The QCAT WLS routine exhibits less tracking error in the steady state step response which is indicated by the overall L_2 peak costs for this nominal case. The effector positions associated with this time history are presented below in Fig. 5.4 and all values are within limits for this nominal case. Note effector #5 is used most heavily by both control allocation routines as the minimizing L_2 solution. This solution is intuitive as the highest effectivity magnitude in the given $\mathbf{B}$ matrix is 2.8 for this effector.

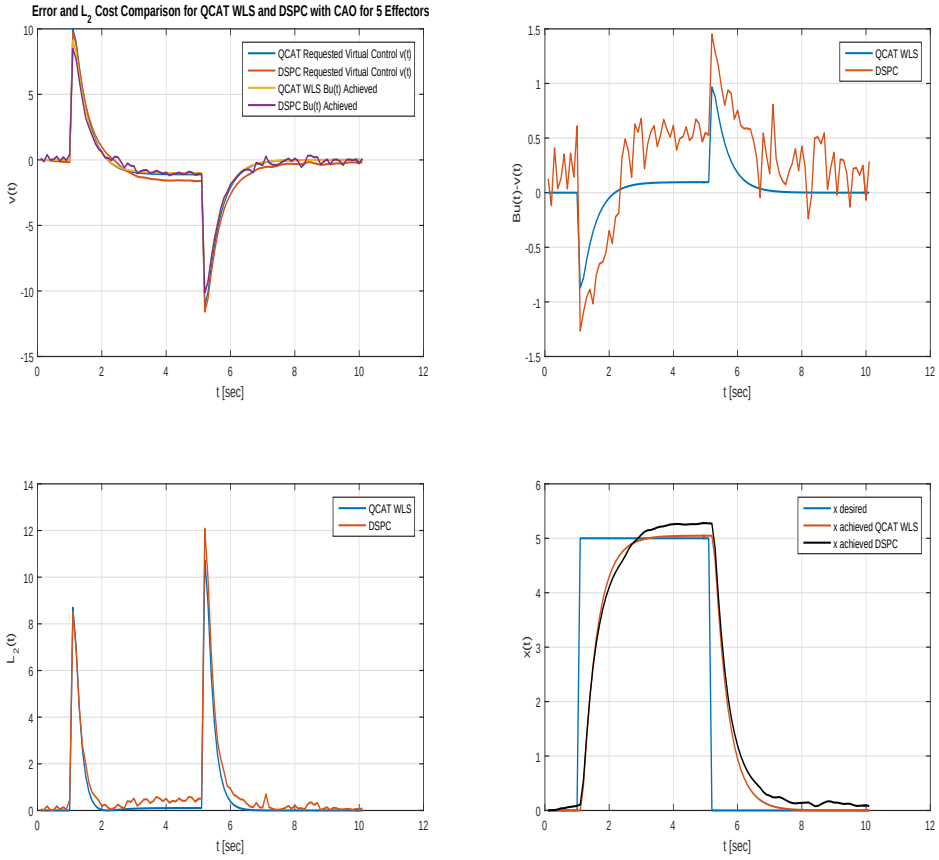


Figure 5.3: Nominal Case: Error and L_2 Cost Comparison for QCAT WLS and DSPC ($T = .1$) with CAO for 5 Effectors

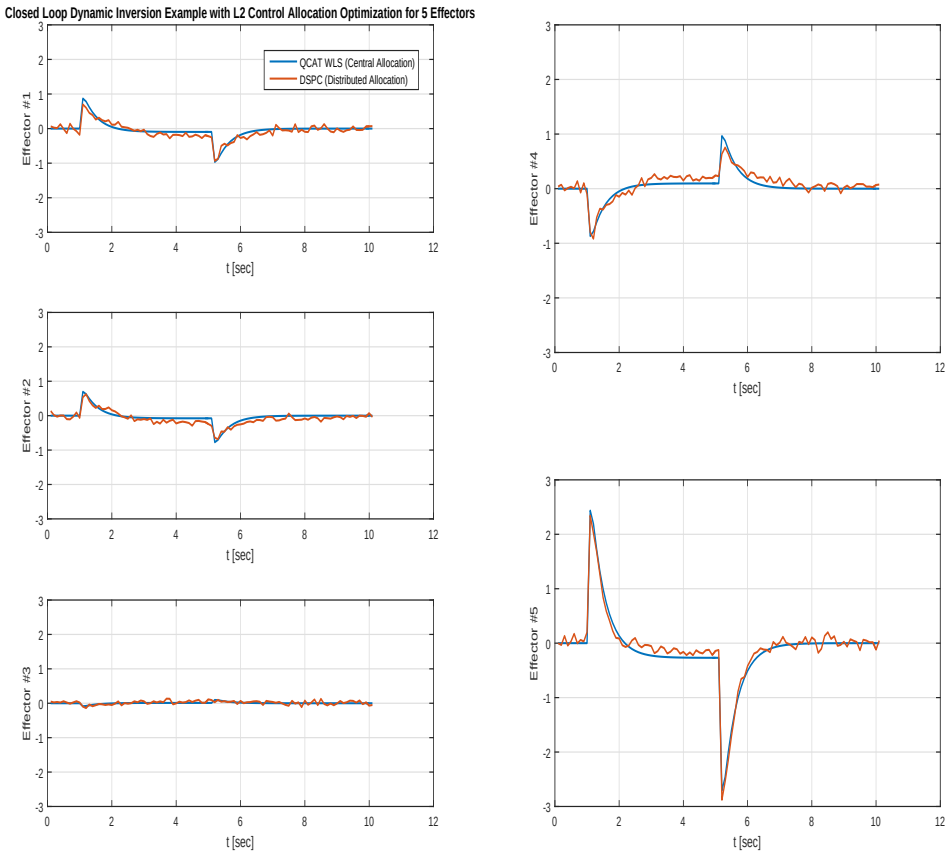


Figure 5.4: Nominal Case: Closed Loop Dynamic Inversion Example with L_2 Control Allocation Optimization for 5 Effectors

5.2.2 Effector Saturation Case

In this example, the effector position limits were arbitrarily chosen to $\overline{\mathbf{u}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \end{bmatrix}'$ and $\underline{\mathbf{u}} = -\overline{\mathbf{u}}$ in order to demonstrate saturation for the two control allocation methodologies. Fig. 5.5 depicts the L_2 cost for both QCAT WLS and DSPC. The same DSPC parameters employed in the previous nominal case were used for this example. Both approaches stabilize the system and track the desired state command fairly well, however in this case, the peak L_2 cost has more than doubled with respect to the nominal case. This response is due to the degraded $\mathbf{B}\mathbf{u}(t) - v(t)$ tracking as well as an increased use of effector #3 during the saturation. Refer to Fig. 5.6 which shows that during the peak commands, all effectors saturate other than #3. Considering this response with respect to the nominal unlimited cases previously shown, the saturation of effector #5 leads to a dramatic increase in the use and saturation of all other effectors.

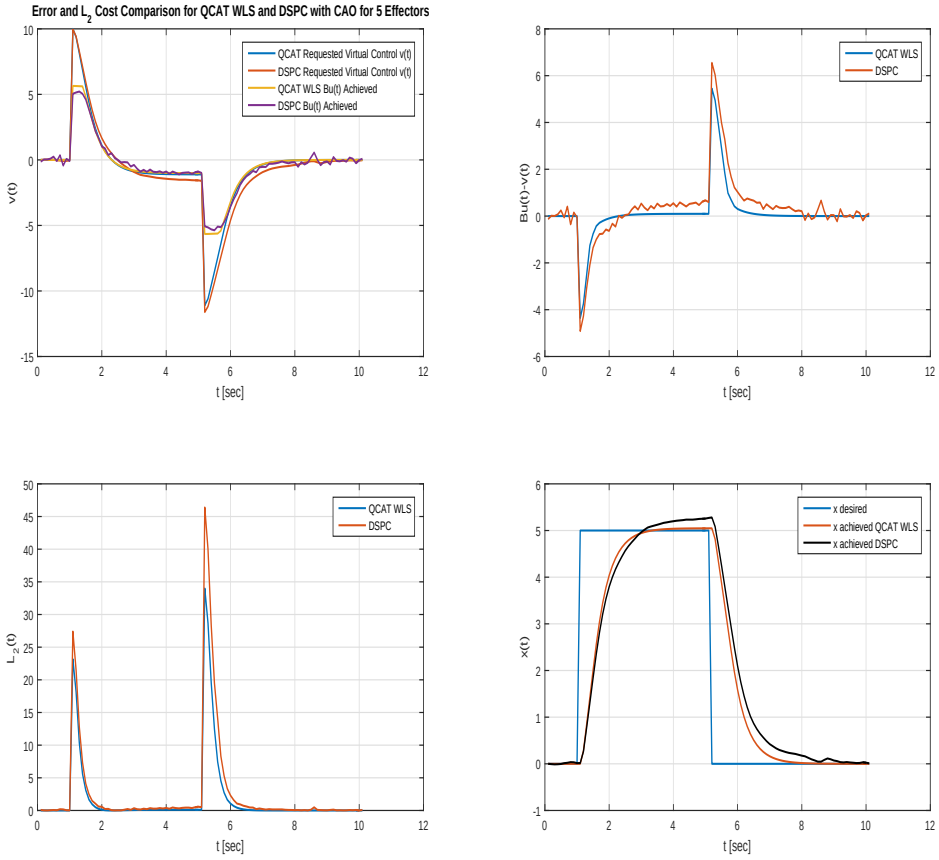


Figure 5.5: Saturated Case: Error and L_2 Cost Comparison for QCAT WLS and DSPC ($T = .1$) with CAO for 5 Effectors

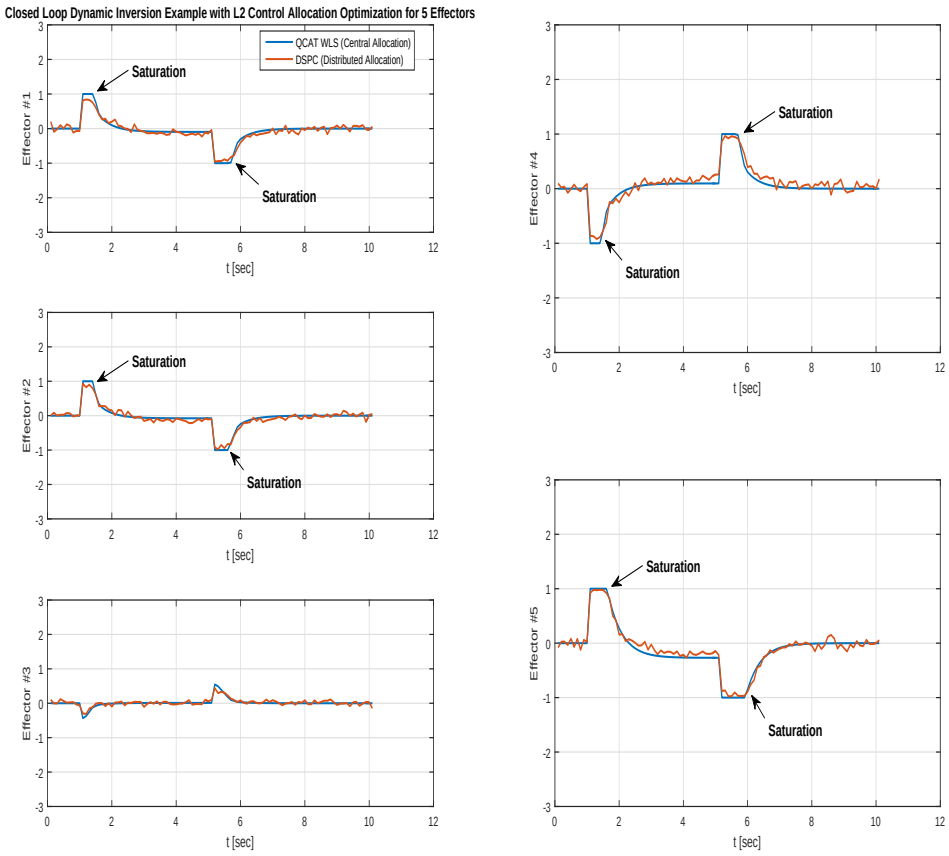


Figure 5.6: Saturated Case: Closed Loop Dynamic Inversion Example with L_2 Control Allocation Optimization for 5 Effectors

5.2.3 Effector Failure Case

In this closed loop example, one effector is simulated to fail after the closed loop system establishes steady state tracking to an initial step response at approximately 8.0 seconds simulation time. The failure scenario is modeled as a runaway effector where the local actuator driving the position of the surface (or engine) erroneously commands to an incorrect value as shown in Fig. 5.8. In this initial case A, observability is modeled with respect to the fail state, and the centralized architecture accommodates the commands to the other effectors accordingly. This approach is accomplished by solving the CAP with QCAT WLS with an additional constraint, where the failed effector #5 is treated as a position limit using the measured value of the runaway. At 8.0 seconds, the central Flight Control Computer (FCC) with QCAT WLS receives an actuator failure report and is programmed to treat the effector as not commandable and solves the allocation problem with $\overline{\mathbf{u}} = \begin{bmatrix} 3 & 3 & 3 & 3 & u_5^- \end{bmatrix}'$ where u_5^- represents the measured value of the failed effector. Here, the upper and lower limit for effector #5 is set to the same value with $\underline{\mathbf{u}} = \begin{bmatrix} -3 & -3 & -3 & -3 & u_5^- \end{bmatrix}'$. On the other hand, no provisions are implemented in the distributed control allocation approach with DSPC. Agent #5 simply ceases to optimize locally and the remaining effector suite continues allocation despite the fixed surface. Fig. 5.7 presents the resultant trajectory for this failure model, referred to as Failed Case A. At the onset of the failure, there is a clear degradation in the desired state tracking, and the L_2 cost increases dramatically. The cost increase is also due to the response and magnitude increase of the remaining healthy effectors at 8.0 seconds simulation time shown in Fig. 5.8. In this Case A with failure observability at the central FCC, where an active monitoring system reports the erroneous surface or disabled engine, the QCAT WLS and DSPC approaches exhibit similar tracking errors, with the exception of a small

bias in the DSPC system. This tracking bias sensitivity to DSPC parameter selection is further discussed ahead in Section 5.3.1.

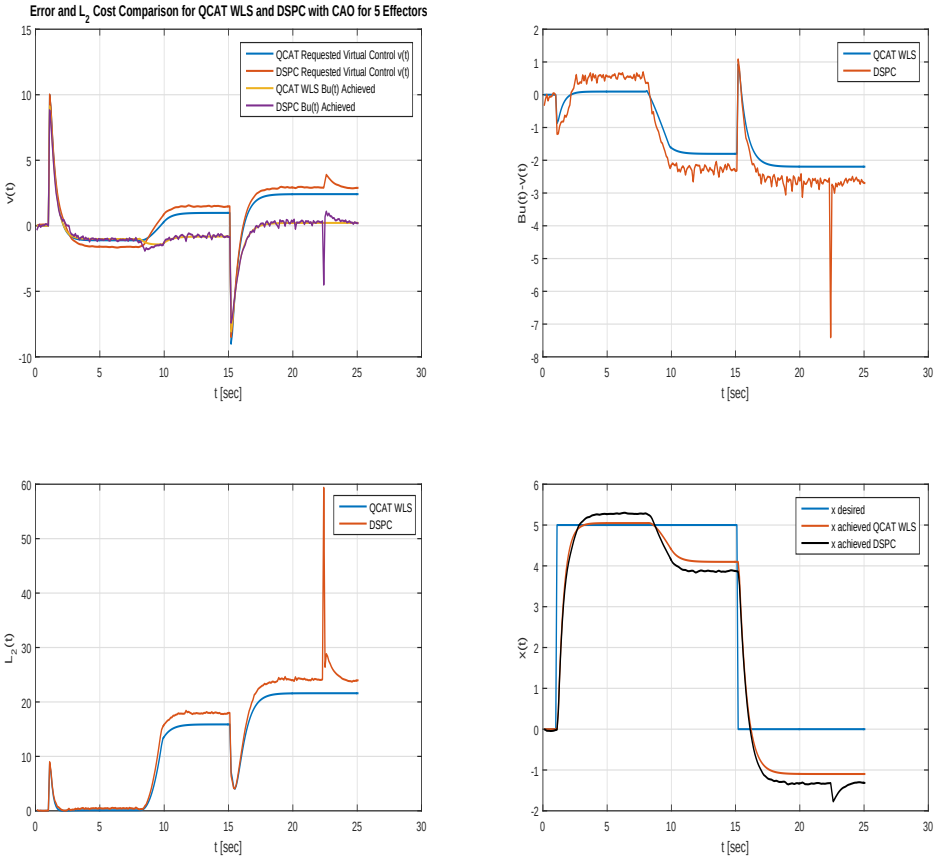


Figure 5.7: Failed Case A: Error and L_2 Cost Comparison for QCAT WLS and DSPC ($T = .1$) with CAO for 5 Effectors

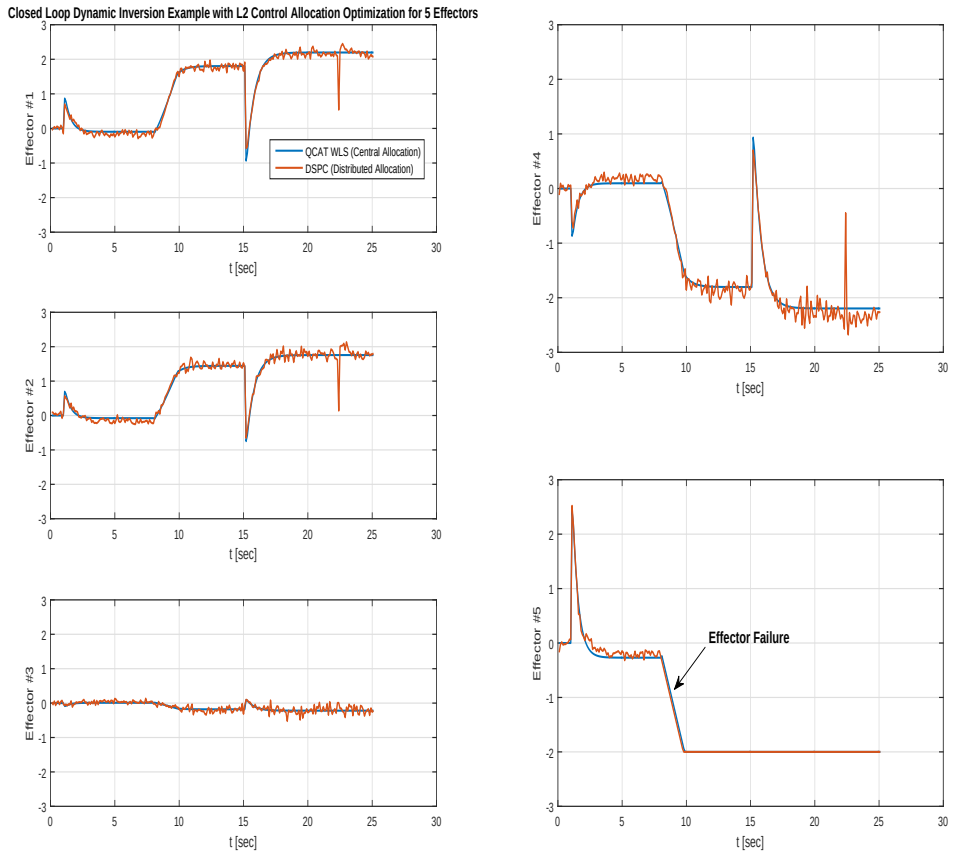


Figure 5.8: Failed Case A: Solution for Closed Loop Dynamic Inversion Example with L_2 Control Allocation Optimization for 5 Effectors

Alternatively, in a case referred to as Failed Case B, the central FCC is modeled to not have the benefit of observability for the failure mode. Specifically, the effector response is not identified as erroneous and no provisions are employed in the control allocation methods. A key assumption in this scenario is that the position of the failed effector remains valid and available on a communications bus. This type of failure highlights the benefit of DSPC. Even without a failure mode identification, the distributed network of DSPC enables the remaining agents to adjust commands accordingly to mitigate the runaway of effector #5. Fig. 5.9 depicts the virtual

command response and error, L_2 cost, and state tracking for both QCAT WLS and DSPC. After the failure at 8.0 seconds, a sharp rise in the L_2 cost is observed in QCAT WLS due primarily to the onset of control allocation error shown as $\mathbf{Bu}(t) - v(t)$. As a result, the QCAT WLS state response in the bottom right subplot deviates significantly, well beyond the error magnitude of the DSPC method. DSPC is able to maintain less virtual command error due to the distributed nature of the algorithm, although only in this case where a distributed network continues to provide accurate information about actual effector positions. The next Failure Case C demonstrates this worst case scenario where there is no information available on the runaway. The corresponding effector time histories for Failed Case B are shown in Fig. 5.10.

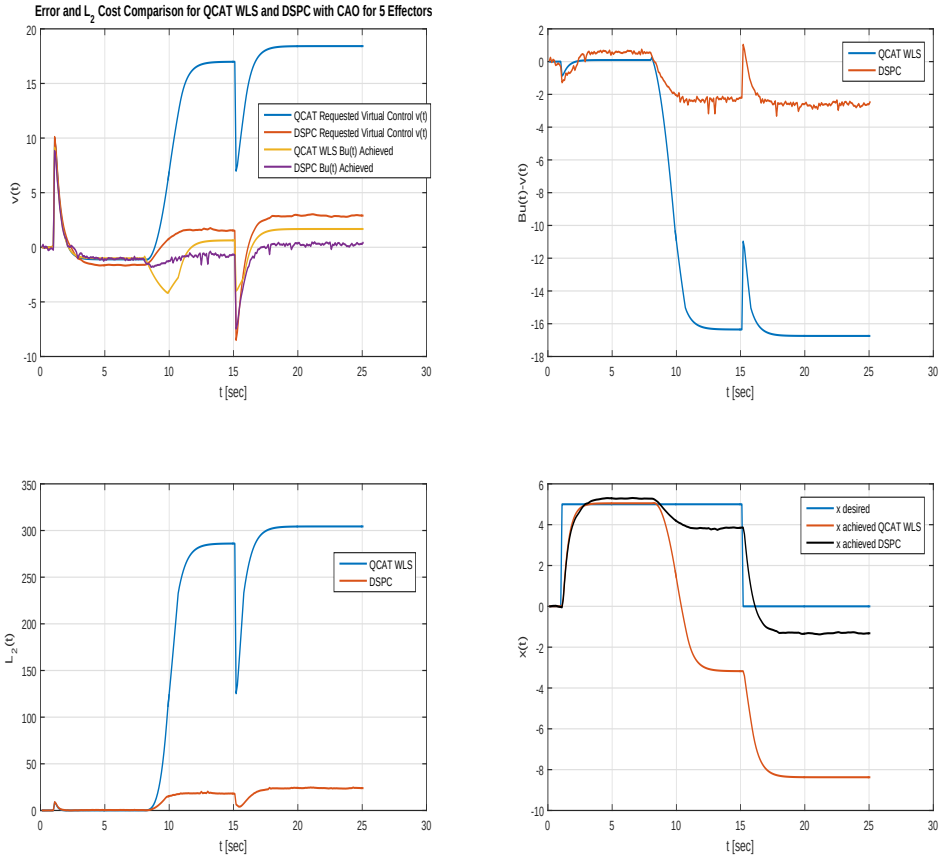


Figure 5.9: Failed Case B: Error and L_2 Cost Comparison for QCAT WLS and DSPC ($T = .1$) with CAO for 5 Effectors

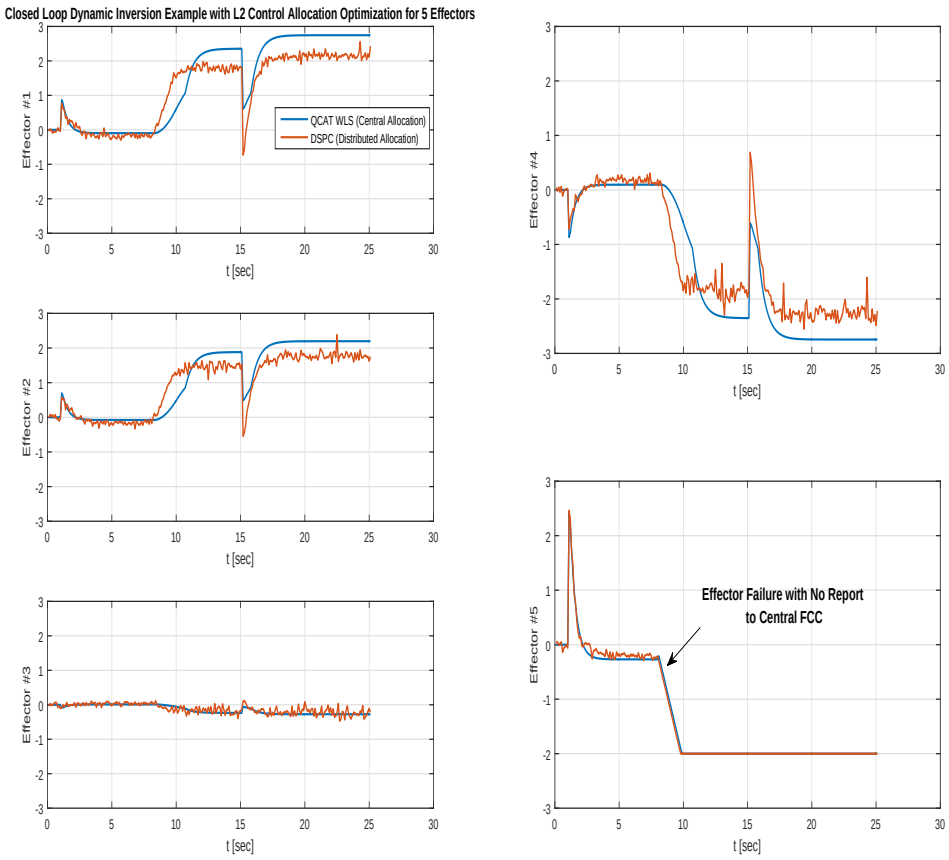


Figure 5.10: Failed Case B: Solution for Closed Loop Dynamic Inversion Example with L_2 Control Allocation Optimization for 5 Effectors

A final Failed Case C is simulated where the runaway effector #5 is completely unobservable. In this scenario, both the central FCC and the distributed network receive an erroneous feedback signal on the measured position of the faulty effector. The control allocation routines continue to generate commands to all effectors without any provisions for the runaway. As shown in Fig. 5.11, neither system responds well to this scenario and, after the onset of the failure, virtual command and state tracking diverge dramatically with a sharp rise in the L_2 cost.

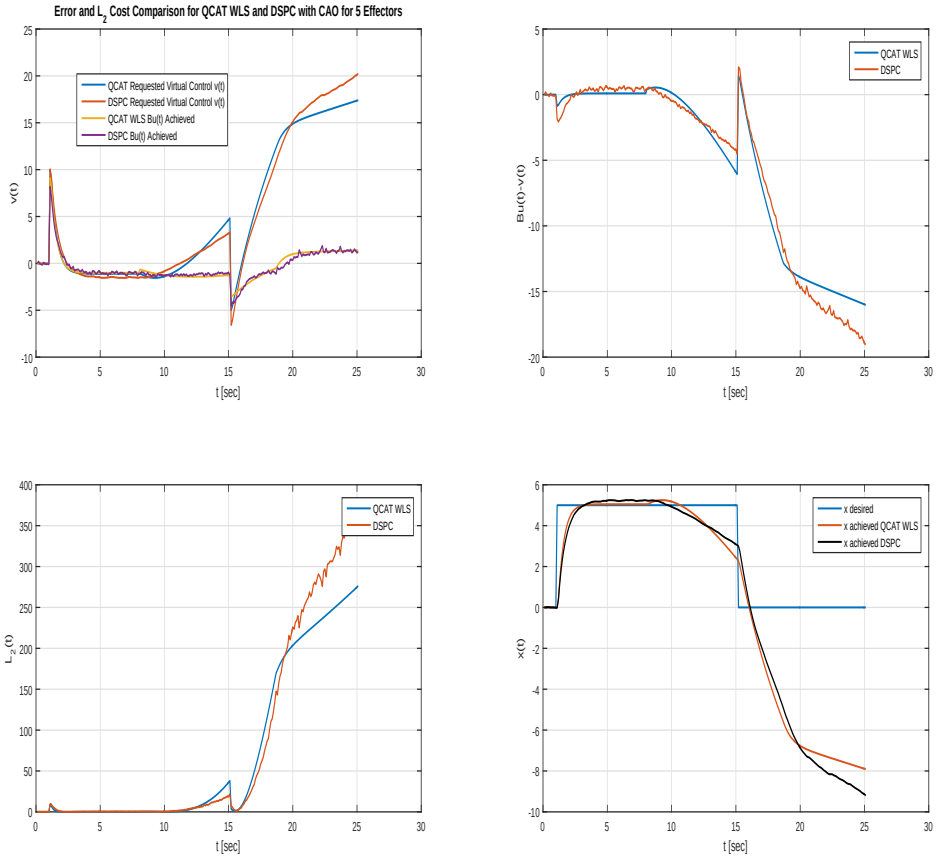


Figure 5.11: Failed Case C: Error and L_2 Cost Comparison for QCAT WLS and DSPC ($T = .1$) with CAO for 5 Effectors

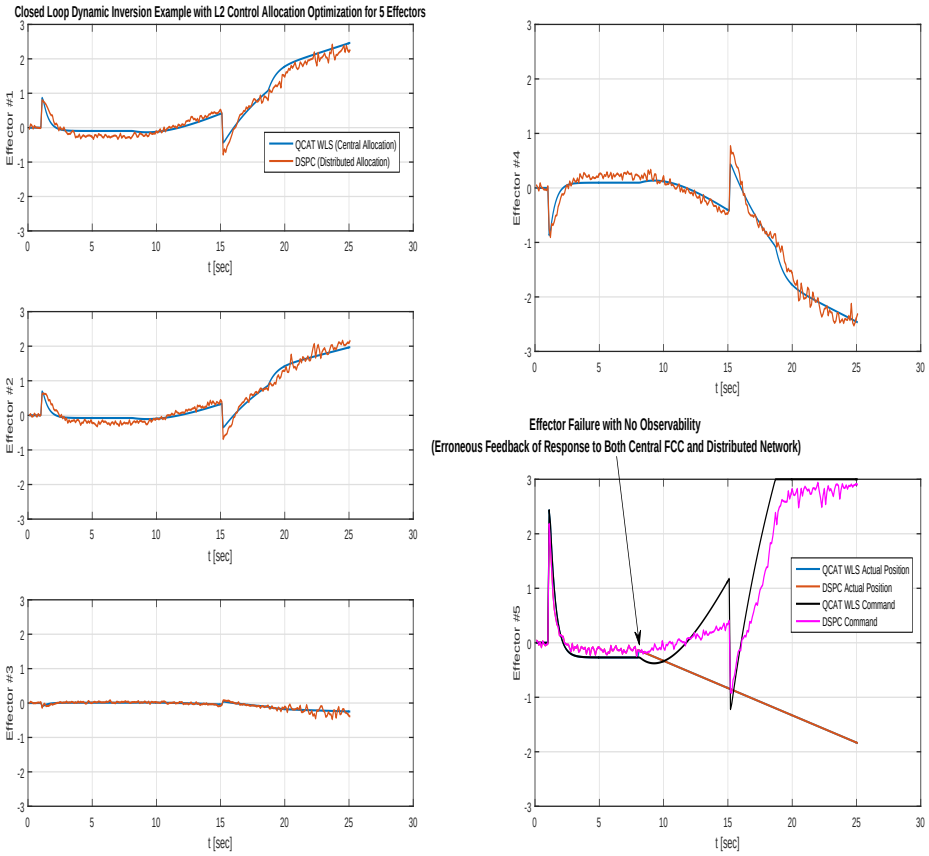


Figure 5.12: Failed Case C: Solution for Closed Loop Dynamic Inversion Example with L_2 Control Allocation Optimization for 5 Effectors

5.3 Closed Loop Nonlinear Examples

In this section, the closed loop control objective is equivalent to the prior examples and one desired output $y_d \in \Re^k$ with $k = 1$ is to be tracked. However, the dynamical system is now nonlinear due solely to the control effectivity mapping and takes the general form

$$\begin{aligned} \dot{x}(t) &= Ax + \mathbf{G}(x, \mathbf{u}) \\ y(t) &= x \end{aligned} \tag{5.4}$$

The system consist of a single state, $x \in \Re^1$ as the desired virtual command to track yielding $n = k = 1$, and $A \in \Re^{1\times 1}$. Additionally, due to perfect state feedback, $\nabla_x \mathbf{h} = 1$ and therefore $\mathbf{g}(x, \mathbf{u}) \equiv \mathbf{G}(x, \mathbf{u}) : \Re^6 \mapsto \Re^1$ for $m = 5$ control effectors constituting the required input signal $\mathbf{u} \in \Re^5$.

5.3.1 Nonlinear Control Allocation Example 1

A contrived nonlinear, non-affine control effectivity mapping is created in this example as

$$\begin{aligned}\mathbf{g}[x(t), \mathbf{u}(t)] =& .05x^2u_1 + .5signum(|x - 10|)u_2^2 - .05u_1\cos(x)e^{u_3} \\ & - .5signum(|x - 10|)cos(u_4)x + u_2u_5\end{aligned} \tag{5.5}$$

This configuration is tested using the identical parameters as previous for the linear control allocation example with $A = \begin{bmatrix} 0.2 \end{bmatrix}$, $\mathbf{W_u} = \mathbf{I} \in \Re^{5x5}$, $W_v = 1$, $\mathbf{u}_d = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \end{bmatrix}'$, $\gamma_{L_2} = 1$, $\overline{\mathbf{u}} = \begin{bmatrix} 3 & 3 & 3 & 3 & 3 \end{bmatrix}'$ and $\underline{\mathbf{u}} = -\overline{\mathbf{u}}$, with DSPC at $T = 0.35$, $\alpha = 0.1$, $\gamma = 0.3$, $n_{agent} = 5$, $\eta = 0.1$, 10 maximum iterations, $n_{moves_i} = 1000$ bins, and $m = 90$ samples. An initial state value, x_0, is set to 20.0.

In this instance and the subsequent examples in this section, the DSPC distributed control allocation scheme is only updated with the ability to evaluate the effectivity mapping at each agent[3]. For the QCAT WLS evaluation, however, a linearized control effectivity mapping ($\mathbf{B}$ matrix) is required. Two methods are considered where the $\mathbf{B}$ linearized matrix is computed analytically and numerically as

[3]Future work will consider estimation of the global objective at the local agent scope as well as reduced communication topologies between agents. This research employs a complete graph for the agent adjacency matrix [33, 36].

Nonlinear Case A and Nonlinear Case B, respectively. In Case A, the analytic expression for $\mathbf{B}$ is derived as $\frac{\partial \mathbf{g}}{\partial \mathbf{u}}(x, \mathbf{u}_0)$ yielding

$$\begin{aligned}\mathbf{B}(x) = [\ & .05x^2 - .05e^{u_{3_o}}\cos(x), \quad u_{5_o} + u_{2_o} signum(|x-10|), \quad -0.05u_{1_o}e^{u_{3_o}}\cos(x), \\ & 0.5x\sin(u_{4_o}) signum(|x-10|), \quad u_{2_o}\ ]\end{aligned} \tag{5.6}$$

where u_{i_o} represents the i^{th} effector position at the point of the gradient evaluation. Case B employs a midpoint numerical differencing scheme in order to compute the linearized $\mathbf{B}$ matrix in real time as [119]

$$\begin{aligned}\mathbf{B}(x) &= \frac{\Delta \mathbf{g}}{\Delta \mathbf{u}} \\ \mathbf{B}(x) &= \frac{1}{2\delta}\{\mathbf{g}[x(t), \mathbf{u}(t) + \delta] - \mathbf{g}[x(t), \mathbf{u}(t) - \delta]\}\end{aligned} \tag{5.7}$$

where δ is chosen as 0.2. Figs. 5.13-5.14 depict Case A for the QCAT WLS approach, where the $\mathbf{B}$ matrix is computed analytically given the time varying effectivity mapping, and Figs. 5.15-5.16 depict Case B using the finite difference approach. In both of these cases for this example, performance is nearly equivalent for both the centralized QCAT WLS and DSPC distributed method, however there is consistently higher virtual control tracking error with DSPC and a resulting slightly higher L_2 on average. This data resulted as the best case using a Monte-Carlo simulation technique with varying DSPC parameters. Specifically, the T temperature parameter was found as a key trade-off configuration for DSPC which directly impacted this tracking bias and the quiescence of the effector response. Higher values of T resulted in less effector activity and smoother overall tracking with the exception of an increased consistent error. Lower values of T on the other hand, reduced the tracking bias but generated more activity in the effectors. This trade-off is discussed additionally in Ch. 7 as one challenge for employing DSPC, with the identification of best parameters for application. Figs. 5.17-5.18 demonstrates this example for three T settings at 0.35, 0.2 and 1.0 where tracking bias is traded for effector activity.

Employing an online finite difference scheme can be problematic in cases with measurement noise and other sources of numerical fluctuation as discussed by Bodson and Bolender in [16, 43], and hence a second nonlinear control effectivity mapping example is demonstrated ahead.

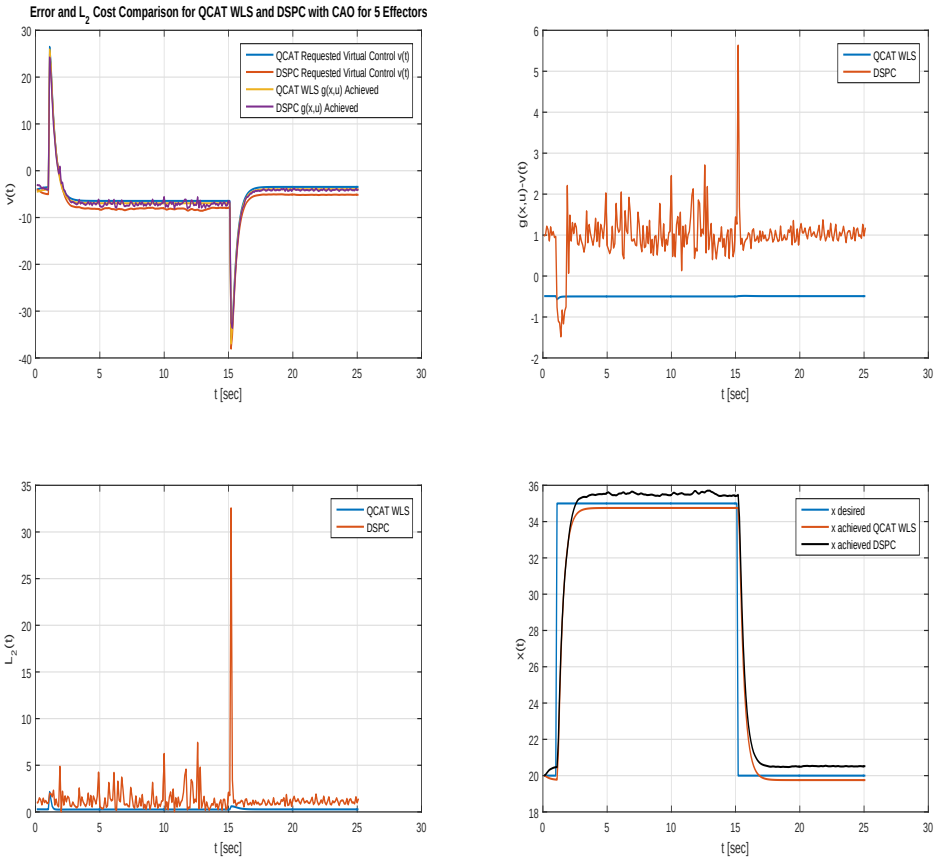


Figure 5.13: Nonlinear Example 1, Case A: Error and L_2 Cost Comparison for QCAT WLS and DSPC ($T = .35$) with CAO for 5 Effectors

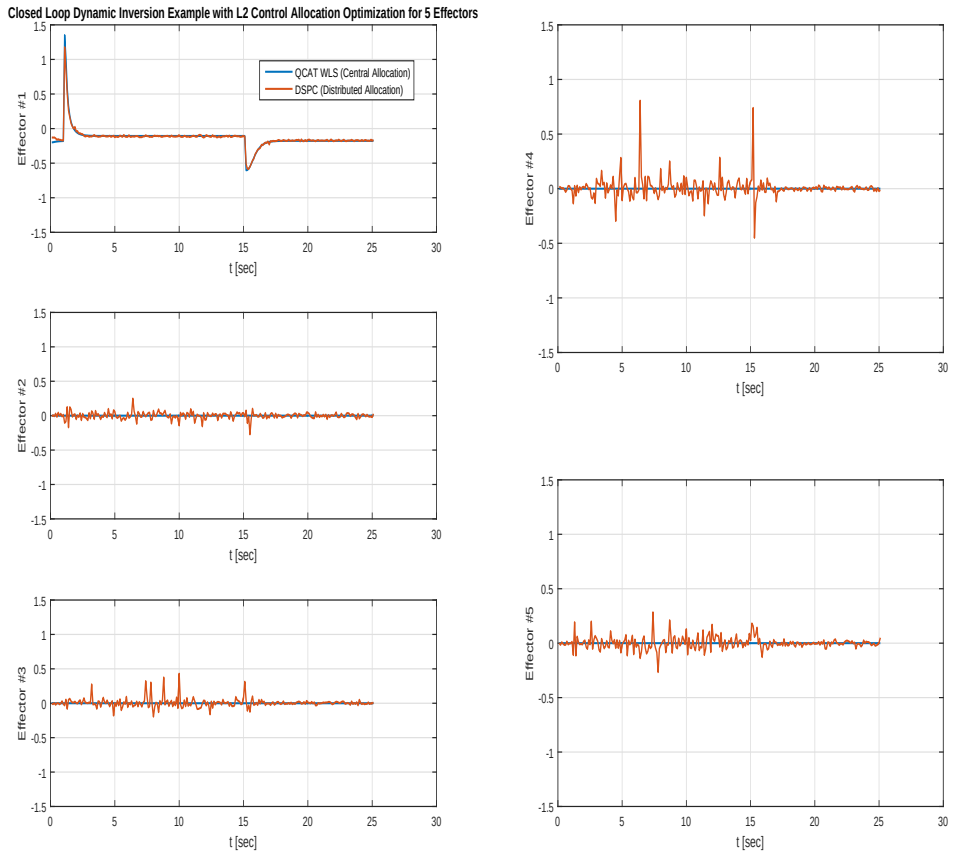


Figure 5.14: Nonlinear Example 1, Case A: Solution for Closed Loop Dynamic Inversion Example with L_2 Control Allocation Optimization for 5 Effectors

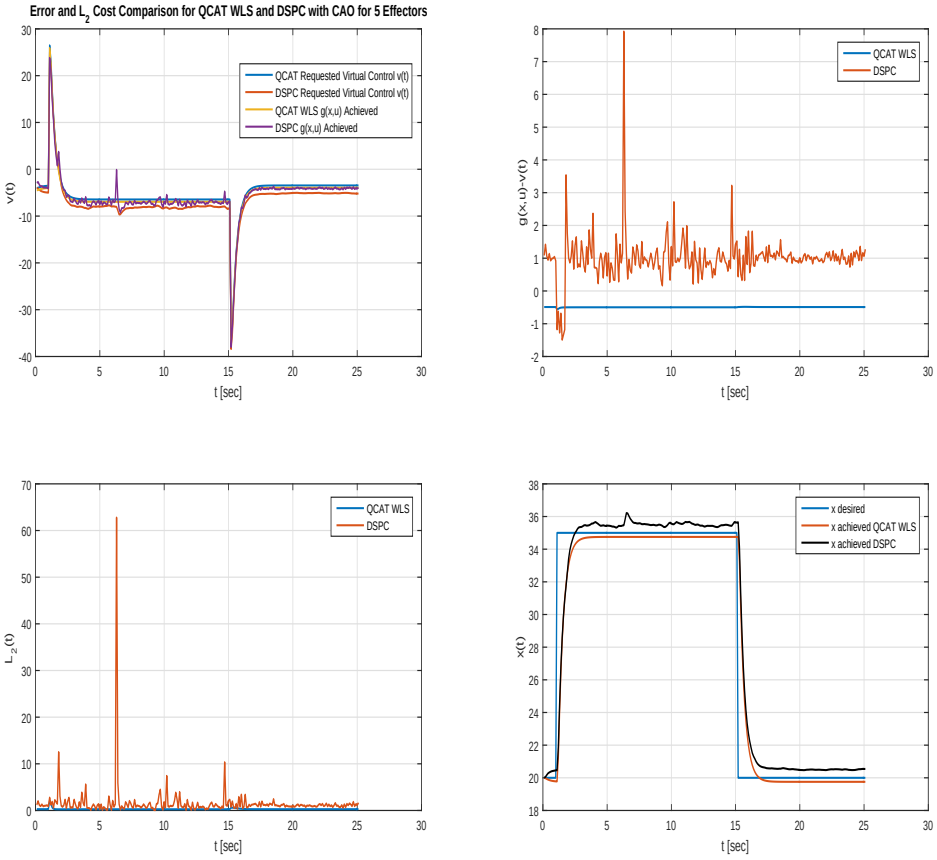


Figure 5.15: Nonlinear Example 1, Case B: Error and L_2 Cost Comparison for QCAT WLS and DSPC ($T = .35$) with CAO for 5 Effectors

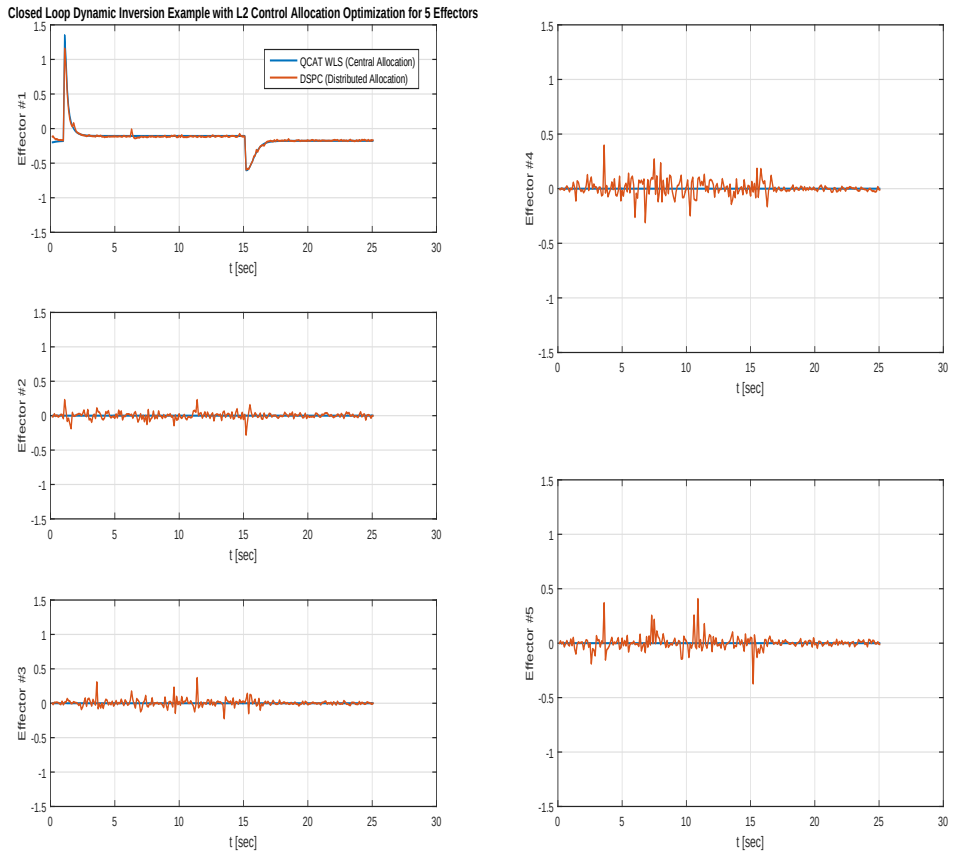


Figure 5.16: Nonlinear Example 1, Case B: Solution for Closed Loop Dynamic Inversion Example with L_2 Control Allocation Optimization for 5 Effectors

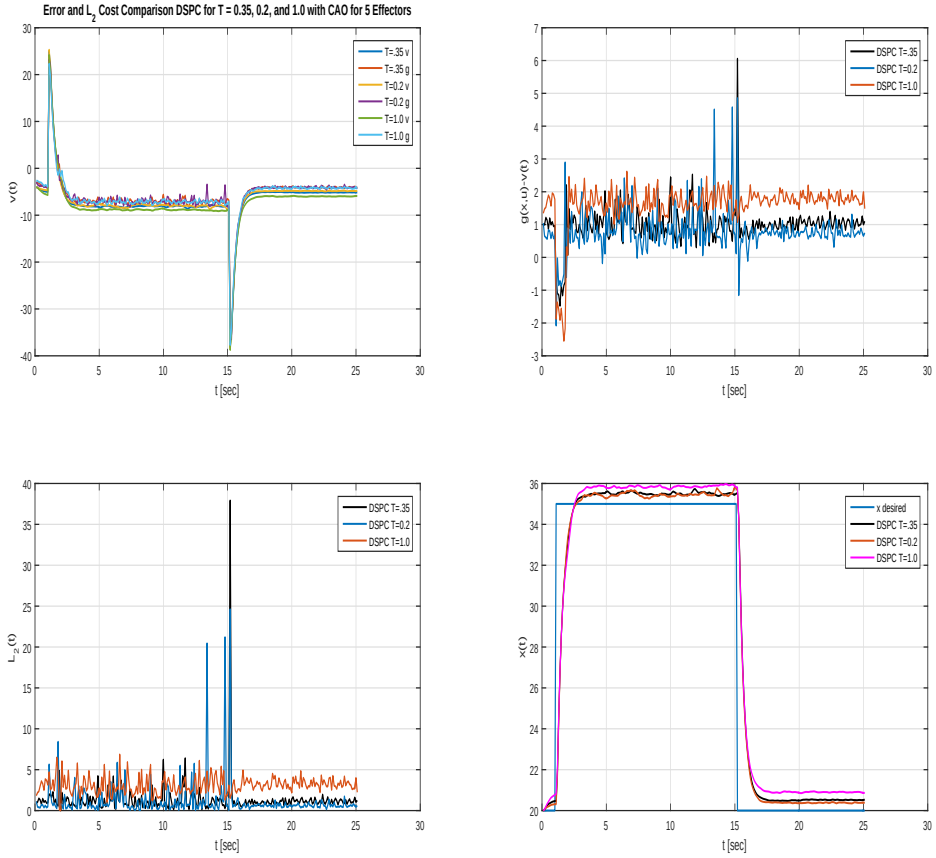


Figure 5.17: Nonlinear Example 1: Error and L_2 Cost Comparison with DSPC ($T = .35, 0.2, 1.0$) with CAO for 5 Effectors

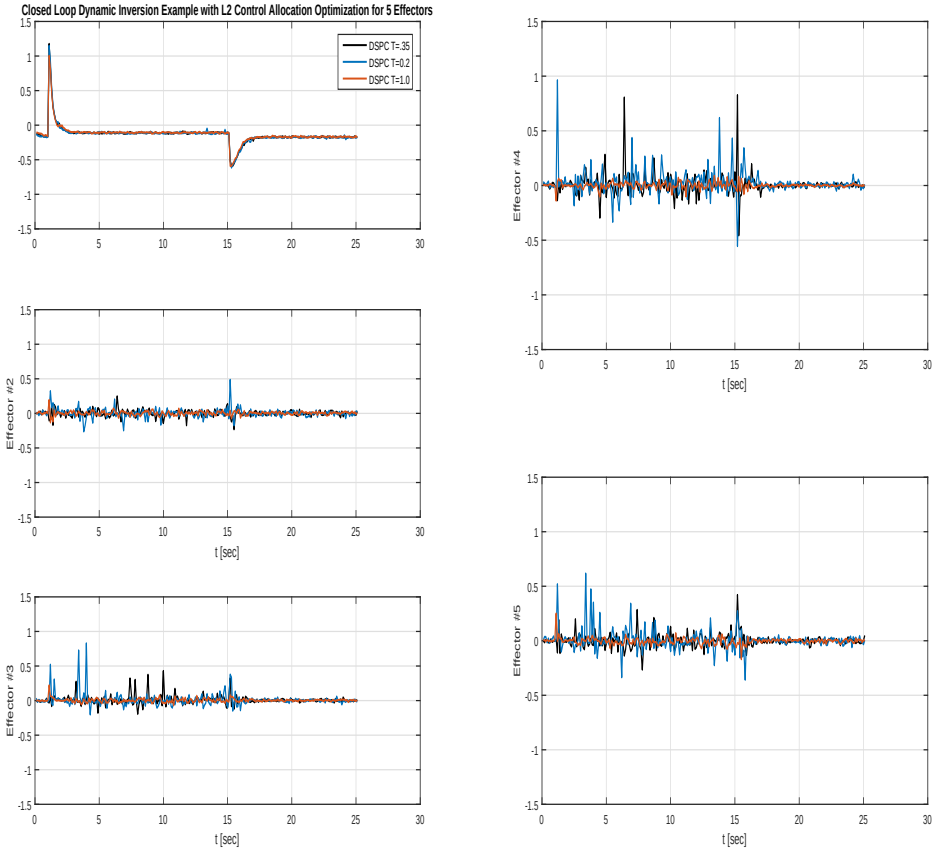


Figure 5.18: Nonlinear Example 1: DSPC ($T = .35, 0.2, 1.0$) Sensitivity: Solution for Closed Loop Dynamic Inversion Example with L_2 Control Allocation Optimization for 5 Effectors

5.3.2 Nonlinear Control Allocation Example 2

A second contrived nonlinear, non-affine control effectivity mapping, $\mathbf{g}[x(t), \mathbf{u}(t)]$ as described in Eq. (3.4), is created in this example as

$$\mathbf{g}[x(t), \mathbf{u}(t)] = .05u_1u_2u_3 + .5signum(|x - 10|)u_4 + .15u_4u_5 - .5u_5u_2 \tag{5.8}$$

For the Case A test, the analytic expression for $\mathbf{B}$ is derived as $\frac{\partial \mathbf{g}}{\partial \mathbf{u}}(x, \mathbf{u}_0)$

$$\mathbf{B}(x) = \begin{bmatrix} .05u_{2_o}u_{3_o} & .05u_{1_o}u_{3_o} - .5u_{5_o} & .05u_{1_o}u_{2_o} & .5signum(|x-10|) + .15u_{5_o} & .15u_{4_o} - .5u_{2_o} \end{bmatrix} \tag{5.9}$$

Clearly, this example involves highly nonlinear coupled control effectivity. This analytic expression for the linearized $\mathbf{B}$ matrix is evaluated in line in Case A with QCAT WLS. For DSPC, only the nonlinear control effectivity mapping in Eq. (5.8) is modeled at the agent scope. Parameters for this example matched the first nonlinear example with the exception of $\overline{\mathbf{u}} = \begin{bmatrix} 10 & 10 & 10 & 10 & 10 \end{bmatrix}'$, $\underline{\mathbf{u}} = -\overline{\mathbf{u}}$, and $T = .5$. Figs. 5.19-5.20 present resulting responses for the closed loop test for both allocation methods. QCAT WLS fails to stabilize the system due to the significant nonlinearities in Eq. (5.8), information that is lost in the analytic form of the linearized control matrix. On the other hand, the distributed DSPC approach evaluates $\mathbf{g}[x(t), \mathbf{u}(t)]$ locally at each agent using the Team Game world utility previously shown in Ch. 4 in Eq. (4.12) and successfully tracks the reference state command.

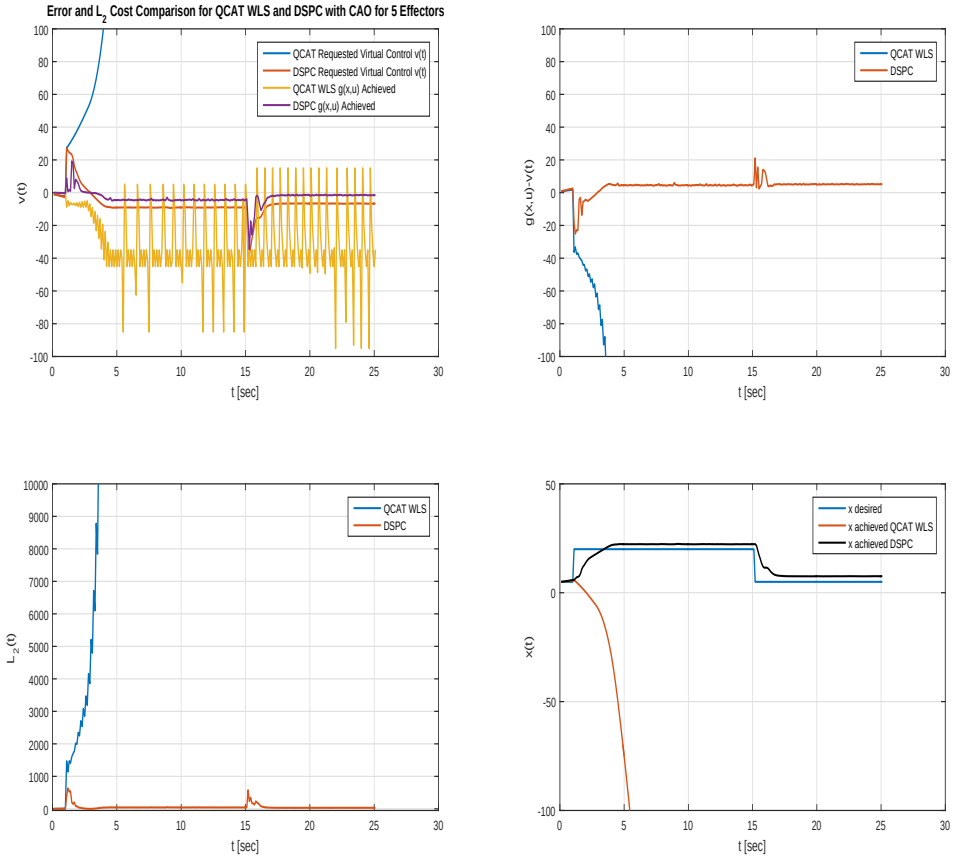


Figure 5.19: Nonlinear Example 2, Case A: Error and L_2 Cost Comparison for QCAT WLS and DSPC ($T = .5$) with CAO for 5 Effectors

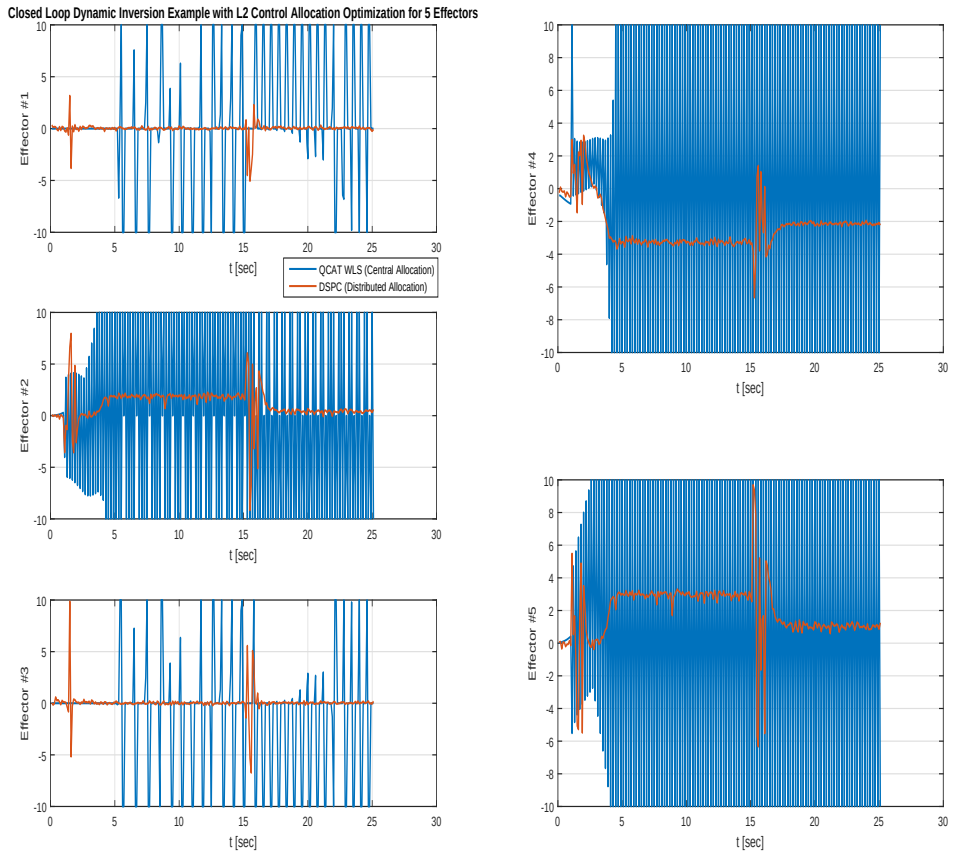


Figure 5.20: Nonlinear Example 2, Case A: Solution for Closed Loop Dynamic Inversion Example with L_2 Control Allocation Optimization for 5 Effectors

Often an analytic expression for the control effectivity or linearized mapping matrix is unknown, and Case B represents one approach, albeit potentially noisy and computationally expensive to formulate the full Jacobian matrix[4]. Figs. 5.21-5.22 depict Case B for both control allocation approaches with the second example in Eq. (5.8). Again, the centralized QCAT WLS allocation response exhibits significant stop-to-stop chatter in the effector response and fails to track the reference command. The distributed DSPC approach successfully determines the optimizing control solution using the nonlinear effectivity mapping.

[4]For example, a model of control effectivity may consist of a number of aerodynamic and propulsive lookup tables and may be available for evaluation. However, an equation describing this amalgamation may not be known, creating a challenge for realizing an analytic expression for the Jacobian per Case A. Case B assumes both DSPC and QCAT WLS have an equivalent ability to evaluate the model of the nonlinear control effectivity at a specific state and control position. However, QCAT WLS requires linearization via the finite difference equation for the first order derivative approximations. For evaluation fairness, a nonlinear optimization routine is used in the concluding example of this chapter.

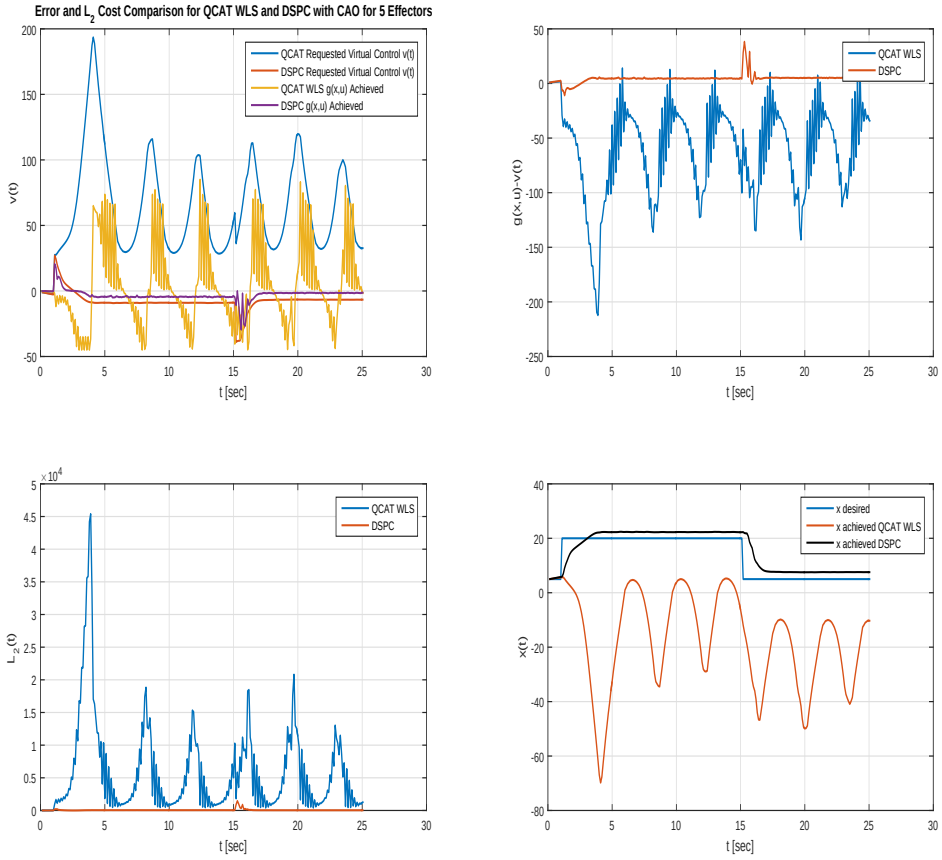


Figure 5.21: Nonlinear Example 2, Case B: Error and L_2 Cost Comparison for QCAT WLS and DSPC ($T = .5$) with CAO for 5 Effectors

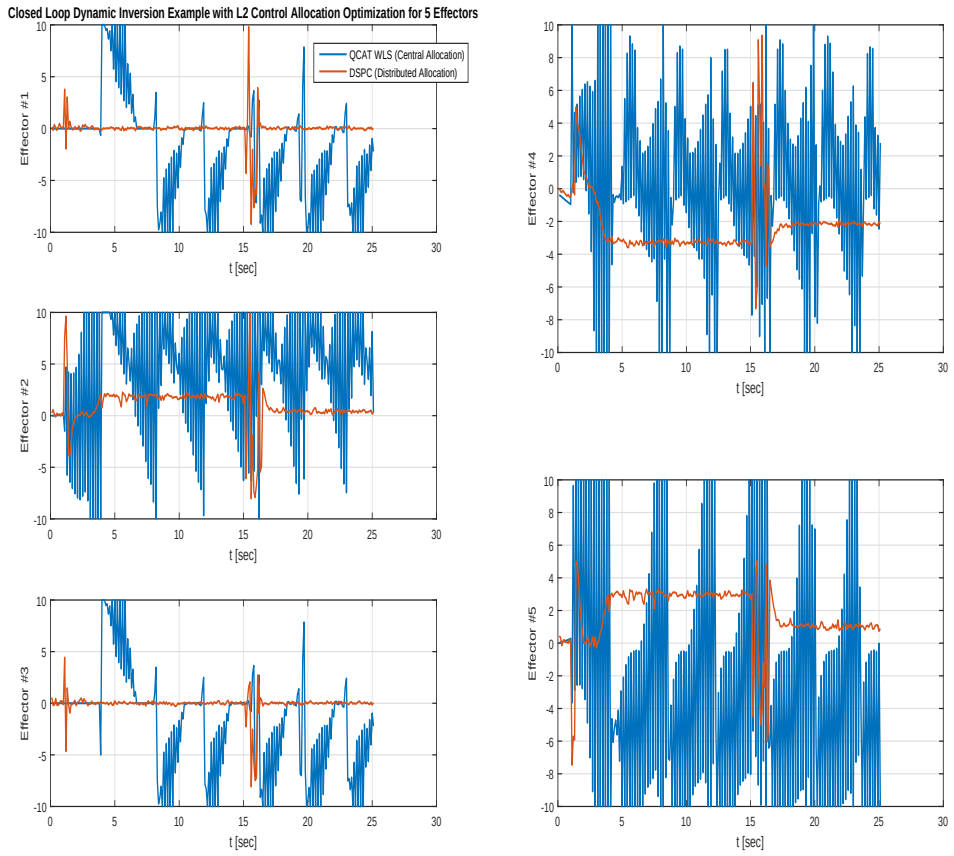


Figure 5.22: Nonlinear Example 2, Case B: Solution for Closed Loop Dynamic Inversion Example with L_2 Control Allocation Optimization for 5 Effectors

In all evaluation fairness, QCAT WLS is a linear control allocation toolbox and is expected to exhibit inferior performance in cases where nonlinearities are significant. Other optimization routines do exist beyond DSPC which can stabilize this system with effectivity mapping in Eq. (5.8). For example, Figs. 5.23-5.24 repeat Nonlinear Example 2 with the MATLAB nonlinear optimization toolset, fmincon, which uses an active-set (line search) method [120]. However, the fmincon approach does not promote a distributed architecture for control allocation, a clear benefit of the DSPC method for joint strategy optimization with agent product distributions.

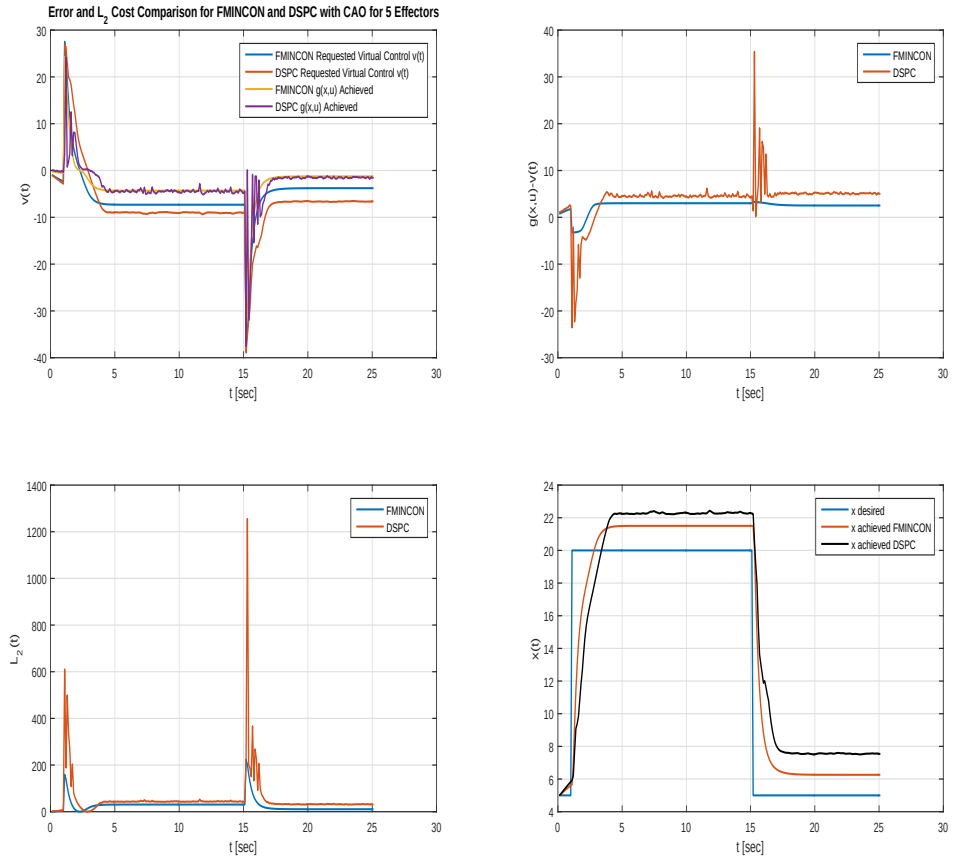


Figure 5.23: Nonlinear Example 2 with Fmincon: Error and L_2 Cost Comparison for QCAT WLS and DSPC ($T = .5$) with CAO for 5 Effectors

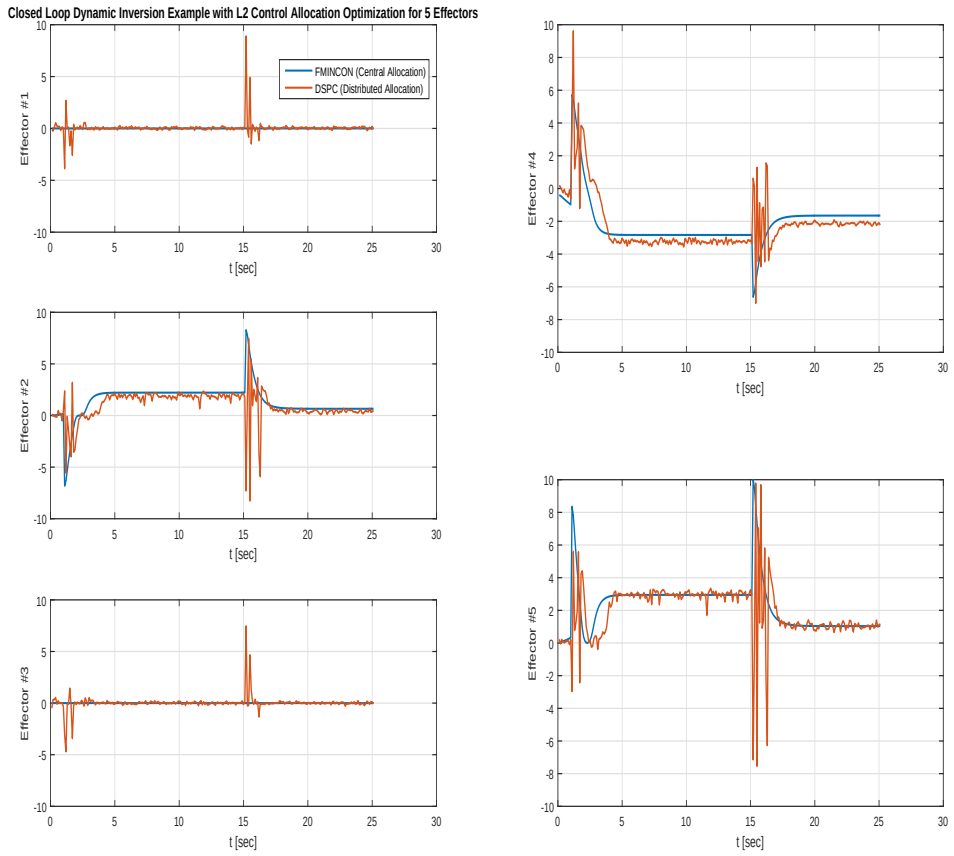


Figure 5.24: Nonlinear Example 2 with Fmincon: Solution for Closed Loop Dynamic Inversion Example with L_2 Control Allocation Optimization for 5 Effectors

A final simulation of Nonlinear Example 2 is provided in Figs. 5.25-5.26 with the MATLAB nonlinear optimization fmincon toolset as a centralized control allocation method versus DSPC. At simulation time 10 seconds, effector #4 fails to a locked position with failure case B observability in place as previously described. Fig. 5.25 shows the failed effector occurs during the steady state tracking of the initial step command, and both systems exhibit minor deviation. However, as the state tracking command steps at $t = 15$ seconds, the impact of the locked effector #4 becomes

apparent. After both systems respond to the new command, the effector solutions diverge at approximately $t = 20$ seconds, and the fmincon L_2 cost dramatically increases due to the onset of virtual command error.

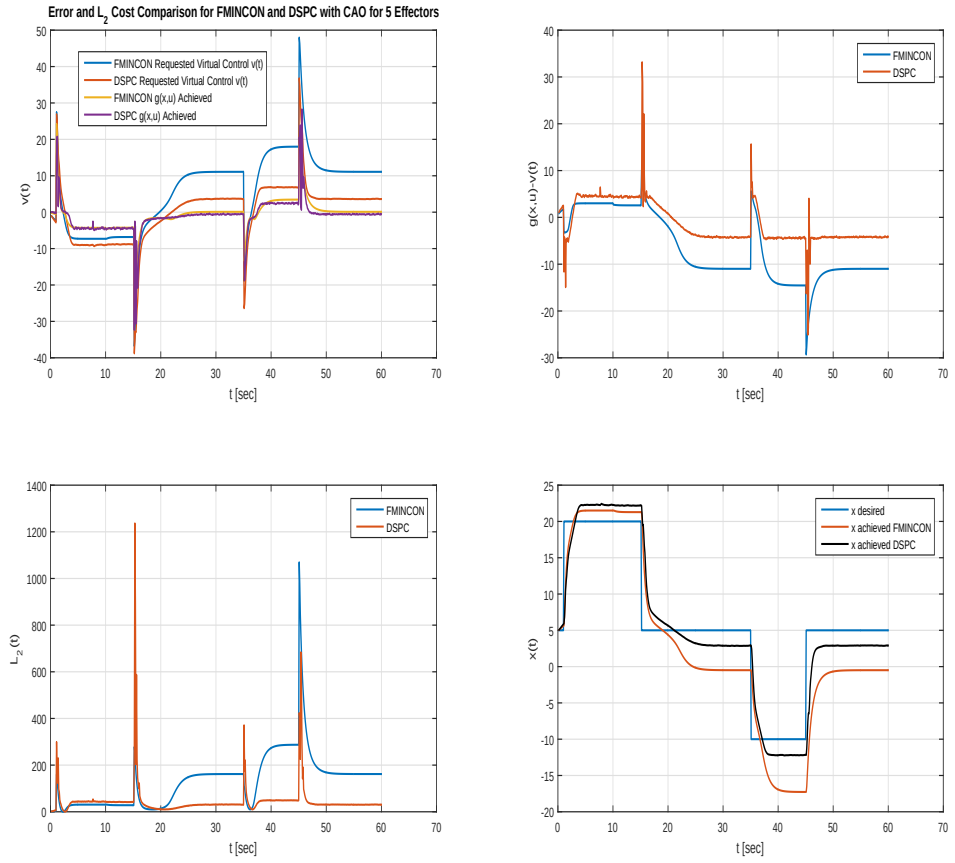


Figure 5.25: Failure Case for Nonlinear Example 2: Error and L_2 Cost Comparison for QCAT WLS and DSPC ($T = .5$) with CAO for 5 Effectors

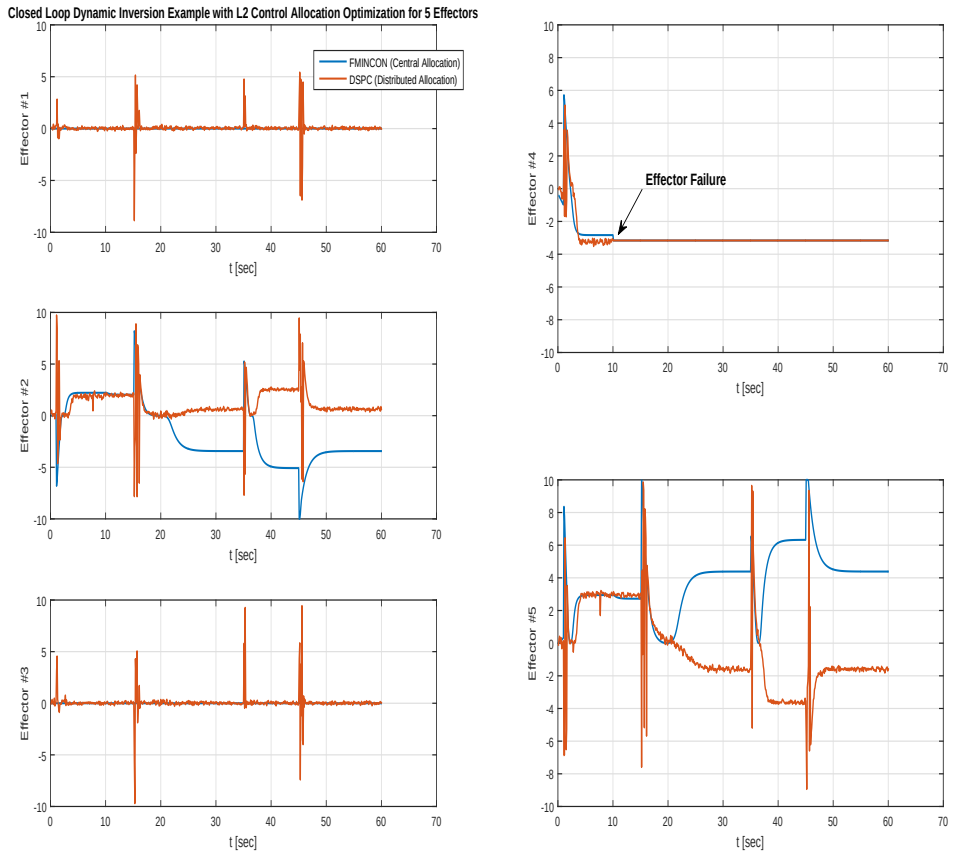


Figure 5.26: Failure Case for Nonlinear Example 2: Solution for Closed Loop Dynamic Inversion Example with L_2 Control Allocation Optimization for 5 Effectors

CHAPTER 6

DISTRIBUTED ALLOCATION FOR A HOVERCRAFT

This chapter presents a hypothetical hovercraft vehicle with 10 effectors (thrusters) and a dynamic inversion architecture with distributed control allocation. After describing the model of the vehicle and details on the mechanization of the control system, nominal and failure cases are presented, exhibiting the benefit of distributed allocation.

6.1 Hypothetical Hovercraft and Control Architecture

A nonlinear dynamic inversion control scheme is implemented in this chapter with a generic six degree of freedom hovercraft as an example application of distributed control allocation using DSPC. As provided in Ch. 2, the translational and rotational dynamics of the vehicle are given as

$$\dot{\mathbf{V}}_B = \mathbf{Rg} + \mathbf{S}(\omega_B)\mathbf{V}_B + \frac{1}{m}\sum_{j=1}^{N}\mathbf{F}j_B \tag{6.1}$$

$$\dot{\omega}_B = \mathbf{I_B}^{-1}\mathbf{S}(\omega_B)\mathbf{I_B}\omega_B + \mathbf{I_B}^{-1}\left[\sum_{j=1}^{N}\mathbf{S}(-\mathbf{d}_j)\mathbf{F}j_B + \tau j_B\right] \tag{6.2}$$

Reference App. A.1 for additional detailed derivations of the expansion of these equations for a hovercraft vehicle with multiple effectors. Four primary stabilization channels are chosen as critical channels to be controlled in the inner loop via distributed allocation, and consist of translational vertical acceleration and angular acceleration with $\mathbf{v}_d = [\dot{w}\ \dot{p}\ \dot{q}\ \dot{r}]^T \in \Re^4$. The hovercraft vertical acceleration, expressed in the

vehicle body frame, is given as the third component in the translational dynamics Eq. (6.1) and is expanded as

$$\dot{w} = -g\cos\phi\cos\theta + qu - pv + \frac{1}{m}\sum_{j=1}^{N} Fj_{z_B} \tag{6.3}$$

where g is the magnitude of gravity, ϕ and θ are Euler angles, and m is the mass of the vehicle with N total effectors aligned with the body z-axis creating thrust Fj_{z_B} at each j^{th} motor. The expansion of $\mathbf{S}(\omega_B)\mathbf{V}_B$ results in the translational velocity and rotational velocity coupled components, $qu - pv$, where $\mathbf{V}_B = [u\ v\ w]^T \in \Re^3$ and $\omega_B = [p\ q\ r]^T \in \Re^3$.

Clearly, after canceling the $f(x)$ nonlinearities in Eq. (6.3) and Eq. (6.2) as described in the dynamic inversion control design approach presented in Ch. 2, the control effectivity mapping for the four chosen stabilization channels is

$$g(u,x,t) = \begin{pmatrix} \frac{1}{m}\sum_{j=1}^{N} Fj_{z_B} \\ \mathbf{I_B}^{-1}\left[\sum_{j=1}^{N} \mathbf{S}(-\mathbf{d}_j)\mathbf{F}j_B + \tau j_B\right] \end{pmatrix} \tag{6.4}$$

This linear mapping is chosen intentionally in order to initially compare the distributed allocation performance to QCAT WLS, a linear routine as discussed at the conclusion of Ch. 5. Additional detail and derivation for this type of linear control effectivity mapping is available for a quadrotor in [121]. After comparison to QCAT WLS, a nonlinear control effectivity mapping is incorporated for evaluation with distributed allocation.

Figure 6.1 depicts the dynamic inversion architecture employed in this study with a distributed control allocation inner loop. Each agent works at the local actuator scope to effectively invert the modeled control mapping, g_i, to determine the required local command to achieve the global desired dynamics, given the perceived strategies of the other agents. These strategies are shared over a high rate bus as expected values of each agent's probably distribution. Multiple future research paths

are of interest with extending this approach and are discussed in the conclusion of this dissertation in Ch. 7. In brief, the local control effectivity mapping can include variability with respect to airframe location as well as estimation techniques for relaxing a priori modeling. Also, the network of agents in this work uses a complete graph of connectivity. Reducing this requirement may prove useful, especially in intuitive cases where a geometric neighborhood of agents are the dominant terms in an axis of interest and negligible for others. For example, in a fixed wing application, a distributed suite of outboard aileron surfaces dedicated to a roll channel may not require complete connectivity to a set of engines at the aft of the fuselage. Reduction of the agent connectivity requirement for distributed allocation must consider the network topology (governing agent adjacency matrix) and the impact on time to consensus. This item is discussed in further detail in App. F.1.

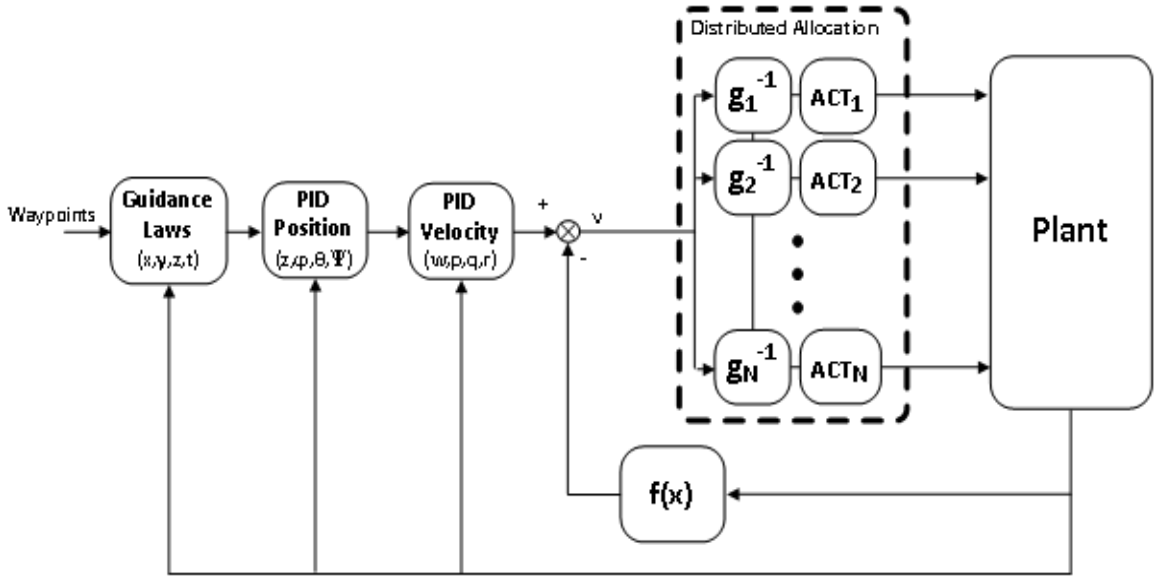


Figure 6.1: Hovercraft Nonlinear Dynamic Inversion Architecture with Distributed Allocation for N Total Effectors

The high rate iterative optimization in Fig. 6.1, captured in the distributed allocation partition, occurs deterministically with a specified maximum number of steps per execution. At the conclusion of the iterations, the command is sampled from the final form of the probability distribution and actuated at the local effector.

After the DSPC and QCAT WLS control allocation methods were implemented into the model environment, an initial validation exercise was conducted with a set of singlet responses for each respective channel for the inner loop buildup. Each singlet command was conducted with QCAT WLS and DSPC as the control allocation routines for comparison. All responses exhibit similar ability to track and for brevity have been reduced here to Fig. 6.2. Note the yaw channel single command was reduced to 0.5 deg/sec in magnitude to prevent effector saturation.

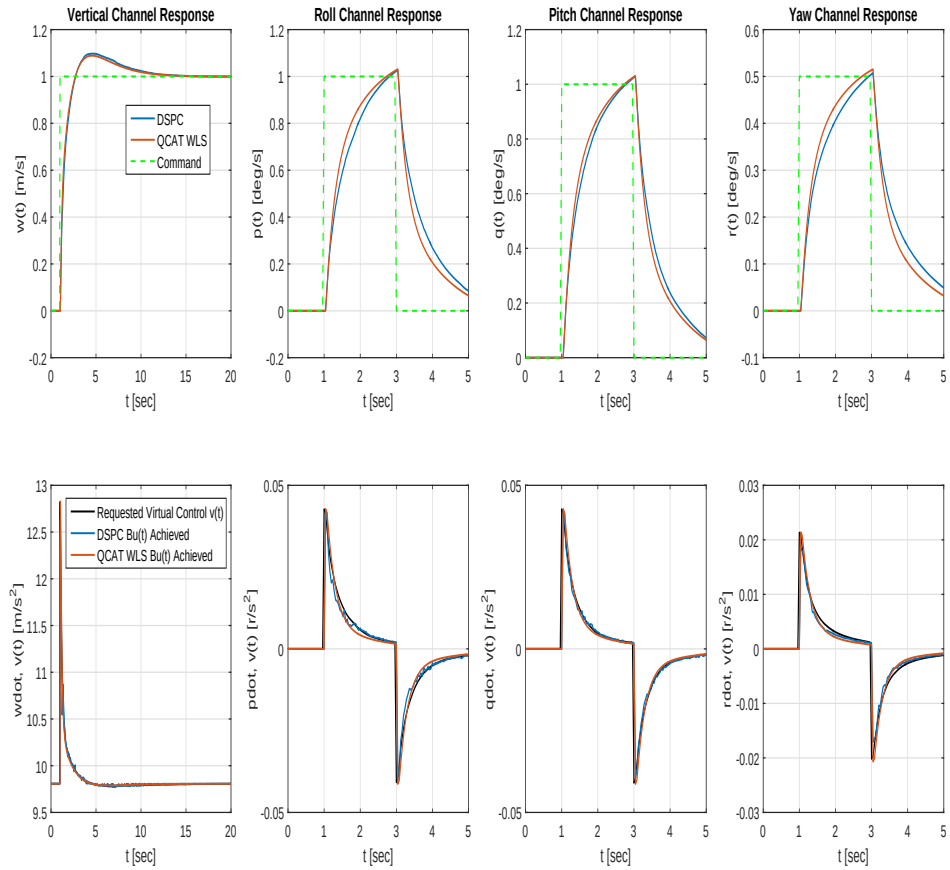


Figure 6.2: Singlet Responses for Hovercraft with Distributed Control Comparing QCAT WLS Centralized Control Allocation and DSPC Distributed Control Allocation

The remainder of the architecture shown in Fig. 6.1 is a traditional approach to establish command tracking loops on the velocity and position of the hovercraft. These outer loops employ classical proportional-integral-derivative (PID) control laws to establish the desired closed loop dynamical tracking behavior. The outermost loop is the mission guidance system which provides waypoint sequencing and height z, roll attitude ϕ, pitch attitude θ, and heading ψ reference commands to the position outer loop. The roll and pitch attitude guidance laws are synthesized by additional PID

controllers on forward and lateral velocity reference commands which are generated as a function of distance to the active waypoint in sequence.

A hypothetical multi-rotor hovercraft is modeled by geometrically distributing thrusters over a planar 2π rotation about the body z_b axis as depicted in Fig. 6.3. For the deca-rotor case with $N = 10$ total thrusters, engine 10 is placed at the positive x_b axis of the vehicle on the $x_b - y_b$ plane at a distance of $L = 1$ m. Neighboring engines are incrementally placed by sweeping with an angle of $\frac{2\pi}{N}$. Fig. 6.4 below presents a conceptual view of the hovercraft in operation with an isometric perspective.

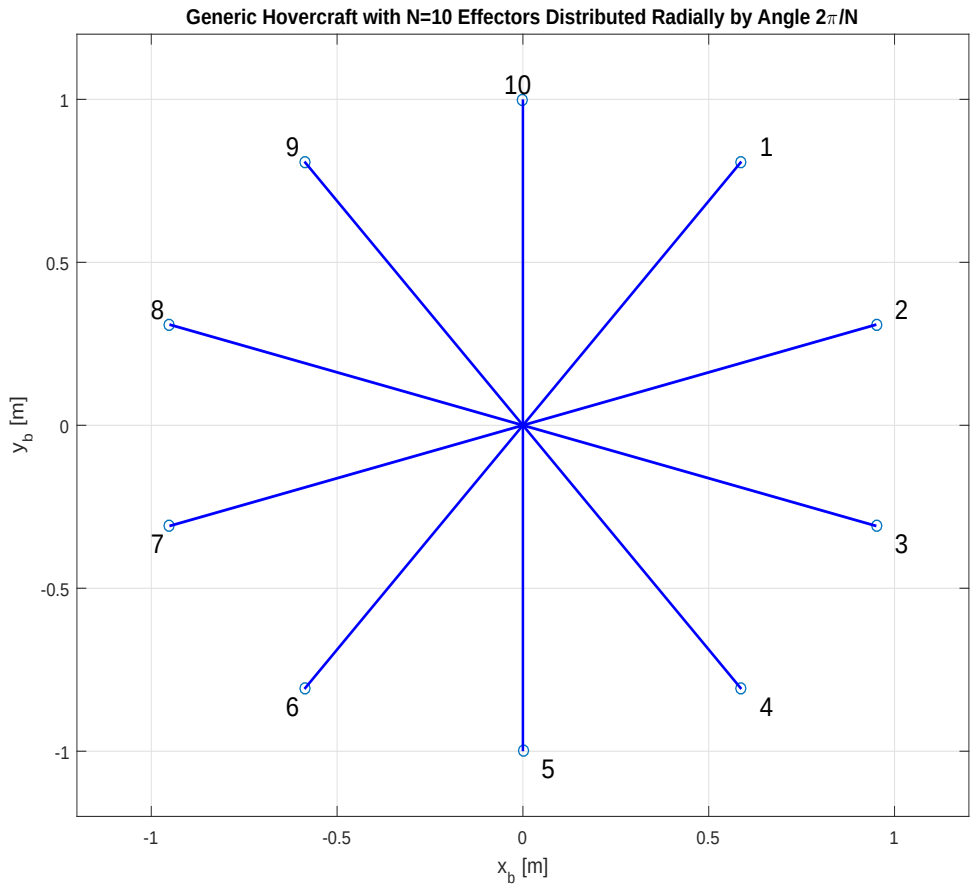


Figure 6.3: Generic Deca-Rotor Hovercraft with $N = 10$ Effectors Distributed Radially by Angle $\frac{2\pi}{N}$

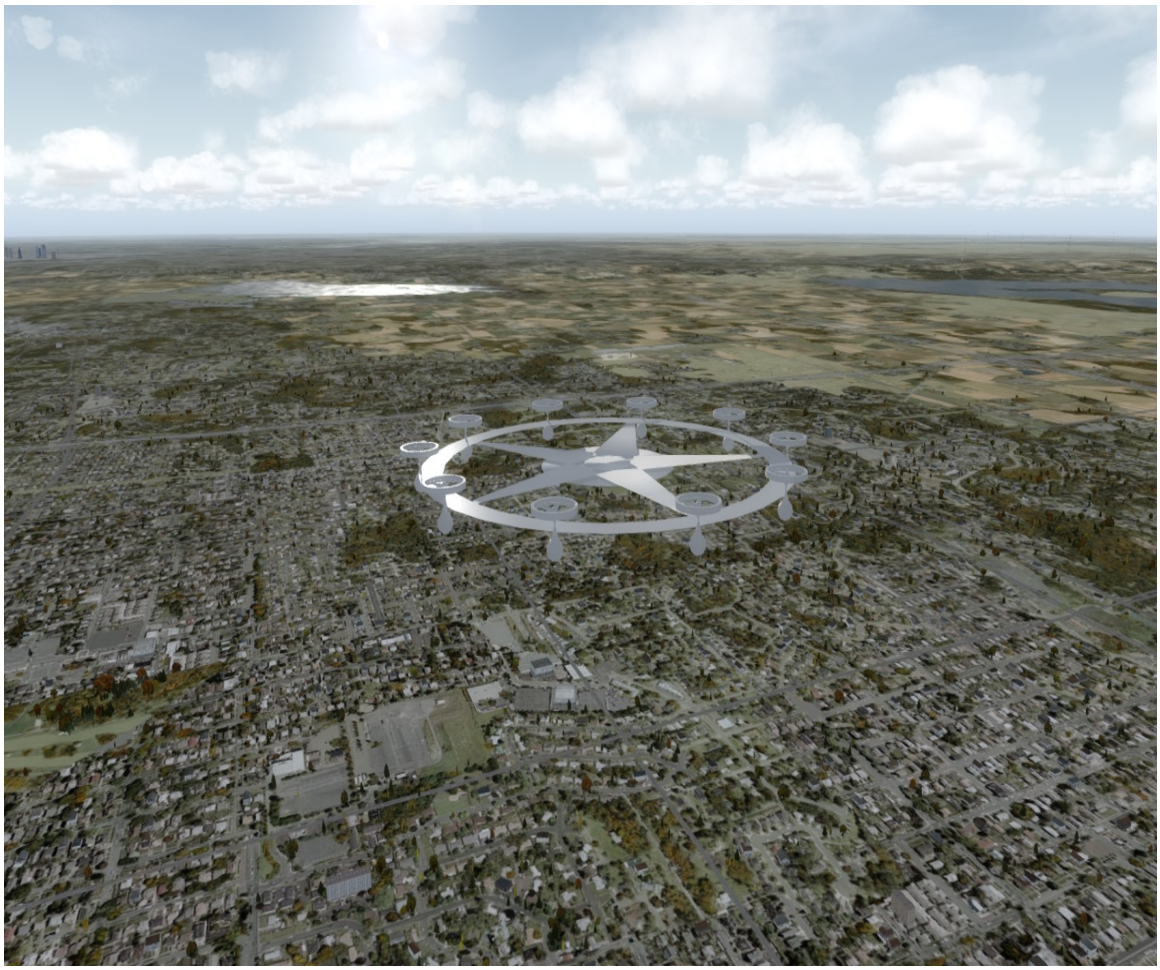


Figure 6.4: Isometric View of Deca-Rotor Hovercraft with $N = 10$ Effectors Distributed Radially by Angle $\frac{2\pi}{N}$

The deca-rotor hovercraft is configured symmetrically with the center of mass at the origin of Fig. 6.3. The selected mass and inertia properties for the vehicle are presented in Table 6.1 below. The thrust response of each engine is modeled as a first order transfer function of the form, $\frac{1}{\tau_E s+1}$ with a time constant $\tau_E = .025$ seconds. Finally, a force-to-moment scaling factor, c, is modeled per the approach in [121] to linearly map a specified thrust to the torque required to spin the propeller to generate the specified thrust. The torque turning the propeller at the given speed is

also exerted on the vehicle as an applied moment. Neighboring propellers are modeled as counter-rotating to create an overall trim balance of torque. This linear torque relation is modeled for each of the j^{th} effectors providing a thrust magnitude aligned with the hovercraft body z_b axis, Fj_{z_B}, as

$$\tau j_B = c(-1)^j F j_{z_B} \tag{6.5}$$

Table 6.1: Hypothetical Deca-Rotor Parameters.

Parameter	**Variable**	**Value**
Mass	m	0.1 kg
Roll Inertia	I_{xx}	0.02 kg-m^2
Pitch Inertia	I_{yy}	0.03 kg-m^2
Yaw Inertia	I_{zz}	0.05 kg-m^2
Thrust Time Constant	τ_E	.025 sec
Lever Arm Distance from Center of Mass to Motor	L	1 m
j^{th} Propeller Torque Coefficient	c	$.05(-1)^j$ m

The full state vector of the hovercraft vehicle is $\mathbf{x} = [u\ v\ w\ p\ q\ r\ \phi\ \theta\ \psi\ x\ y\ z\ ]^T \in \Re^{12}$. With the assumption that these states are observable and available as outputs to be controlled, the $\mathbf{y}$ desired vector of outputs for tracking are chosen as $[w\ p\ q\ r]^T \in \Re^4$ yielding $\mathbf{C} = \mathbf{1}^{4\times 12}$ as the output matrix with ones only populating matrix elements in columns corresponding to the mapped state to yield $\mathbf{y} = \mathbf{Cx}$. Evaluation of Eq. (6.4) with the vehicle properties in Table (6.1) and linearization at a steady state hovering flight condition with $\mathbf{x_0} = \mathbf{0} \in \Re^{12}$ generates the linear control effectivity $\mathbf{CB} \in \Re^{4\times 10}$ matrix as

$$\mathbf{CB} = \begin{bmatrix} 10.0000 & 10.0000 & 10.0000 & 10.0000 & 10.0000 & 10.0000 & 10.0000 & 10.0000 & 10.0000 & 10.0000 \\ -29.3893 & -47.5528 & -47.5528 & -29.3893 & -0.0000 & 29.3893 & 47.5528 & 47.5528 & 29.3893 & 0.0000 \\ 26.9672 & 10.3006 & -10.3006 & -26.9672 & -33.3333 & -26.9672 & -10.3006 & 10.3006 & 26.9672 & 33.3333 \\ 1.0000 & -1.0000 & 1.0000 & -1.0000 & 1.0000 & -1.0000 & 1.0000 & -1.0000 & 1.0000 & -1.0000 \end{bmatrix}$$

Intuitively, the top row of the linear matrix indicates that symmetric thrust for all engines leads to an increase in height rate. Row two reflects the differential thrust between port and starboard engines, where the difference in thrust between engines $1-2-3-4$ and $6-7-8-9$ leads to a rolling moment and $\dot{p}$ contribution with respect to the body x_b axis as shown in Fig. 6.3. Engine number 10 and 5 have zero impact on roll due to alignment on the body x_b axis. Likewise for pitch control, row three of the B matrix shows the thrust differential between $1-2-8-9-10$ and $3-4-5-6-7$ contributes to $\dot{q}$ due to the geometric placement and lever arm distance with respect to the body y_b axis. Finally, yaw control power is established solely through the torque contribution due to each individual propeller spinning in either the clockwise or counter-clockwise rotational direction, as indicated by the sign in row four of the **CB** matrix.

With four acceleration channels now for the control allocation problem, the acceleration penalty matrix $\mathbf{W_v} \in \Re^{4\times4}$ as defined earlier in Ch. 3 in Eq. (3.11) is set as the following diagonal matrix

$$\mathbf{W_v} = \begin{bmatrix} Wv_{\dot{w}} & 0 & 0 & 0 \\ 0 & Wv_{\dot{p}} & 0 & 0 \\ 0 & 0 & Wv_{\dot{q}} & 0 \\ 0 & 0 & 0 & Wv_{\dot{r}} \end{bmatrix}$$

where $Wv_{\dot{w}}$, $Wv_{\dot{p}}$, $Wv_{\dot{q}}$, and $Wv_{\dot{r}}$ penalizes acceleration error on the vertical, roll, pitch, and yaw channel, respectively. The thruster use penalty matrix, $\mathbf{W_u} \in \Re^{10\times10}$, is also diagonal with an identical weighting term per engine, Wu_j as provided by Table (6.2). Additionally, Table (6.2) lists the thruster desired values u_{d_j} and the total L_2 acceleration error penalty, γ_{L_2}.

Table 6.2: L_2 Control Allocation Optimization Weights.

Parameter	**Variable**	**Value**
Vertical Acceleration Error Penalty	$Wv_{\dot{w}}$	200
Roll Acceleration Error Penalty	$Wv_{\dot{p}}$	100
Pitch Acceleration Error Penalty	$Wv_{\dot{q}}$	200
Yaw Acceleration Error Penalty	$Wv_{\dot{r}}$	2000
Thruster Use Penalty	Wu_j	1.0
Thruster Desired Value	u_{d_j}	0.0
Effector Constraint Penalty	λ_{L_2}	0, 1e9
L_2 Acceleration Error Penalty	γ_{L_2}	1.0

The L_2 control allocation optimization problem is solved with both the Quadratic Control Allocation Toolbox using Weighted Least Squares (QCAT WLS) provided by Härkegård in [18] and distributed allocation with Delayed Sampling Probability Collectives (DSPC). DSPC is configured for the deca-rotor with 10 distributed agents, $n_{agent} = 10$, and $T = 0.5$, $\alpha = 0.001$, $\gamma = 0.03$, with a stopping condition at 15 maximum iterations as provided in Table (6.3). The j^{th} agent mixed strategy $\mathbf{q_j}(\mathbf{x_j})$ used $n_{moves_j} = 1000$ bins with $m = 20$ samples drawn per iteration. A numerical trim solution is used to initialize all engines to a thrust value to balance the weight of the vehicle in level hovering flight at simulation time $t = 0$. Each agent in DSPC is initialized with this trim solution, u_0, and the effector limits for DSPC are set to bound u_0 with $\overline{\mathbf{u}} = \begin{bmatrix} 0.5 & \cdots & 0.5 \end{bmatrix}'$ and $\underline{\mathbf{u}} = -\overline{\mathbf{u}}$ as search domain limits for the optimal thrust solution.

Table 6.3: DSPC Parameter Settings.

Parameter	**Variable**	**Value**
Number of Agents	n_{agent}	10
Maxent Lagrangian Temperature	T	0.5
Probability Update Gain	α	0.001
Data Aging Gain	γ	.03
Samples Drawn Per Iteration	m	20
Iterations Per Frame	n_{step}	15
Moves Per Agent	n_{moves_j}	$1e3$
Agent Effector Limit Constraint Gain	η	0.1
Agent Effector Upper Limit	$\overline{\mathbf{u}}$	$\mathbf{0.5}$
Agent Effector Lower Limit	$\underline{\mathbf{u}}$	$-\overline{\mathbf{u}}$

6.2 Linear Control Effectivity, Nominal Case

After takeoff and climb to 1 m of altitude, the deca-rotor is commanded to four waypoints in a racetrack pattern beginning with a translate left to waypoint 2, to a forward translation to waypoint 3, and a translate right leg to waypoint 4. Finally, a backwards translation brings the hovercraft to the initial position at waypoint 5 prior to translation. Waypoints are programmed with three dimensional desired positions as well as desired heading and time of acquisition as shown below in Table (6.4).

Table 6.4: Deca-rotor Flight Profile #1.

Waypoint ID	x [m]	y [m]	z [m]	ψ [deg]	t_a [sec]
1	0.0	0.0	1.0	0.0	5.0
2	-5.0	0.0	1.0	0.0	1.0
3	-5.0	5.0	1.0	0.0	1.0
4	0.0	5.0	1.0	0.0	1.0
5	0.0	0.0	1.0	0.0	1.0

Figure 6.5 compares the achieved trajectories using QCAT WLS as a centralized control allocation method and DSPC as a distributed control allocation method for the inner loop stability augmentation system. In this case where all motors behave

nominally throughout the hovering mission, the trajectories correspond closely. After takeoff from the initial condition shown at $(0, 0, 0)$ m for x, y, and z, both QCAT WLS and DSPC control allocation approaches complete the profile. Fig. 6.6 depicts the time history form of the inertial translational and rotational position with reference commands. Other than a subtle difference in the recorded heading response of the vehicle (with magnitudes less than .01 deg), the responses are nearly identical, demonstrating the distributed allocation approach successfully mimics the QCAT WLS method.

The translational and rotational velocity vectors, expressed in the body frame of the hovercraft, are presented in Fig. 6.7. Similar to the position data, responses between the two allocation methods are nearly identical with the exception of the pitch rate and yaw rate channel, where DSPC exhibits some higher frequency content due to increased effector activity, potentially mitigated with additional DSPC parameter optimization to reduce agent strategy variances at the conclusion of each DSPC instance[1]. Magnitudes for this difference are subtle for the yaw channel at less than 0.02 dps. The pitch channel however is slightly noisier with peak differences in q at approximately 0.5 dps.

The virtual control command request and actual response is presented in Fig. 6.8 for both DSPC and QCAT WLS. Clearly the source of higher frequency noise resides in this inner loop acceleration channel where DSPC searches through the optimization landscape. This response is also evident in Fig. 6.9 with the effector data for all 10 motors. After the initial transient to establish altitude after takeoff, average thrust remains at approximately 60% throughout the mission entirety. Still, with only $m = 20$ samples per iteration for 15 steps, the DSPC distributed allocation method successfully stabilizes the deca-rotor and completes the mission. In future work,

[1]Reference DSPC T sensitivity and parameter optimization discussion in Ch. 5

DSPC parameters listed in Table (6.3) may be further optimized to quiet effector response to the level of QCAT WLS.

Finally, Fig. 6.10 presents the L_2 objective cost as a function of time over the mission duration. For legibility, the left column provides the first two seconds of time history data due to the initial climb transient after takeoff while the right column zooms to the remainder of the mission profile. DSPC accomplishes a similar average cost to QCAT WLS with the exception of the higher frequency behavior evident in the effector responses. Overall, the DSPC distributed allocation method accomplishes inner loop stability for the deca-rotor and performs sufficiently, enabling mission waypoint completion via the velocity and position tracking loops shown previously in Fig. 6.1.

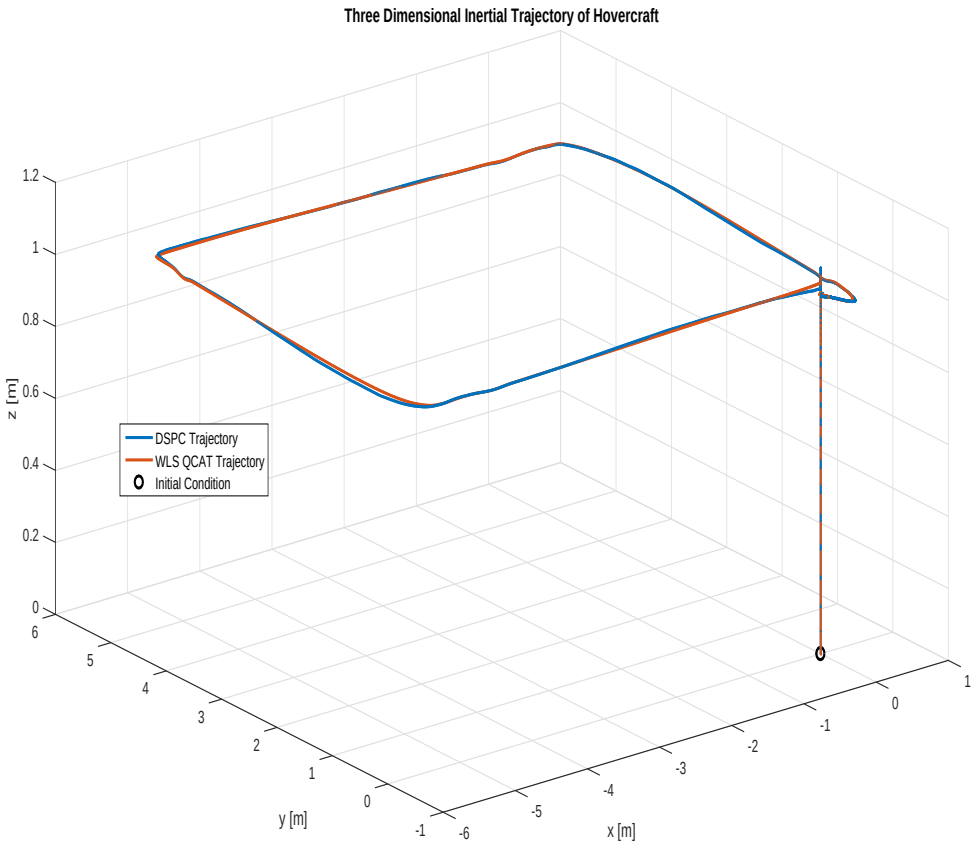


Figure 6.5: Nominal Case: Three Dimensional Inertial Trajectory for Deca-Rotor Hovercraft Comparing QCAT WLS Centralized Control Allocation and DSPC Distributed Control Allocation

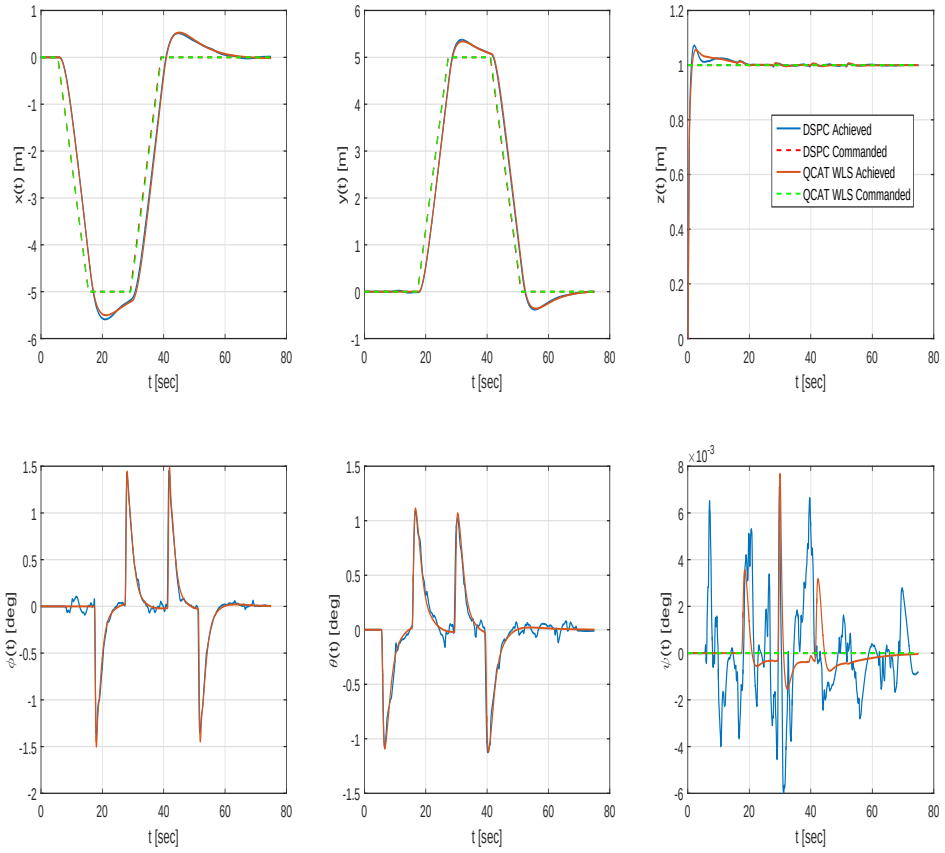


Figure 6.6: Nominal Case: Inertial Position Time History for Deca-Rotor Hovercraft Comparing QCAT WLS Centralized Control Allocation and DSPC Distributed Control Allocation

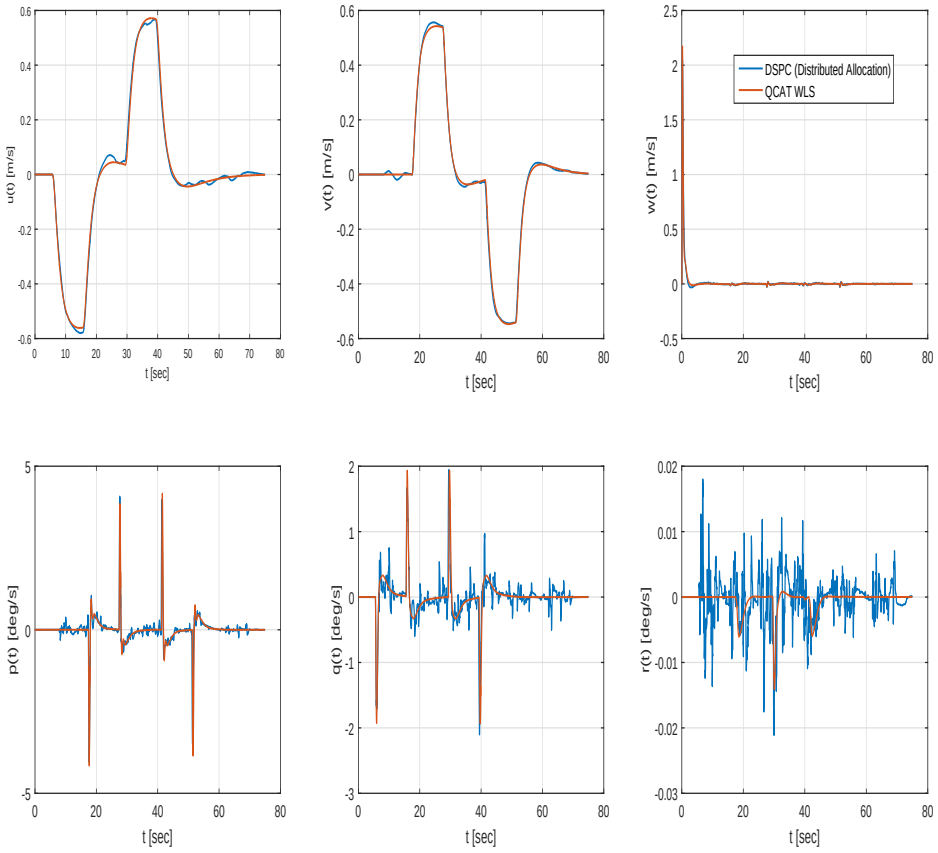


Figure 6.7: Nominal Case: Body Frame Velocity for Deca-Rotor Hovercraft Comparing QCAT WLS Centralized Control Allocation and DSPC Distributed Control Allocation

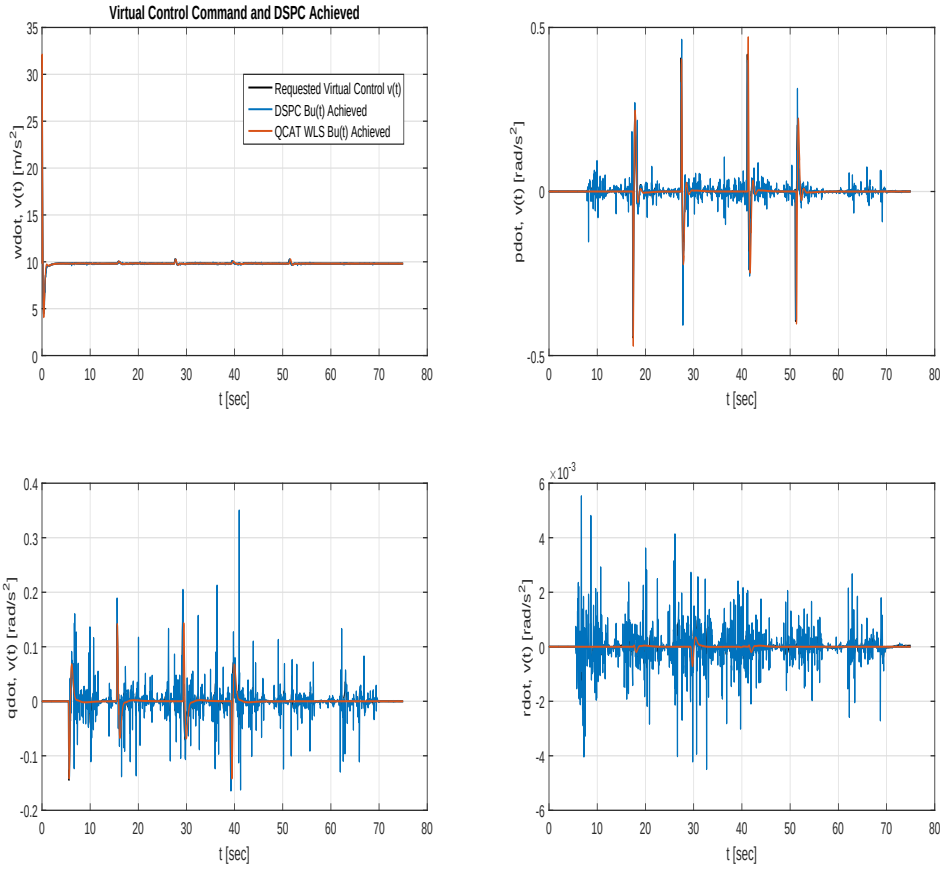


Figure 6.8: Nominal Case: Virtual Control Performance Using Distributed Control for Deca-Rotor Hovercraft Comparing QCAT WLS Centralized Control Allocation and DSPC Distributed Control Allocation

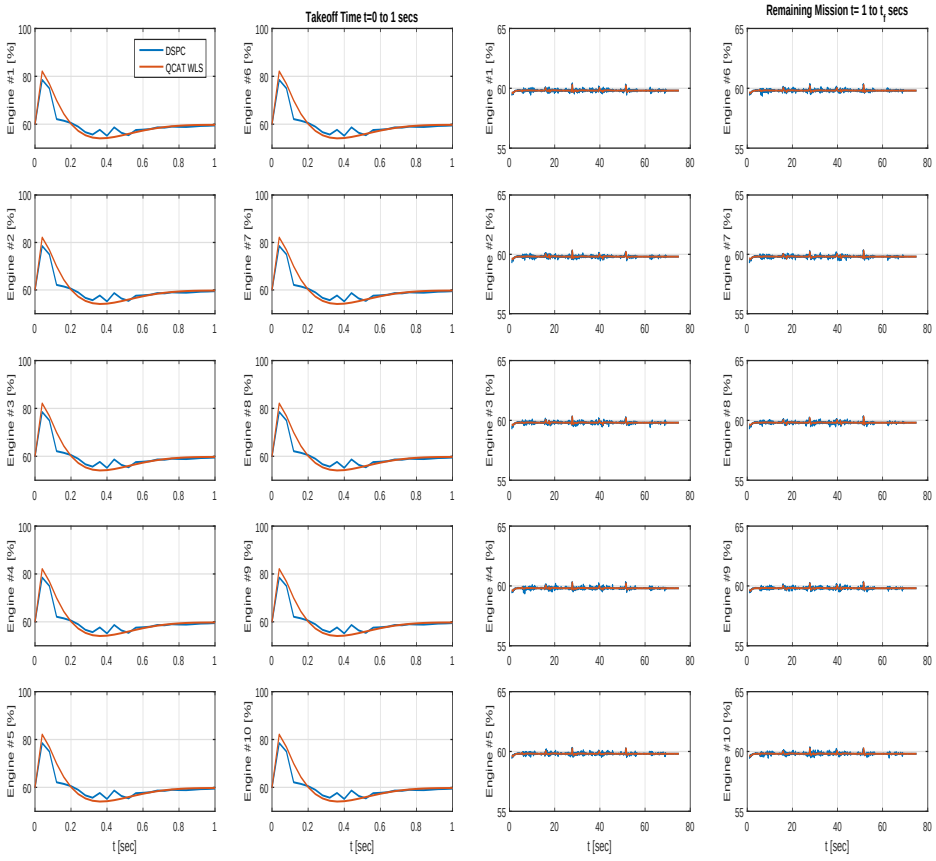


Figure 6.9: Nominal Case: Effector Thrust Levels Using Distributed Control for Deca-Rotor Hovercraft Comparing QCAT WLS Centralized Control Allocation and DSPC Distributed Control Allocation

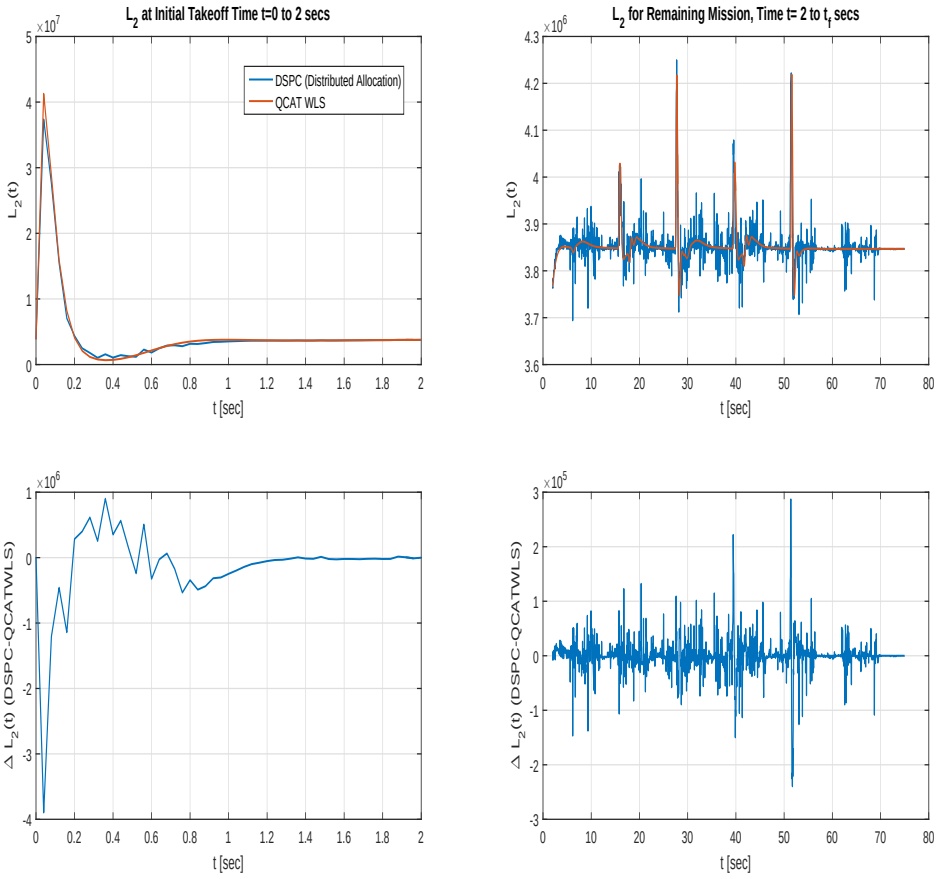


Figure 6.10: Nominal Case: L_2 Objective Values Using Distributed Control for Deca-Rotor Hovercraft Comparing QCAT WLS Centralized Control Allocation and DSPC Distributed Control Allocation

Figures 6.11, 6.12, 6.13 present the agent strategies for DSPC during the initial takeoff segment of the mission at $t = 0$ seconds for $k = 1, 2, 15$, respectively. After initialization as uniform distributions at $k = 1$, the optimization iterations rapidly provide a substantial increase in the probability bin heights and reduction in variance[2]. Fig. 6.14 depicts a time history view of the evolution of the expected value of each strategy over multiple optimizations. The red line indicates the reset point in between optimizations with 15 steps per optimization. These steps are set to deterministically align with the base execution rate of the simulation at 25 hz, and the final expected value of the strategy is sampled and down-rated as the actuator command to the engine[3]. In this evolution, each expected value initializes at zero due to the initial symmetric uniform distribution, and then evolves to the minimizing L_2 solution. Likewise, in Fig. 6.15, the standard deviation is shown for the multiple optimizations over the equivalent time history during the takeoff segment. In this data, the standard deviation initializes high at the red line, again indicative of the initial uniform distribution, and then reduces throughout the optimization in some cases. Cases exhibiting an increase in standard deviation indicate a need for additional DSPC parameter optimization or lack of sufficient convergence and should be investigated in future work.

[2]While the rapid change in probability bin height from $k = 1$ to $k = 2$ indicates early convergence, this response also may be indicative of too high of a DSPC update gain α and will be explored in future work.

[3]The DSPC distributed control allocation toolbox constructed in this work employs a high rate for loop with a deterministic 15 step iteration within the 6DOF (Six Degree of Freedom) hovercraft simulation executing at 25 hz. For real time deployment, each agent must process the local 15 step optimization within the time window corresponding to the allocation execution rate. Therefore, in this hovercraft application, $\Delta T_{max} = .04$ secs.

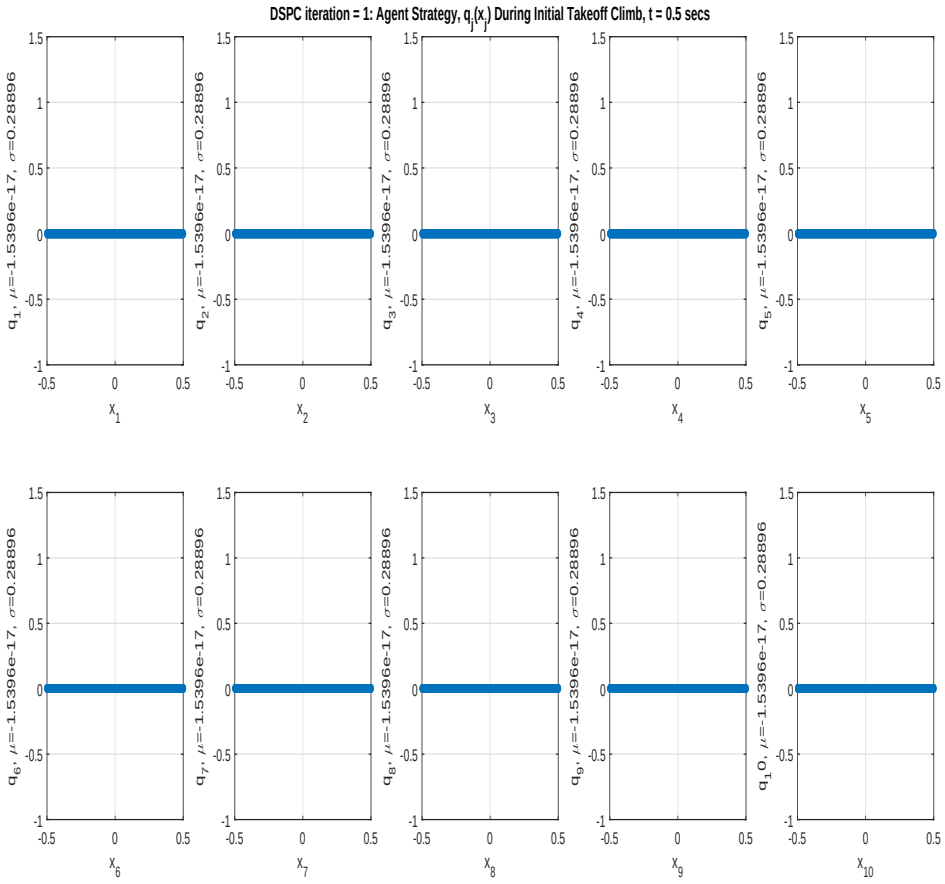


Figure 6.11: Nominal Case: DSPC Agent Strategy, $\mathbf{q}(\mathbf{x})$ at $k = 1$, During Initial Hovercraft Takeoff Climb

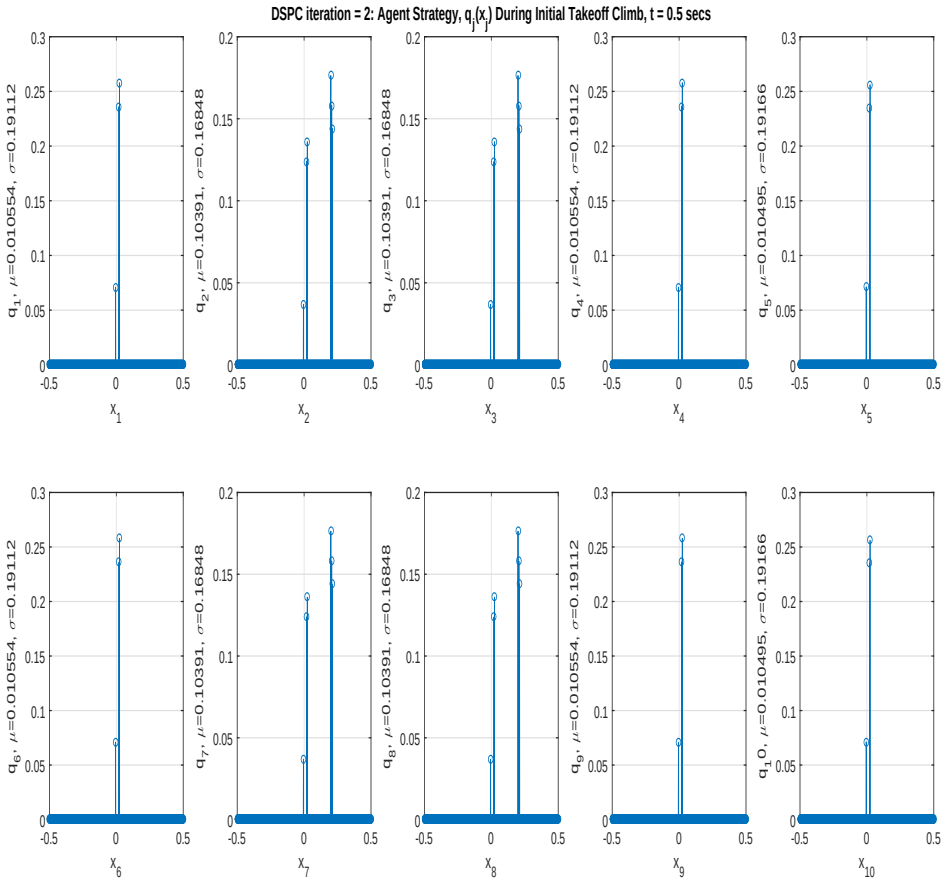


Figure 6.12: Nominal Case: DSPC Agent Strategy, $\mathbf{q}(\mathbf{x})$ at $k = 2$, During Initial Hovercraft Takeoff Climb

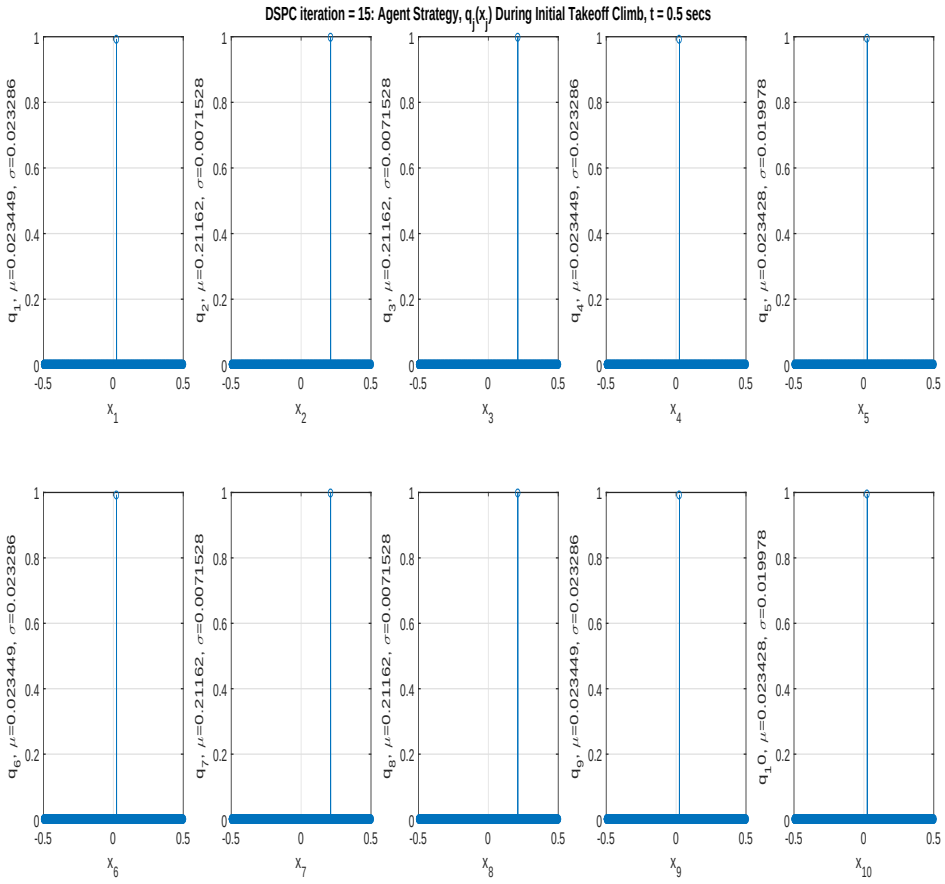


Figure 6.13: Nominal Case: DSPC Agent Strategy, $\mathbf{q}(\mathbf{x})$ at $k = 15$, During Initial Hovercraft Takeoff Climb

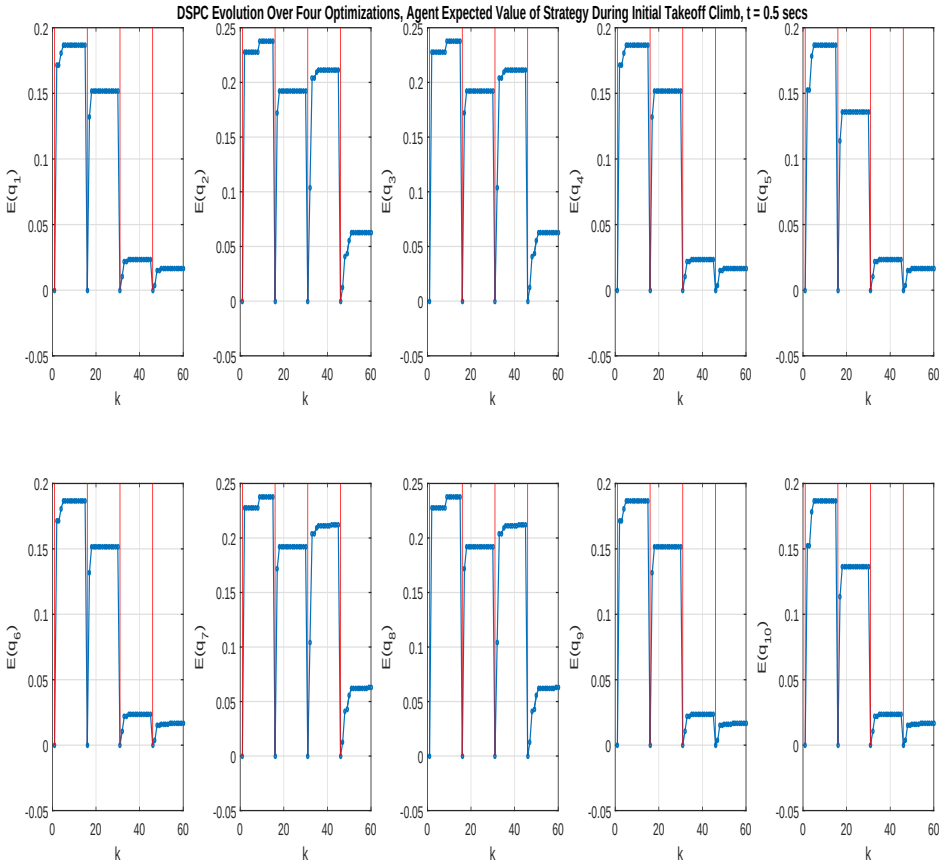


Figure 6.14: Nominal Case: DSPC Strategy Expected Value Evolution $E[\mathbf{q}(\mathbf{x})]$, During Initial Hovercraft Takeoff Climb

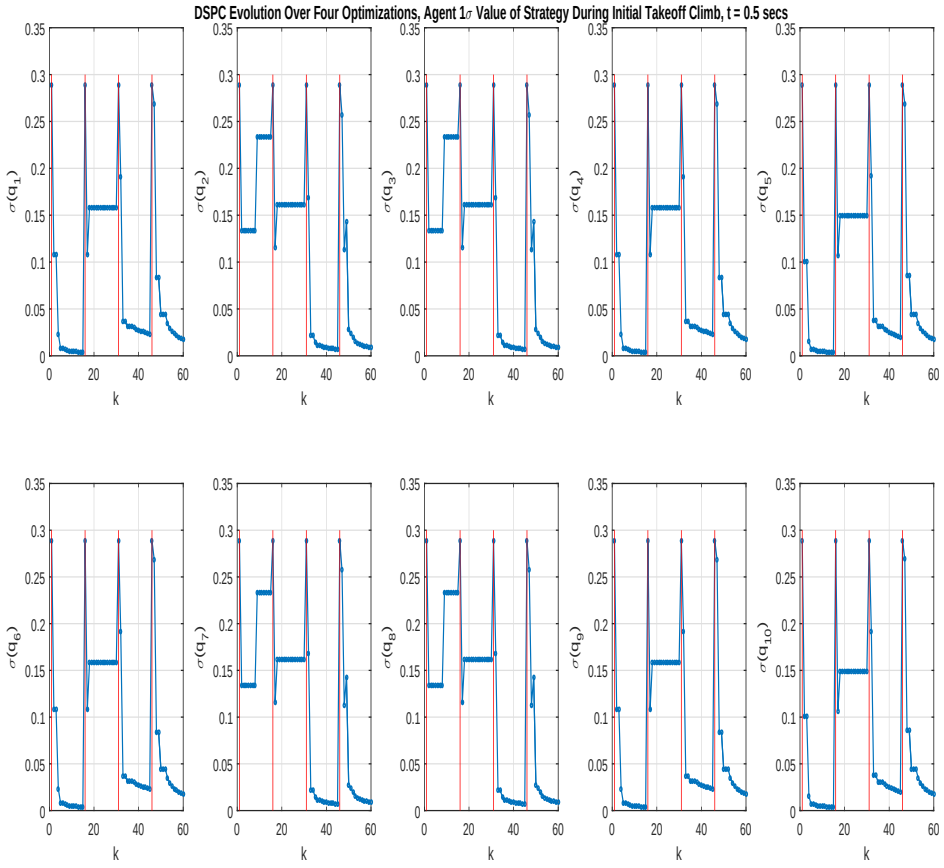


Figure 6.15: Nominal Case: DSPC Strategy Standard Deviation Evolution $\sigma[\mathbf{q}(\mathbf{x})]$, During Initial Hovercraft Takeoff Climb

6.3 Linear Control Effectivity, Failure Case 1

In this section, the equivalent deca-rotor model with linear control effectivity matrix previously evaluated is simulated, with the exception of failed engine controller #5 at $t = 10$ seconds simulation time when the hovercraft is translating through the first leg of the mission to the second waypoint. Here, as simulated in Ch. 5, global failure observability is not assumed. The on board model of the control effectivity matrix, $\mathbf{B}$, is not updated for either QCAT WLS or DSPC. The failure is modeled as a local digital engine controller latching to a zero command, yielding an abrupt decrease in thrust to 0%[4].

Fig. 6.16 compares the achieved trajectories for QCAT WLS and DSPC. Immediately after the failed engine, the QCAT WLS system loses stability and the vehicle is lost with altitude dropping to 0 m. The DSPC distributed allocation system however maintains stability and completes the mission, although tracking performance is degraded with increased deviation in the trajectory paths. Figs. 6.17-6.18 present the time history tracking of the position and velocity states for both methods throughout the failure scenario. Subsequent to the engine out, the DSPC rotational position and velocity tracking exhibits increased activity as the system compensates for the failure, also apparent in Fig. 6.19 in the virtual control commands.

Additional gain limiting and optimization in future work may quiet the post-failure response. For example, rather than solely effector position and rate limiting, the effective gain of the total solution can be computed and limited per the method

[4]The failure is modeled as agent #5 freezing computation with a stagnant local strategy. The high rate distributed network bus continues to sample and provide an expected value of $u_5^- = 0$ to the remaining collective subsequent to $t = 10$ seconds simulation time per the Failure Scenario B as detailed in Ch. 5.

discussed by Bordignon and Bessolo in [17] to preserve structural coupling gain margin. This item is further discussed in Ch. 7 as a consideration for future work.

The key takeaway from this failure case result is the increased resiliency offered by the distributed control allocation approach, apparent in Fig. 6.20 where DSPC simply perceives the agent #5 command level at 0% as the expected value of the local strategy for #5 and compensates with the remaining effectors accordingly. Finally, Fig. 6.21 depicts the L_2 objective value for the initial takeoff segment and through the failure transient. Initially through the takeoff mission segment, both control allocation methods compare in magnitude of L_2 cost. The bottom row of plots shows the difference in cost with a sign convention of the DSPC L_2 minus the QCAT WLS L_2 cost. The right column of plots zooms on the time axis of interest about the failure at $t = 10$ seconds simulation time. DSPC, in this case, maintains the average L_2 cost throughout the transitory period while the QCAT WLS L_2 cost triples in magnitude as the hovercraft destabilizes.

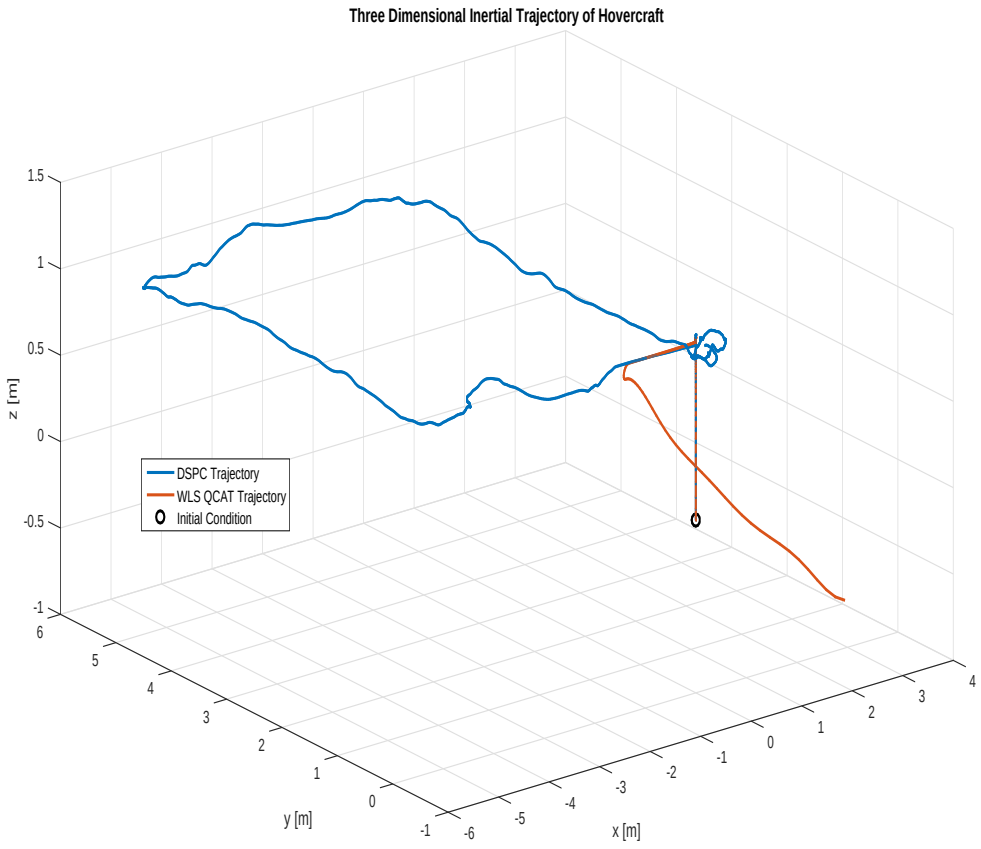


Figure 6.16: Failure Case 1: Three Dimensional Inertial Trajectory for Deca-Rotor Hovercraft Comparing QCAT WLS Centralized Control Allocation and DSPC Distributed Control Allocation

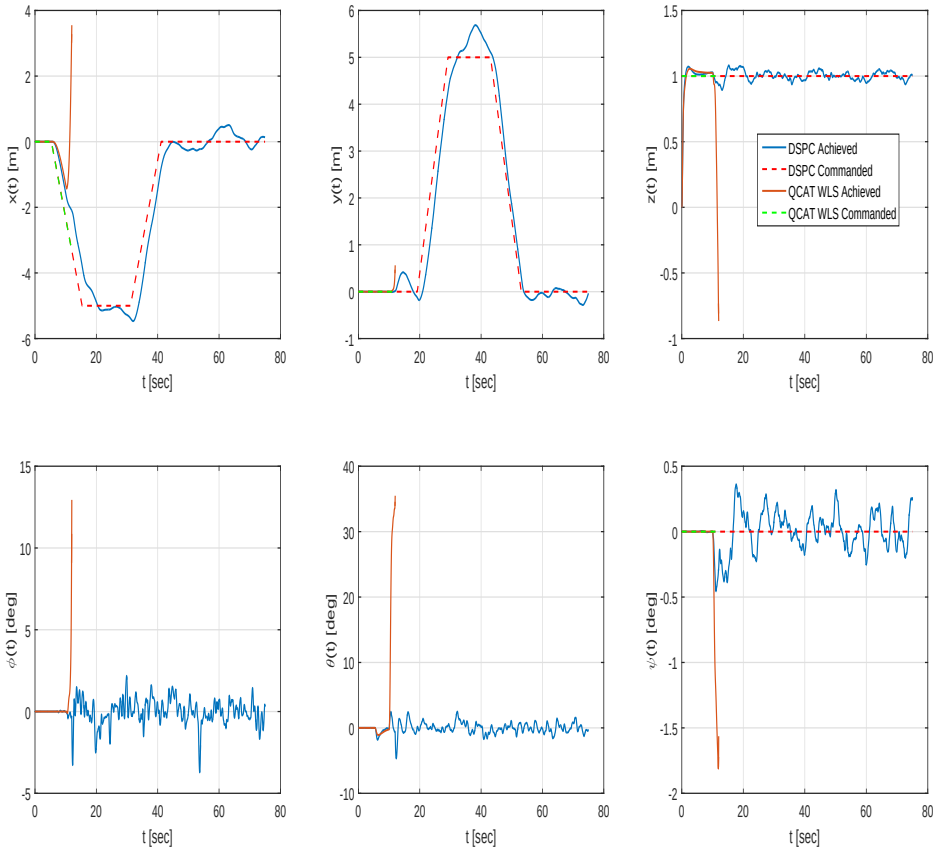


Figure 6.17: Failure Case 1: Inertial Position Time History for Deca-Rotor Hovercraft Comparing QCAT WLS Centralized Control Allocation and DSPC Distributed Control Allocation

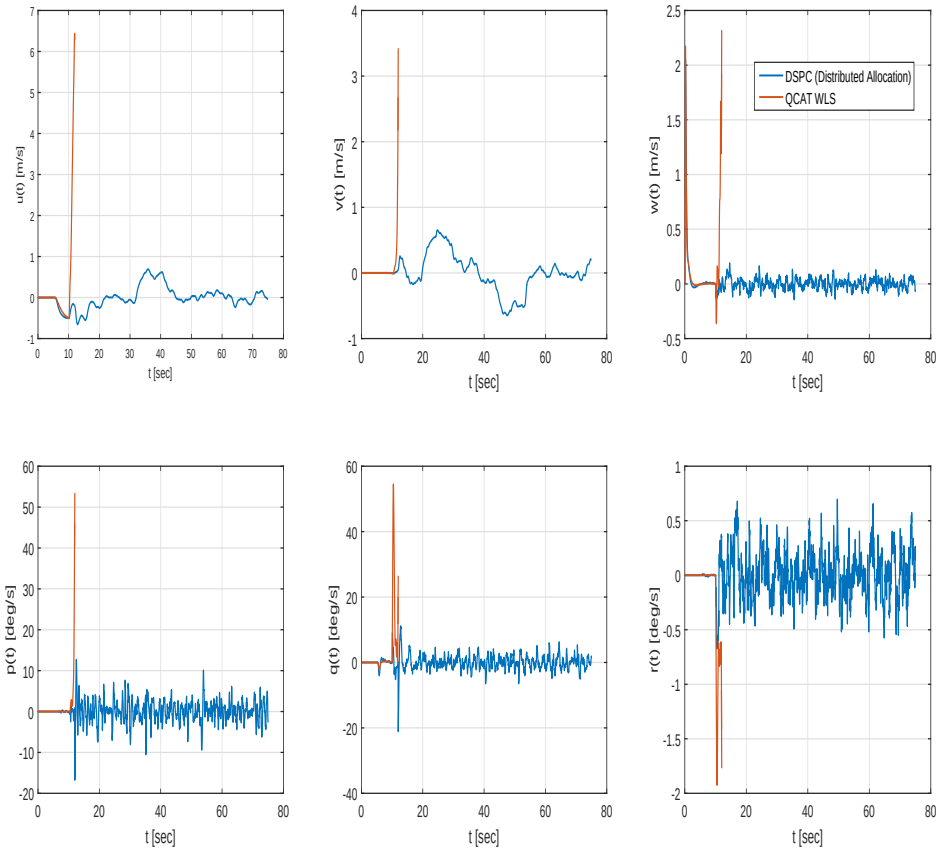


Figure 6.18: Failure Case 1: Body Frame Velocity for Deca-Rotor Hovercraft Comparing QCAT WLS Centralized Control Allocation and DSPC Distributed Control Allocation

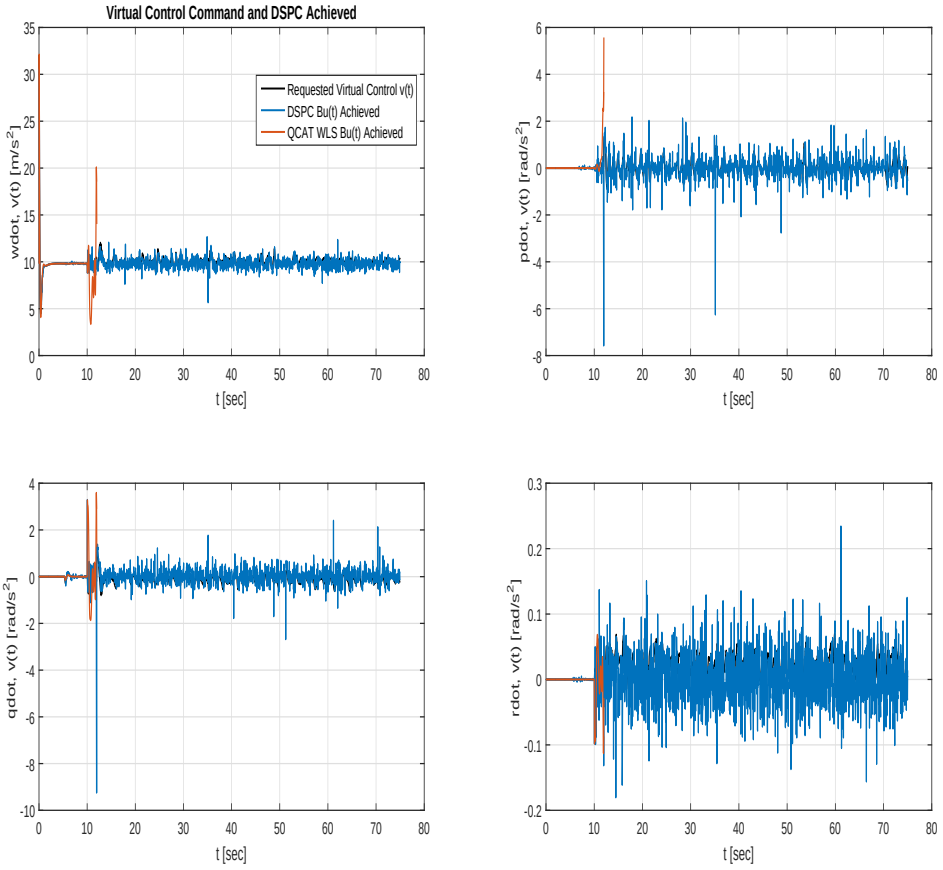


Figure 6.19: Failure Case 1: Virtual Control Performance Using Distributed Control for Deca-Rotor Hovercraft Comparing QCAT WLS Centralized Control Allocation and DSPC Distributed Control Allocation

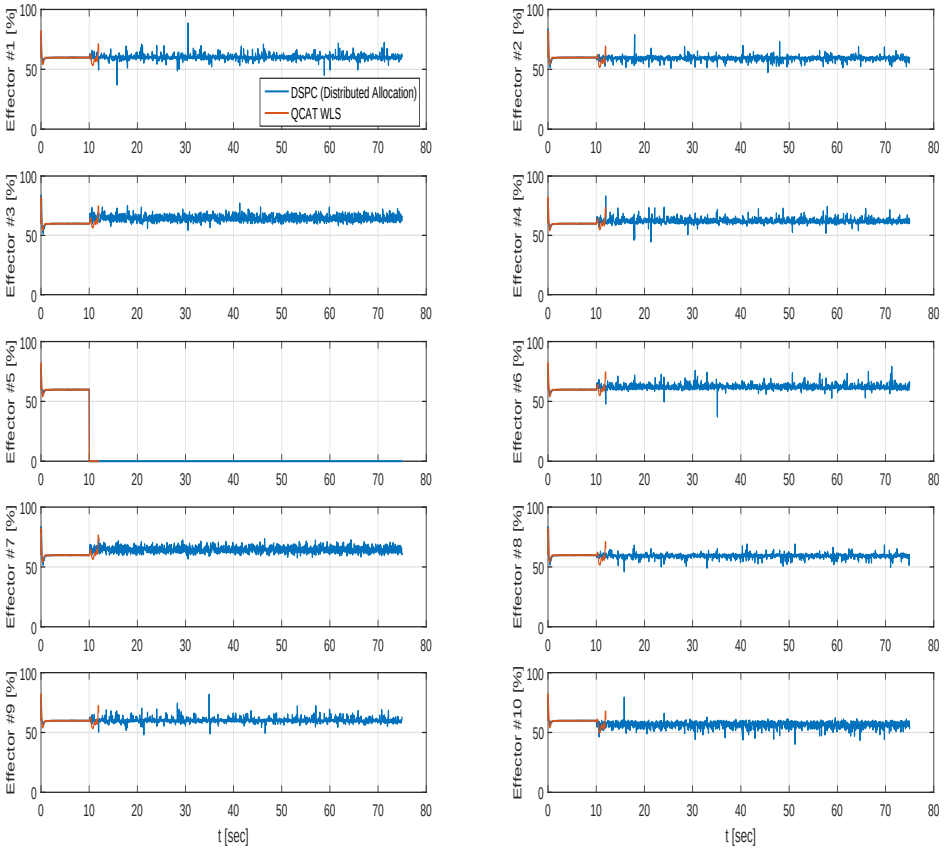


Figure 6.20: Failure Case 1: Effector Thrust Levels Using Distributed Control for Deca-Rotor Hovercraft Comparing QCAT WLS Centralized Control Allocation and DSPC Distributed Control Allocation

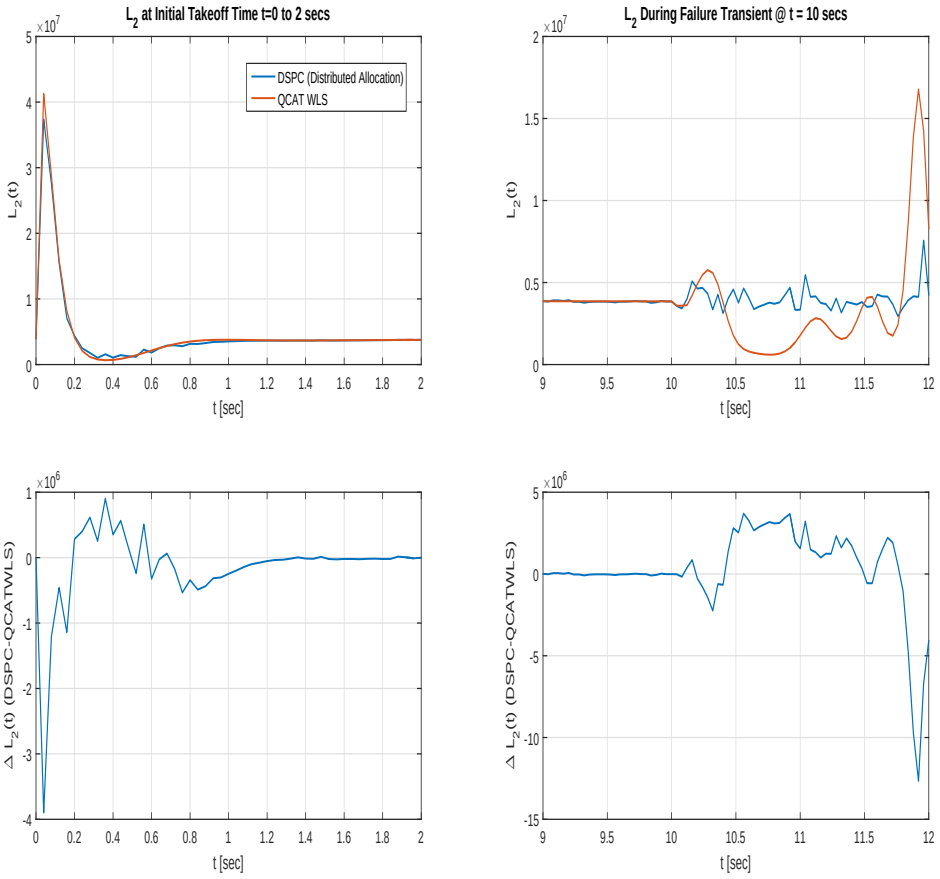


Figure 6.21: Failure Case 1: L_2 Objective Values Using Distributed Control for Deca-Rotor Hovercraft Comparing QCAT WLS Centralized Control Allocation and DSPC Distributed Control Allocation

6.4 Linear Control Effectivity, Failure Case 2

This section simulates a second failure case where multiple engines are lost throughout the mission to further demonstrate the robustness gained with distributed control allocation. Engine #5, engine #8, and engine #1 fail at $t = 10$, $t = 15$, and $t = 20$ seconds simulation time, respectively. While the distributed control allocation method maintains stability, tracking performance degrades noticeably as shown in Fig. 6.22 where the trajectory deviates after waypoint 2. Additionally, with the increased trajectory deviation, the navigational control law governing the waypoint auto-sequencing increases the overall time to complete the mission at approximately 75 seconds simulation time, as opposed to less than 60 seconds for the single failure case. This later sequencing is apparent in Fig. 6.23 where the dashed command reference lines transition at delayed times due to the increased deviation in position. After the second failure on engine #8, the y position deviates off course by approximately 1.5 m. The third failure on engine #1 then occurs as the system is recovering the y error and this response couples into a deviation on the x position. Altitude is maintained throughout the failure scenario, however, and after the course overshooting, the distributed allocation control routine successfully returns the vehicle to the initial position. Figs. 6.24-6.25 present the time history of the deca-rotor velocity states and virtual control tracking for both methods throughout the failure scenario. The engine failures are presented in Fig. 6.26 with the remaining effector time histories. Finally, Fig. 6.27 presents the L_2 control allocation objective value for the initial takeoff segment and through all failure transients. Interestingly due to the increased activity post single-fail, the additional failures at $t = 15$ and $t = 20$ seconds are nearly indistinguishable in terms of total magnitude of L_2 cost.

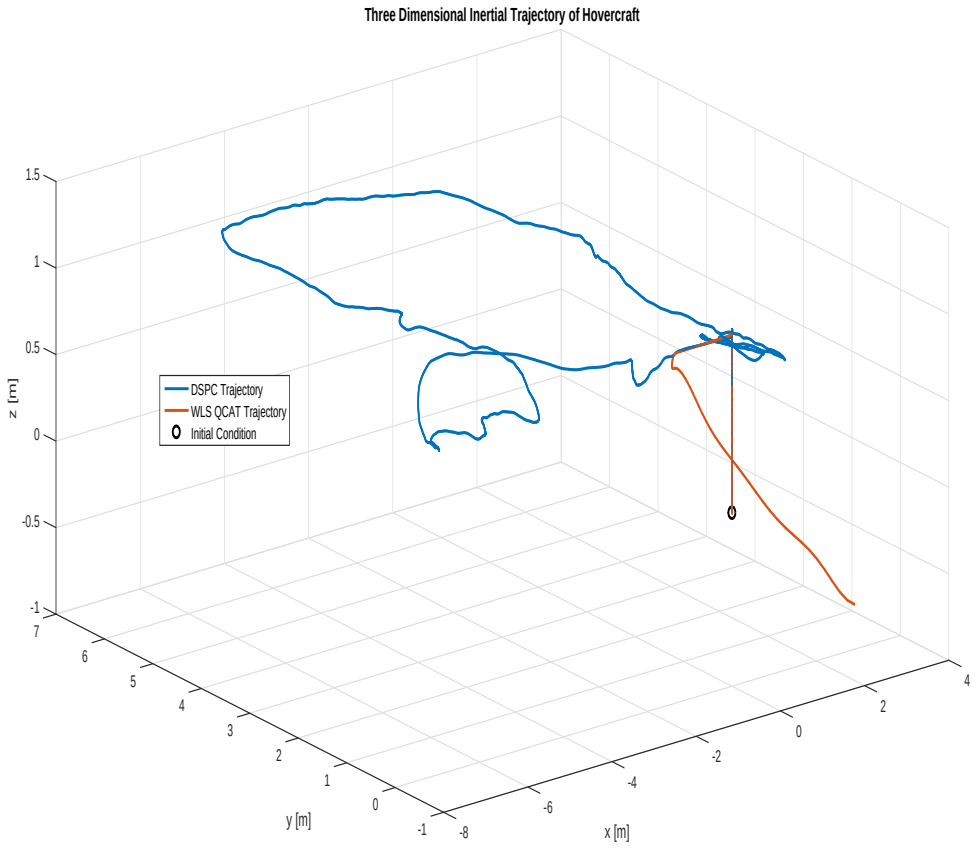


Figure 6.22: Failure Case 2: Three Dimensional Inertial Trajectory for Deca-Rotor Hovercraft Comparing QCAT WLS Centralized Control Allocation and DSPC Distributed Control Allocation

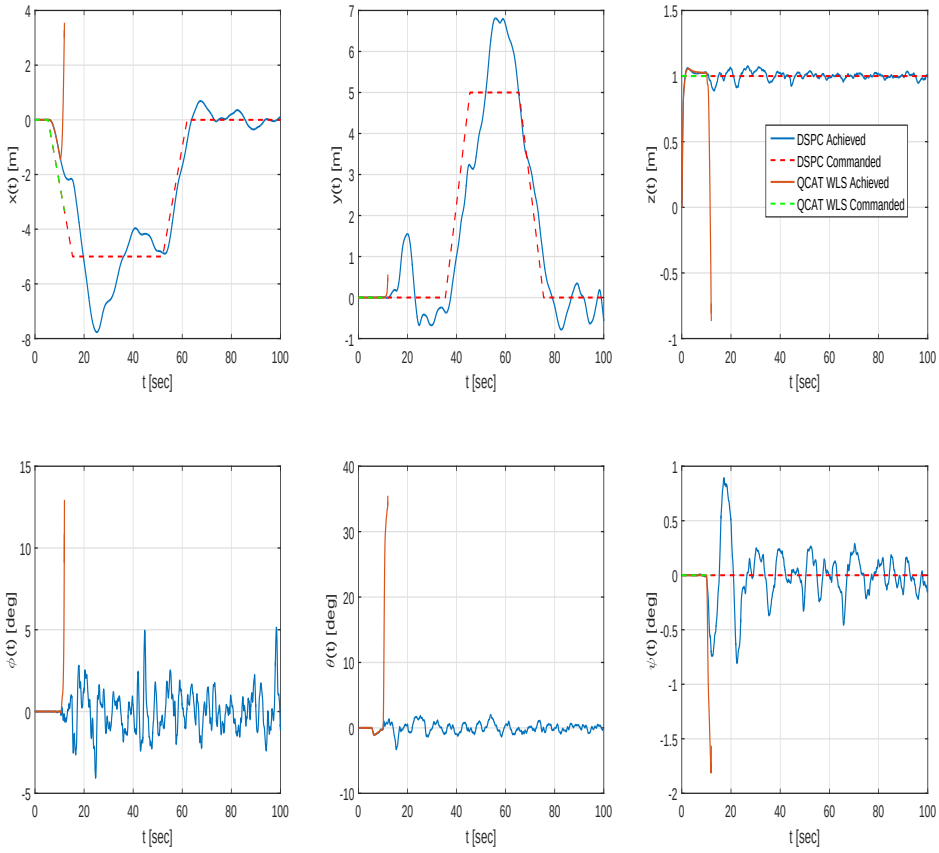


Figure 6.23: Failure Case 2: Inertial Position Time History for Deca-Rotor Hovercraft Comparing QCAT WLS Centralized Control Allocation and DSPC Distributed Control Allocation

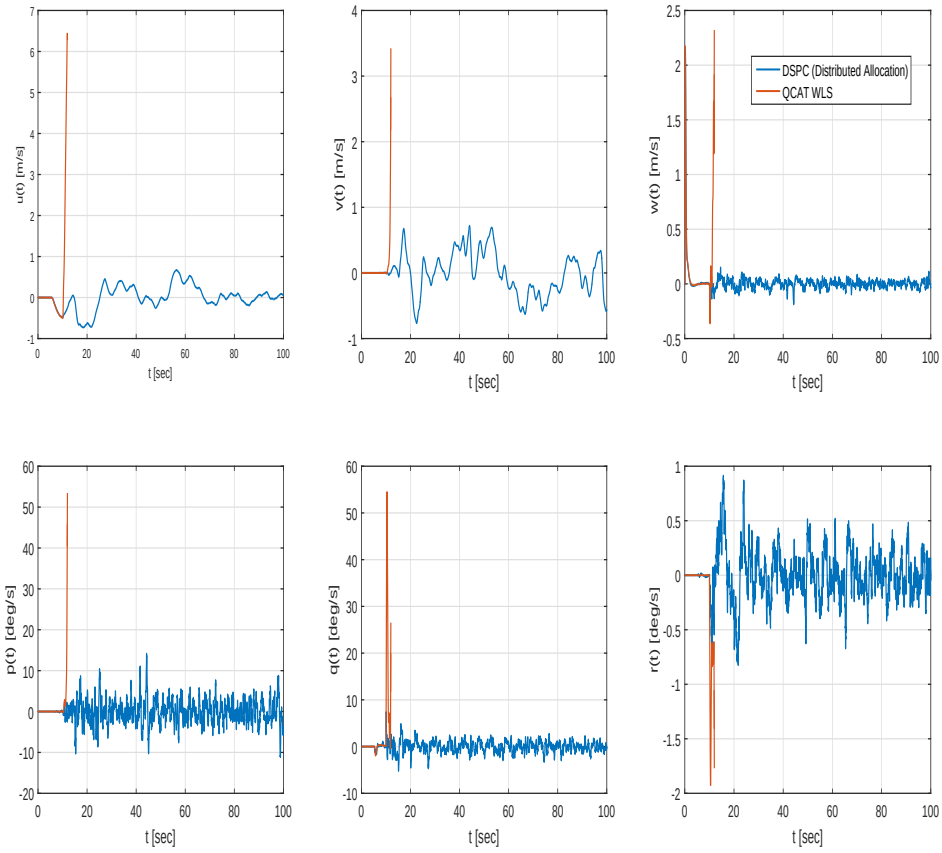


Figure 6.24: Failure Case 2: Body Frame Velocity for Deca-Rotor Hovercraft Comparing QCAT WLS Centralized Control Allocation and DSPC Distributed Control Allocation

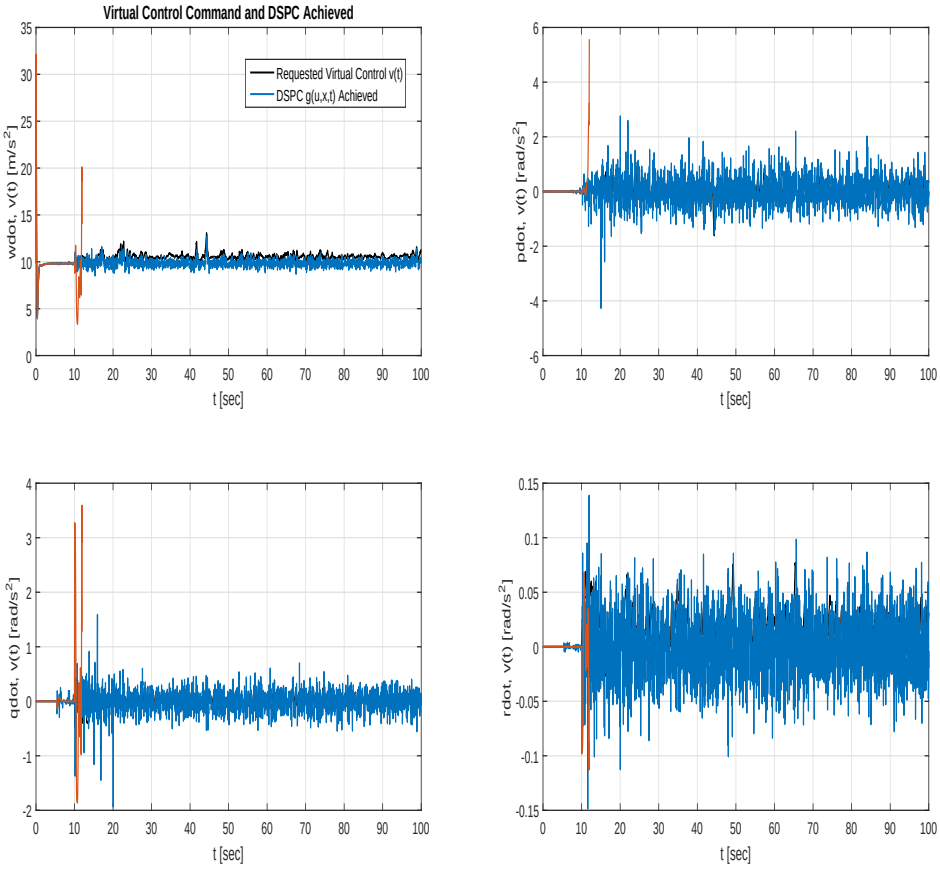


Figure 6.25: Failure Case 2: Virtual Control Performance Using Distributed Control for Deca-Rotor Hovercraft Comparing QCAT WLS Centralized Control Allocation and DSPC Distributed Control Allocation

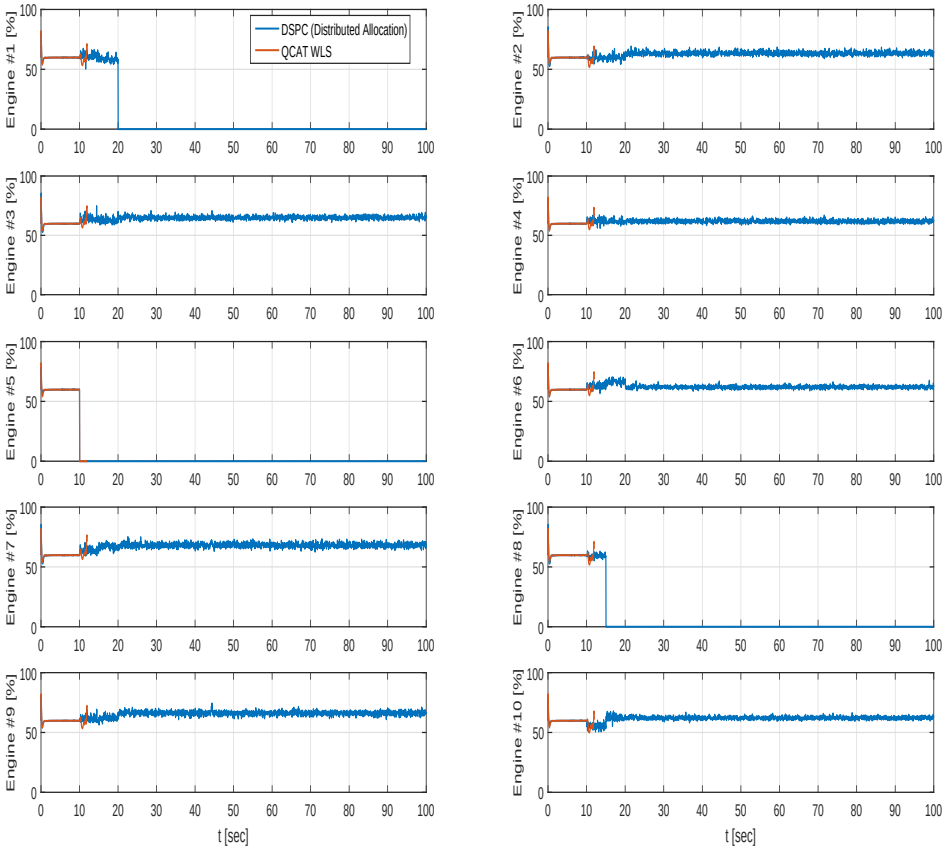


Figure 6.26: Failure Case 2: Effector Thrust Levels Using Distributed Control for Deca-Rotor Hovercraft Comparing QCAT WLS Centralized Control Allocation and DSPC Distributed Control Allocation

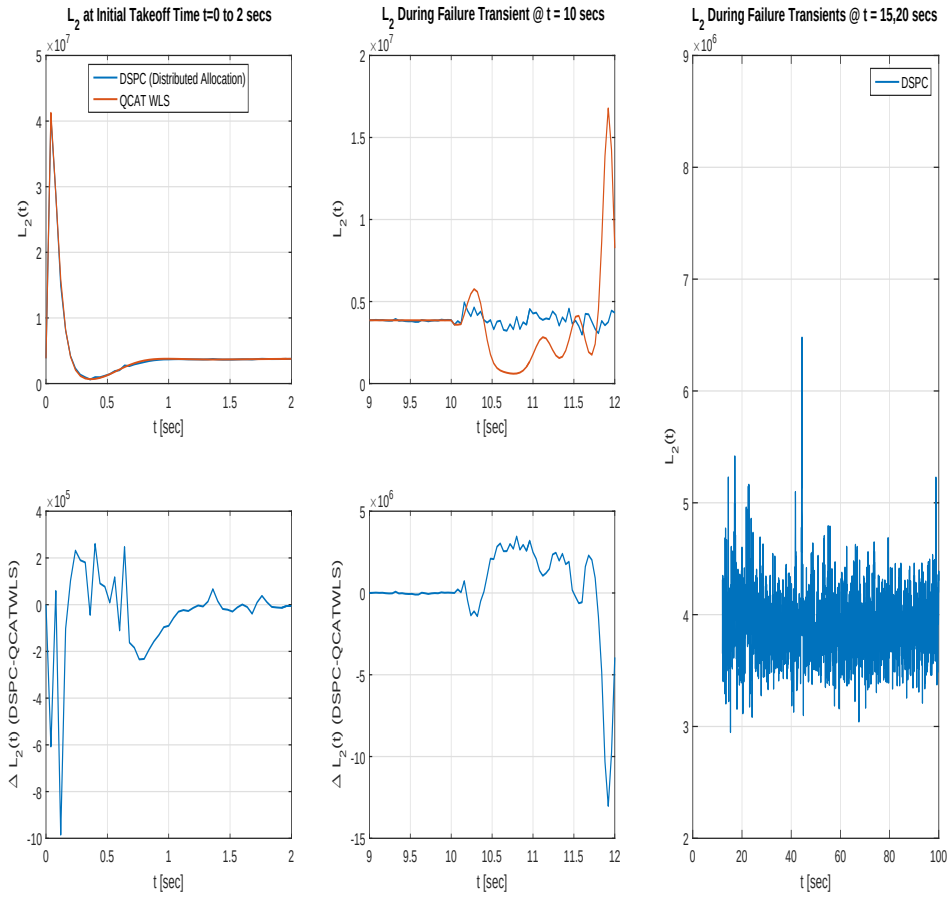


Figure 6.27: Failure Case 2: L_2 Objective Values Using Distributed Control for Deca-Rotor Hovercraft Comparing QCAT WLS Centralized Control Allocation and DSPC Distributed Control Allocation

6.5 Nonlinear Control Effectivity

This section incorporates a nonlinear control effectivity mapping that is a function of the vehicle states. Specifically, as the local velocity at each propeller on the deca-rotor increases or decreases, thrust efficiency fluctuates. The total local velocity, $\mathbf{u}_{loc_{j_B}} \in \Re^3$, is determined by both the translational velocity of the center of mass of the vehicle, $\mathbf{V}_B = [u\ v\ w]^T \in \Re^3$, and the angular velocity of the rigid body, ω_B as

$$\mathbf{u}_{loc_{j_B}} = -\mathbf{S}(\omega_B)\mathbf{d}_j + \mathbf{V}_B \tag{6.6}$$

where $\mathbf{d}_j$ represents the lever arm distance from center of mass to the center of thrust of the j^{th} motor. Referencing [94], nonlinear thrust efficiency for the j^{th} motor, η_j can be modeled as a quadratic decreasing function with increasing local velocity. However, in this study, the interest in thrust efficiency pertains to both increasing and decreasing local velocity. For example, an angular velocity maneuver during a zero translational rate hovering position will expose the retracting rotors on the body to negative local velocities. Likewise, advancing rotors are exposed to an increase in local velocity. This phenomenon is modeled with a notional relation for mapping the normal component of the local velocity vector $u_{loc_{j_{B_z}}}$ to nonlinear thrust efficiency η_j as [94, 122, 123, 124, 125, 126]

$$\eta_j = \max[-.12(u_{loc_{j_{B_z}}} - 0.3)^2 + 1, 0.8] \tag{6.7}$$

which is used as a multiplicative factor on actual thrust achieved for the hovercraft. Fig. 6.28 depicts η_j as a function of $u_{loc_{j_{B_z}}}$.

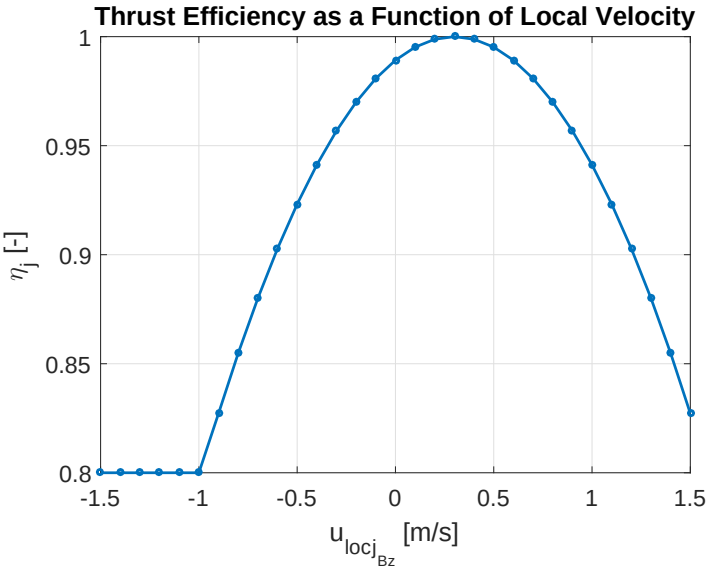


Figure 6.28: Nonlinear Case: Thrust Efficiency, η_j, a Function of the Normal Component of Local Velocity at the Propeller

Each agent depicted previously in Fig. 6.1 now evaluates the nonlinear control effectivity mapping, g, with the inclusion of η_j efficiencies in Eq. (6.7),

$$g(u,x,t) = \begin{pmatrix} \frac{1}{m}\sum_{j=1}^{N} F j_{z_B}\eta_j \\ \mathbf{I_B}^{-1}\left[\sum_{j=1}^{N} -\mathbf{S}(\mathbf{d}_j)\mathbf{F}j_B\eta_j + \tau j_B\right] \end{pmatrix} \tag{6.8}$$

For this testing, the waypoints were modified to include coupled navigational segments where the hovercraft translates and turns during altitude maneuvers. The updated waypoints are provided below in Table (6.5). After initial takeoff and climb to 1 m, a forward translation is commanded concurrent with a second climb to 5 m. A lateral translation is then commanded to -5 m during a descent to 1 m with a simultaneous heading change to 90 deg.

Table 6.5: Deca-rotor Flight Profile #2.

Waypoint ID	x [m]	y [m]	z [m]	ψ [deg]	t [sec]
1	0.0	0.0	1.0	0.0	5.0
2	5.0	0.0	5.0	0.0	1.0
3	5.0	-5.0	1.0	90.0	1.0

The roll acceleration penalty, $Wv_{\dot{p}}$, was set to 170 as shown below in Table (6.6). Two cases for DSPC are simulated and described below in Table (6.7) to enable comparison for the distributed allocation performance with an increased number of samples drawn per optimization step. Case one uses the default $m = 20$ and case two decreases to $m = 10$ samples drawn per optimization step.

Table 6.6: L_2 Control Allocation Optimization Weights for Nonlinear Effectivity.

Parameter	**Variable**	**Value**
Vertical Acceleration Error Penalty	$Wv_{\dot{w}}$	200
Roll Acceleration Error Penalty	$Wv_{\dot{p}}$	170
Pitch Acceleration Error Penalty	$Wv_{\dot{q}}$	200
Yaw Acceleration Error Penalty	$Wv_{\dot{r}}$	2000
Thruster Use Penalty	Wu_j	1.0
Thruster Desired Value	u_{d_j}	0.0
Effector Constraint Penalty	λ_{L_2}	0, 1e9
L_2 Acceleration Error Penalty	γ_{L_2}	1.0

Table 6.7: DSPC Parameter Settings for Nonlinear Effectivity.

Parameter	**Variable**	**Value**
Number of Agents	n_{agent}	10
Maxent Lagrangian Temperature	T	0.5
Probability Update Gain	α	0.001
Data Aging Gain	γ	.03
Samples Drawn Per Iteration	m	20, 10
Iterations Per Frame	n_{step}	15
Moves Per Agent	n_{moves_j}	$1e3$
Agent Effector Limit Constraint Gain	η	0.1
Agent Effector Upper Limit	$\overline{\mathbf{u}}$	$\mathbf{0.5}$
Agent Effector Lower Limit	$\underline{\mathbf{u}}$	$-\overline{\mathbf{u}}$

Fig. 6.29 presents the three dimensional inertial trajectory for DSPC distributed control allocation with 20 versus 10 samples per optimization step with the nonlinear control effectivity mapping. Both configurations accomplish the mission to takeoff and capture the three waypoints in Table 6.6. The inertial position time history is shown in Fig. 6.30. Translational state tracking nearly matches the DSPC-20 (20 samples per iteration) and DSPC-10 configurations, however the rotational position data shows increased activity in ϕ and θ with the reduced samples. This difference is also apparent in the body frame velocity time history comparisons in Fig. 6.31 with the p and q subplots. The corresponding inner loop virtual control commands and effector data are presented for the two configurations in Fig. 6.32 and Fig. 6.33, respectively. The local normal velocity at each effector and the effector efficiency is given in Fig. 6.34 and Fig. 6.35. Here, the new mission profile with the climb and dive segments is apparent as the efficiency gains for all propellers consistently drop below 1.0. Finally, the L_2 costs are depicted in Fig. 6.36 for the two DSPC configurations with 20 versus 10 samples per optimization step.

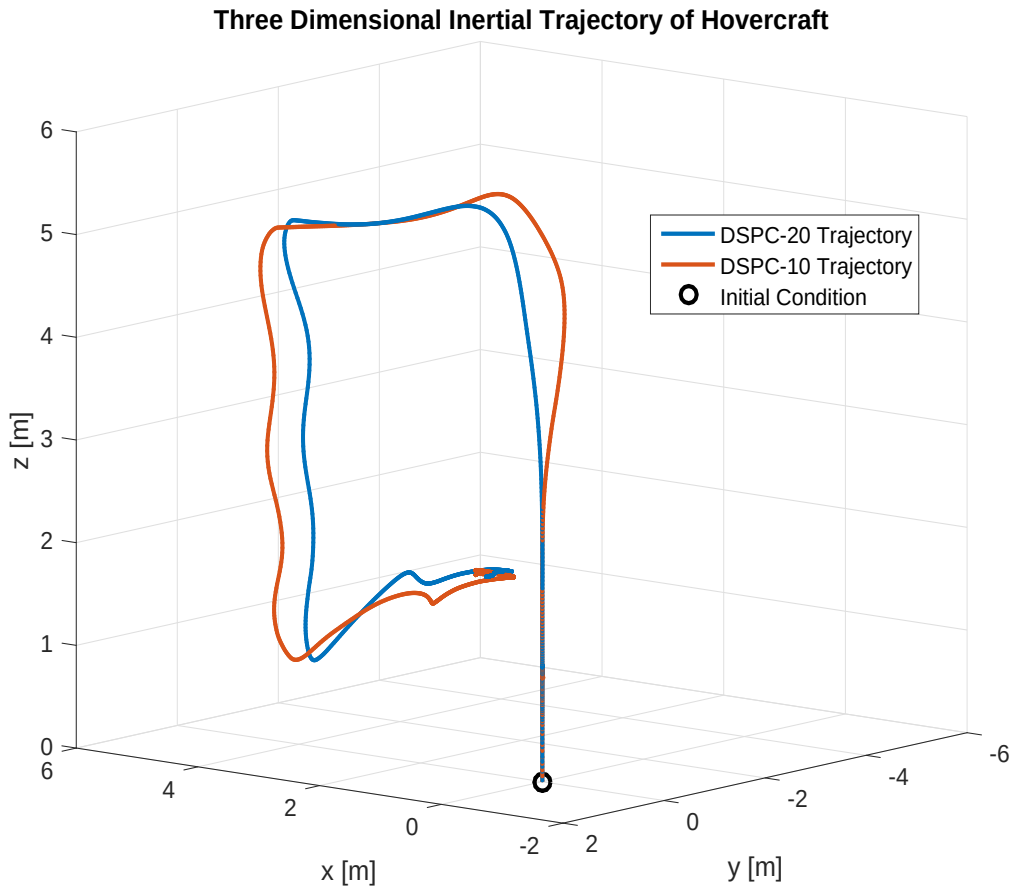


Figure 6.29: Nonlinear Case: Three Dimensional Inertial Trajectory for DSPC Distributed Control Allocation with 20 versus 10 Samples per Frame

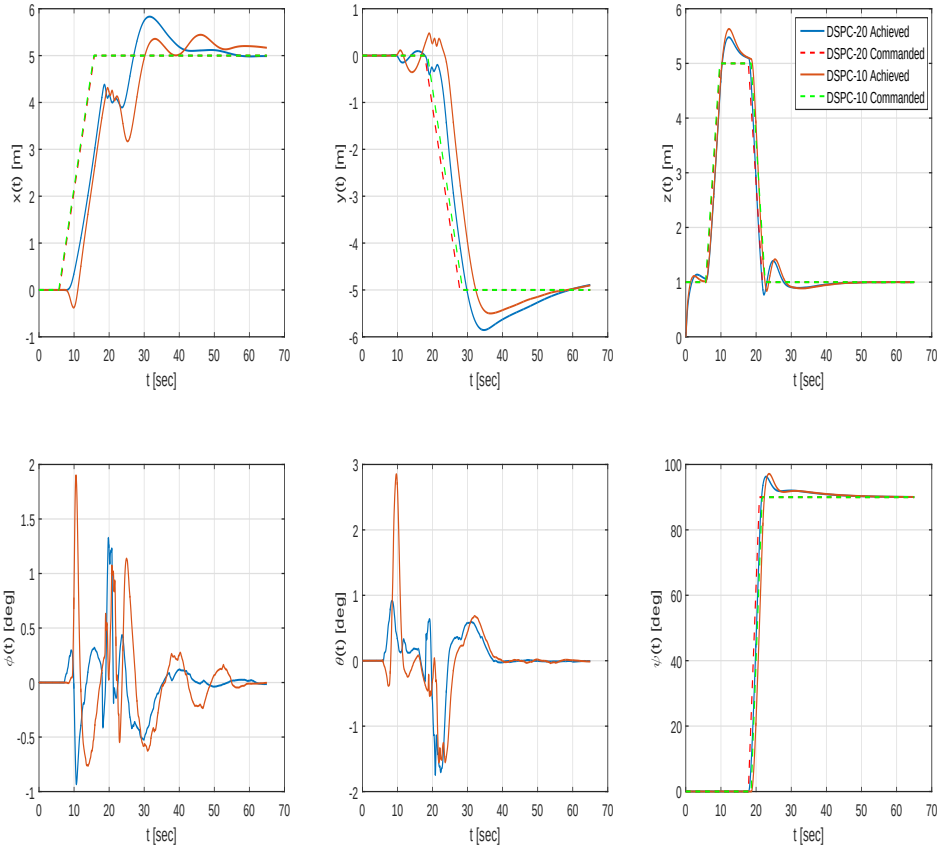


Figure 6.30: Nonlinear Case: Inertial Position Time History for DSPC Distributed Control Allocation with 20 versus 10 Samples per Frame

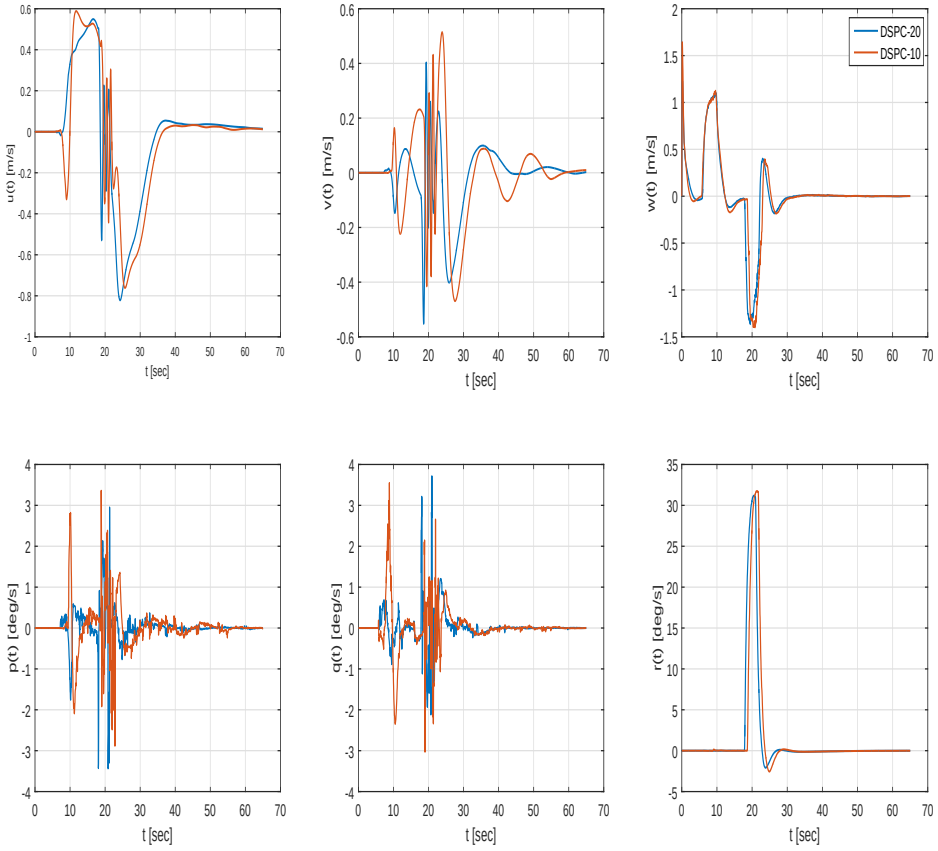


Figure 6.31: Nonlinear Case: Body Frame Velocity for DSPC Distributed Control Allocation with 20 versus 10 Samples per Frame

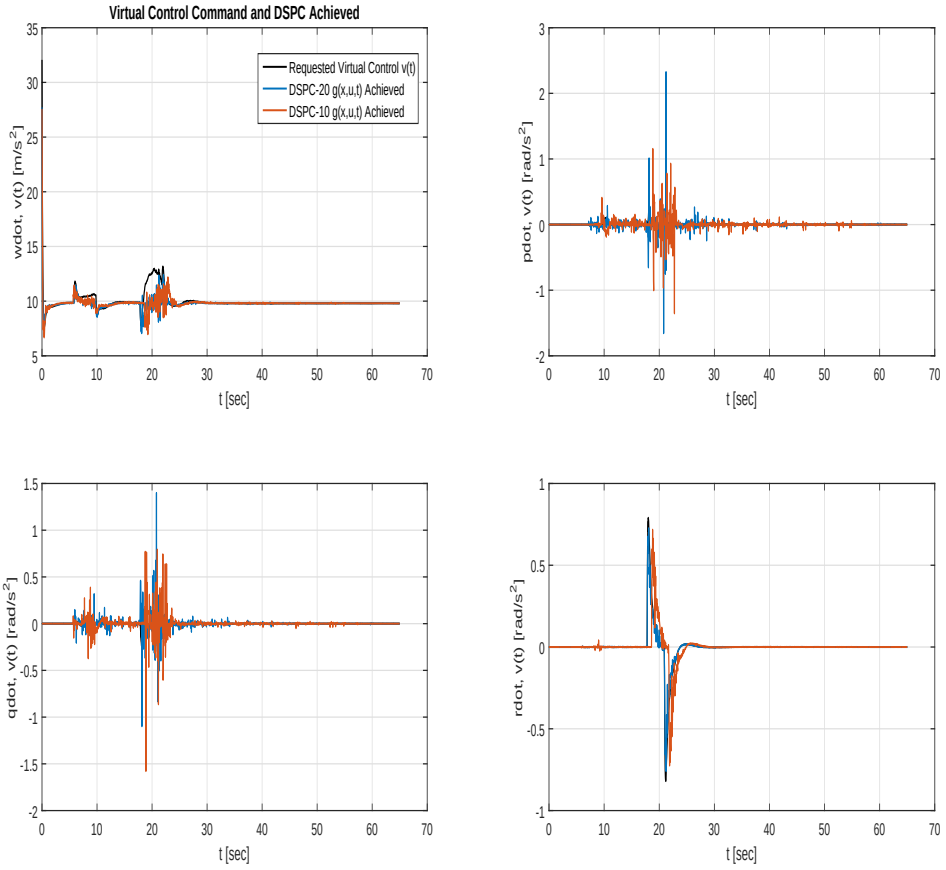


Figure 6.32: Nonlinear Case: Virtual Control Performance Using Distributed Control for DSPC Distributed Control Allocation with 20 versus 10 Samples per Frame

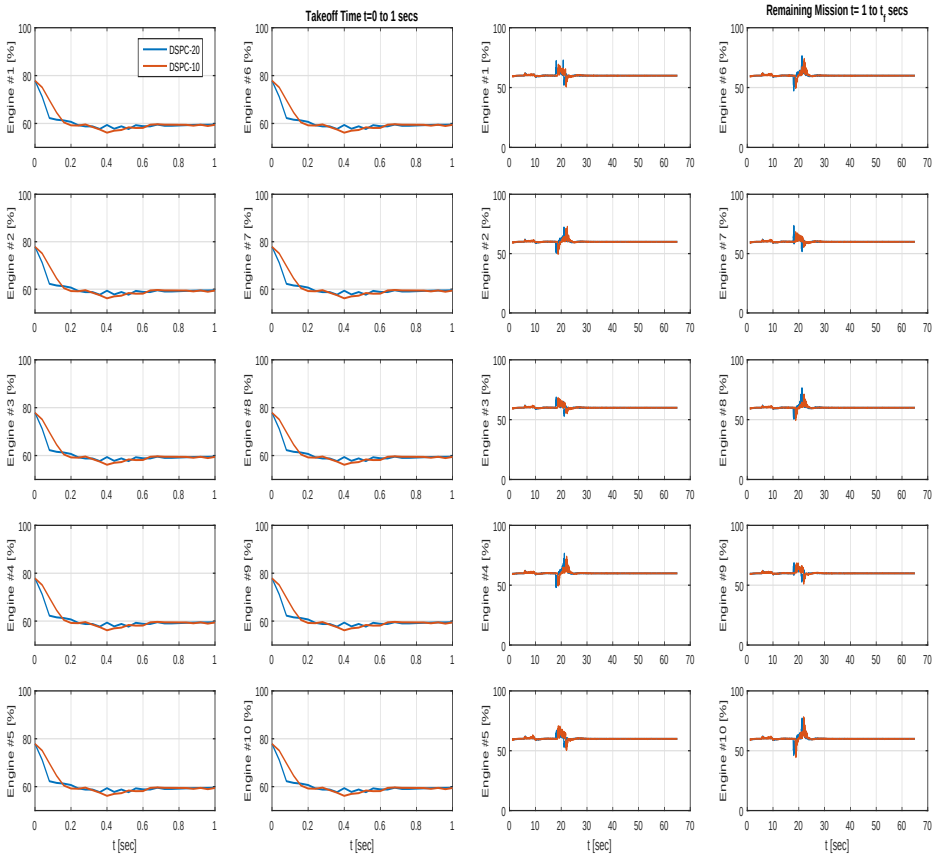


Figure 6.33: Nonlinear Case: Effector Thrust Levels for DSPC Distributed Control Allocation with 20 versus 10 Samples per Frame

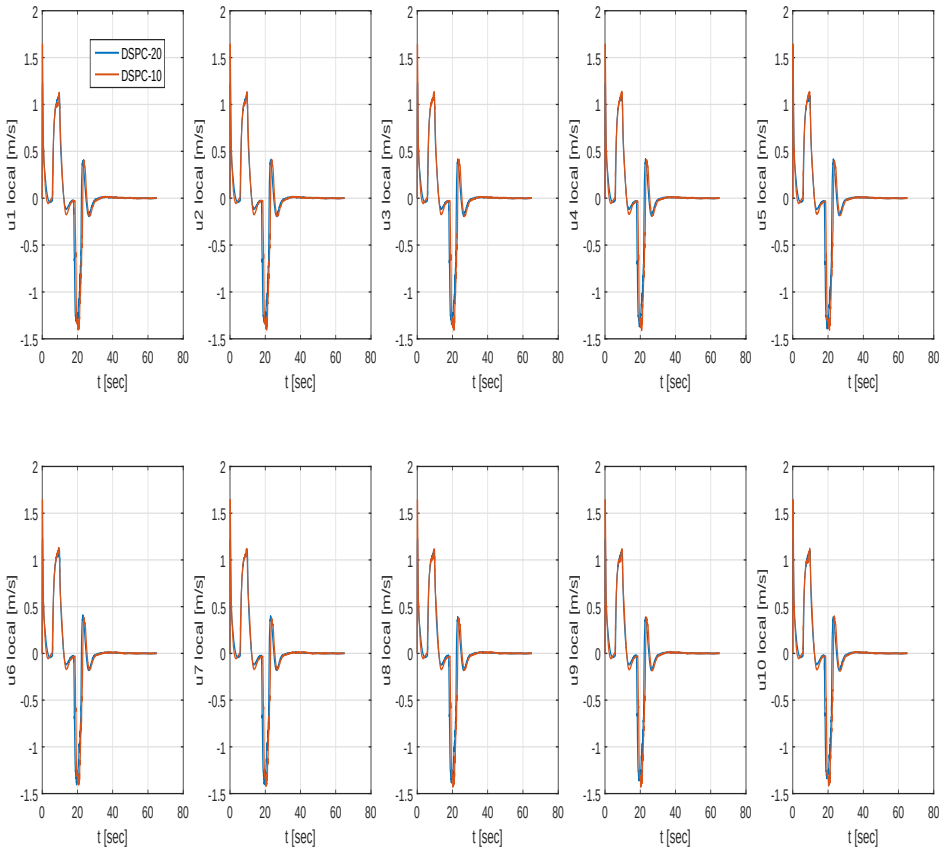


Figure 6.34: Nonlinear Case: Effector Local Normal Velocity for DSPC Distributed Control Allocation with 20 versus 10 Samples per Frame

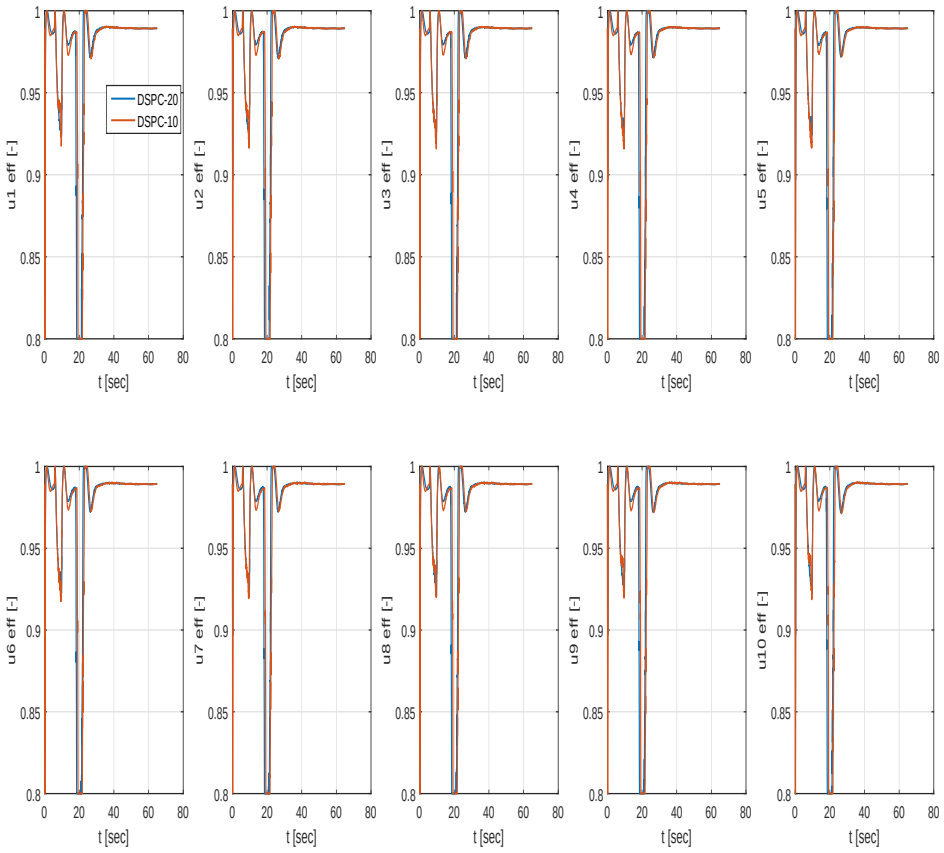


Figure 6.35: Nonlinear Case: Effector Thrust Efficiencies for DSPC Distributed Control Allocation with 20 versus 10 Samples per Frame

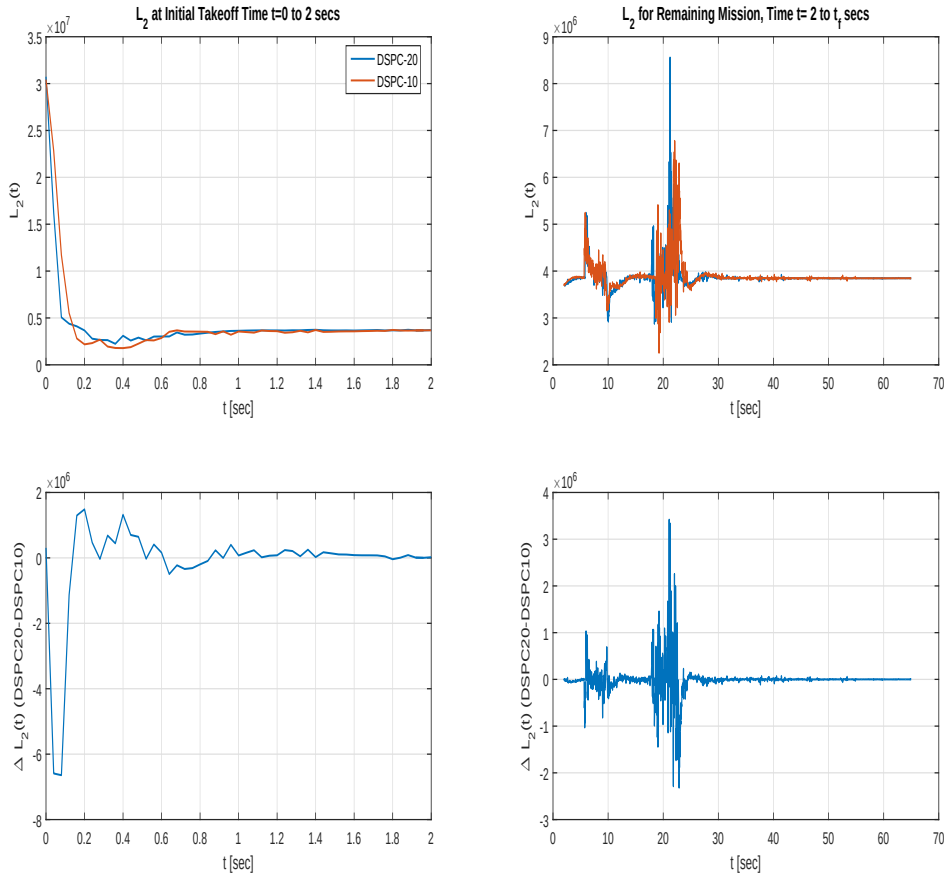


Figure 6.36: Nonlinear Case: L_2 Objective Values Using Distributed Control for DSPC Distributed Control Allocation with 20 versus 10 Samples per Frame

CHAPTER 7

CONCLUSION AND FUTURE WORK

This chapter concludes with a summary of findings and notable results. Finally, a future work road map is discussed in detail and enumerated at end of the section for quick reference.

7.1 Conclusion

In this research, a novel approach to the control allocation problem for a distributed effector suite has been developed by leveraging Wolpert Probability Collectives (PC). Both delayed and immediate sampling PC methods have been explored for use as decentralized techniques for local agent optimization for an underdetermined system, where centralized allocation may become intractable for a future generation vehicle with a large number of effectors. Modifications have been integrated into the delayed sampling PC (DSPC) approach with a technique to refine the sampling domain; a mechanism to iteratively latch the best local agent strategy for the L_2 control allocation optimization application; and a provision to use the expected value of the agent strategy over the distributed network rather than a single sample. Results from a recent construction of the tool set have been generated for a range of problems from simple unconstrained optimization to multi-dimensional control allocation optimization with effector position and rate limits.

Both open loop and closed loop example applications have been evaluated with the distributed control allocation method, with a range of off-nominal conditions explored from effector saturation to effector failures, with and without failure reporting.

The distributed control allocation method designed in this work demonstrated a natural resiliency to failure, where local agents continue to optimize individual effector position despite another engine failure, effector lockup, or the loss of a flight critical centralized vehicle management system. The local agent simply continues nominal optimization with the sampled strategy of other neighbors, regardless of their state as failed or healthy, and uses this last reported value of the neighbors in local cost evaluation.

Additionally, both linear and nonlinear control effectivity mappings have been considered in this work. In cases with highly nonlinear or non-affine control effectivity mappings, distributed control allocation with stochastic optimization via probability collectives offers a clear benefit. Often, linearization may not be practical due to the lack of availability of an analytic expression; processor expense for computing a finite difference Jacobian for a large number of effectors; or even in cases where severe nonlinearities preclude a valid linear approximation via finite differencing. In the latter, traditional control allocation which typically requires linear control effectivity mappings may fail to find a stabilizing solution, as demonstrated in this research.

Finally, a six degree of freedom simulation for a hypothetical hovering platform with a distributed effector suite has been constructed in this work for evaluation with a new toolbox for distributed control allocation. A 10 agent delayed sampling PC approach was applied to a deca-rotor, with one agent per motor serving as a high rate networked and distributed control allocation system. A traditional dynamic inversion (input-output feedback linearization) approach was employed for the inner loop closure, and classical proportional-integral-derivative control laws were used to establish desired dynamics and tracking for the deca-rotor, including a navigational outer loop for waypoint guidance and sequencing.

Multiple challenges for using delayed sampling probability collectives as a distributed control allocation routine have been identified as a result of this research. The most significant of these challenges is the (at times) difficult process of identifying the best set of parameters for the optimizer. Exhaustive Monte-Carlo sweeps were conducted with a matrix of varying parameters to search for the best performing set. In some cases after optimization, agent strategies exhibit high variance leading to excessive activity (on the contrary, the ability to observe the variance during control allocation is also a useful feature for determining health of the system). Additionally, the current implementation of DSPC on a single computer workstation requires additional computational processing to handle the significant number of function evaluations required for the optimization. Without embedding the algorithm to multiple processing cores, or multiple hardware targets, parallel processing benefits are expected, but have not been substantiated. Finally, the implementation of DSPC in this work employs a high rate network bus to allow all iterations to complete for the optimization in between discrete execution frames of a stability augmentation system (SAS), a potential challenge for practical use. If the SAS inner loop system executes at 100 hz, the DSPC high rate network bus must complete the maximum number of optimization iterations (or converge to a stopping tolerance) within .01 seconds.

7.2 Notable Results

In summary, the following notable results have been accomplished in this research

i. Distributed architecture for control allocation and inner loop stabilization: employing Wolpert probability collectives as a novel control allocation toolbox enables a decentralization of the modern day flight control computer, effectively

increasing resiliency by moving primary inner loop functionality to the local effector scope as an actuator or engine digital controller.

ii. DSPC Control Allocation enables online evaluation of nonlinear control effectivity: typical control allocation methodologies to date require linearization of the control effectivity matrix or mapping, requiring online finite differencing or the evaluation of a linearized analytic expression, introducing computational expense or noise in Jacobian formulation and the loss of potentially critical nonlinearities for the stabilizing effector solution. The distributed control allocation method designed in this work, however, circumvents these issues by making use of the nonlinear control effectivity mapping for agent cost evaluation throughout the L_2 optimization process.

iii. MATLAB Simulink® toolbox allows rapid integration into future studies: this work has constructed a Simulink library that leverages model referencing and embedded code to create a drag and drop control allocation block for the control law designer. The nonlinear control allocation routine is loaded into the referenced model for all agents to evaluate throughout the optimization process. The distributed architecture of the algorithm is apparent by inspection of the model diagram and is conducive to deployment to an implementation on hardware with local processors per agent and a physical network. At the time of this work, the physical network constitutes a complete graph where the expected value of each strategy is shared for the collective, although future work may reduce this requirement further as discussed ahead.

iv. Generic Hovercraft simulation with dynamic inversion scheme: a model environment has been constructed to allow rapid analysis and simulation for a hovering platform with a specified number of effectors. The current implementation applies a radial distribution of the engine locations over a 2π rotation with respect to the

center of mass, although this can be modified to any geometry of interest. After setting mass properties of the vehicle, the simulation tools linearize and extract the control effectivity mapping for controllability checks and for implementation into linear control allocation tools (e.g. the Quadratic Control Allocation Toolbox by Härkegård using weighted least squares with constraints) for comparison to the distributed approach. The inner and outer loop control system architecture employing dynamic inversion can be used interchangeably with various hovercraft concepts.

7.3 Future Work Road Map

As with many research endeavors, this investigation has addressed specific questions, and along the way, has identified a multitude of new questions, creating a rich environment for future study and contribution. The purpose of this section is to provide an outlook on these potential items in a future work road map for consideration with efforts ahead.

Future work involving higher dimensional examples with noisy and potentially discontinuous objectives is warranted. For example, a mechanical hysteresis with an actuator could be problematic for typical optimization routines requiring a smooth landscape. This work has focused on the L_2 norm for the control allocation optimization problem and future work should include other types of norms, L_1 and L_∞. Additionally, the objective function may have internal states in the high rate cost forecasting for each agent. The potential for state inclusion could allow including actuator dynamics and other predicted vehicle responses in order to improve agent compensation for the actual achieved response with respect to the commanded action.

Methods for algorithm parameter optimization for various applications should be investigated in order to quicken convergence, reduce variance, improve robustness,

and facilitate the use of Probability Collectives methods for other applications. In the two dimensional case (with two agents), the L_2 and Maxent Lagrangian cost can be viewed as a three dimensional surface or contour plot as a technique to facilitate choosing the appropriate algorithm values for temperature and probability distribution update rate. However, for higher dimensional examples, these selections become more complex, and additional insight into the approximate values or even bounds for convergence will be helpful in future applications. One potential investigation could be to bound algorithm parameter selections as a function of expected maximum change in the expected objective, barring computational limits for making this determination. For example, the probability update law in DSPC includes a proportional gain multiplied by the difference in the objective function. Considering these maximum computable differences in the objective function may allow for an analytical selection of the required gain for convergence within an acceptable duration. Another trade study could consider longer optimization cycles with reduced samples drawn per iteration. Ultimately, a superior performing approach minimizes the total number of objective evaluations throughout the optimization.

Advanced adaptive sampling techniques should be explored from more recent literature on Probability Collectives to further improve convergence times. Rather than a fixed domain search, prior work investigated an adaptation throughout the agent optimization process by centering the domain about the expected value of the strategy and shrinking the overall interval at a fixed rate. Rather than solely considering a deterministic shrink rate for the domain, this work explored modifying the sampling domain as a function of the variance of the current probability distribution. An alternative potential direction could be to initialize the search domain with a coarse set of bins, where the domain bin represents a discretized axis of agent actions for the effector commands. Performance may be increased by refining this

discretization accordingly throughout optimization. For example, probability bins that do not increase in height throughout iteration may remain wide, alleviating additional computation with a reduction in required objective evaluations. However, bins that exhibit higher probability may be split to allow additional resolution into the specific action. Additional work on this front may further improve convergence time and accuracy of the final solution.

Observability studies with additionally constrained network topologies should be considered where inter-agent communication exists on non-complete graphs. In this work, the expected value of the strategy for all agents is shared over a high rate network bus, constituting a complete graph as the network topological adjacency matrix. Every agent observes the strategy of the entire collective. Relaxing this requirement will improve the practicality of this distributed control allocation method and is intuitive for task dependent control allocation requirements. For instance, often in parallel processing systems, a clustering approach is leveraged where subgroups are allocated rather than every computational node. In this case, the network topology is governed by the boundary conditions between clusters. For example, a span-wise distribution of engines for yaw control could warrant subgroup clustering where perhaps a more efficient distributed allocation routine commands four groups of engines rather than each individual agent. In this hypothetical case, perhaps only the outboard starboard engine cluster communicates directly with the inboard starboard engine cluster. Likewise, the port side of the aircraft would mirror this connectivity with the outboard and inboard clusters. To achieve consensus over the collective, naturally, the two inboard clusters would require a communication edge to formulate a spanning tree topology. Finally, often designers choose between scheduling versus allocating effectors in complex hybrid mixer schemes. Allocated groups communicat-

ing with scheduled groups will involve varying communication topologies and should be considered.

Also, in this work, a model of the vehicle control effectivity mapping is assumed to be known a priori. This perfect model is evaluated by each agent during the distributed optimization process[1]. Valuable future work would be to relax this assumption and incorporate non-perfect models with estimation techniques to accommodate error. Additionally, each individual effector currently employs a model of the global mapping for the vehicle. A geometrical distribution of effectors over an aerospace planform invites the research topic of local estimation at the agent scope, as well as local variability. In the latter case, a localized sensor suite may further improve the ability to refine the individual control effectivity mapping for deviations that occur with respect to location on the vehicle body.

The distributed control allocation method in this work solves a new optimization problem at every major time step in the simulation, which in the deca-rotor hovercraft simulation here, executes at a 25 hz frequency. At each step, the DSPC algorithm initializes all agent strategies to uniform probability distributions within the specified search bounds, governed by the effector position limits. Each optimization instance, therefore, is independent from the prior optimization[2]. Future research on a smart initialization feature may offer a substantial improvement in performance and algorithm convergence time. For example, rather than initializing all strategies to uniform distributions, the agent probability distributions could be seeded with a mean and variance to favor the unconstrained analytical weighted least squares

[1]The perfect control effectivity model at the agent scope represents the identical effectivity mapping used in the equations of motion for the hovercraft dynamical simulation.

[2]All parameters are independent in each DSPC instance other than the prior values of the effector positions used for the computation of effector rate limits.

(WLS) solution if a form of the linear **B** control effectivity matrix is available. The WLS solution with effector position and rate limit constraints is the computational bottleneck typically, and employing the initial unconstrained analytical solution may offer a rapid method to seed DSPC and yield superior convergence. Alternatively, if the **B** matrix is unknown, a smart initialization method could be explored with a hybrid use of the ISPC (Immediate Sampling Probability Collectives) method explored in this work. In this latter case, ISPC essentially seeds the DSPC probability search space with Gaussian distributions, which should offer a head start over a uniform distribution initialization.

The constrained optimization problem solved in this work includes both effector position and rate limits in the search for the minimizing solution to the L_2 objective. However, future work should also investigate the inclusion of an additional constraint for limiting the effective gain of the solution. Real time control allocation is known to include an effective gain, and may be dynamic and a function of failure condition, flight condition, and other factors. Provisions should be investigated to limit excessive gain amplification, especially in the case of failure, in order to prevent degradation of closed loop stability margin.

Future research potential exists in the direction of distributed control allocation as a means for run time assurance, an online method for guaranteeing safety and certification properties of a complex aerospace system. This topic is a commonly discussed avenue today in the field of next generation aerospace control systems, specifically with adaptive methodologies that involve nondeterministic behavior. Traditional software validation and verification becomes nearly infeasible in these circumstances, and a system that exhibits self-monitoring and the ability to detect and adapt to health issues may be the key. In this regard, the distributed control allocation methods developed in this work may not necessarily alleviate a centralized flight control ap-

proach, but instead, complement the system as a real time monitor and contingency option.

Finally, the distribution of the flight control process requires increased communication, however this bus does not have to be a physical entity, opening the door for a future research topic where multiple wireless agents reconfigure to achieve a team objective. For instance, multiple agents can work collectively to accomplish a task of moving a heavy object as the primary planform of a hovercraft. After attaching to the heavy load and continually communicating via a wireless network, the collective task is accomplished by distributed control allocation to lift, translate, and set down the load.

This section has provided a detailed set of recommended future research paths. For convenience, the following list provides a brief description of each of major topics delineated:

i. Other optimization objectives including noise and discontinuities, L_1 control allocation optimization, projection of actuator dynamical response
ii. Automated parameter tuning methods
iii. Adaptive sampling intervals
iv. Reduced agent network topologies and hybrid mixer schemes with scheduled vs allocated effectors
v. Objective estimation and variation at the local agent scope
vi. Smart initialization: analytical weighted least squares solution if available, or Immediate Sampling Probability Collectives method as DSPC seed
vii. Inclusion of gain limiting provisions into distributed control allocation method
viii. Distributed control allocation as a run time assurance system

APPENDIX A

Derivation of Equations of Motion for a Distributed Effector Suite

A.1 Equations of Motion for Distributed Effector Suites

This section presents the equations of motion for a distributed propulsion vehicle. Beginning with first principles for a rigid body, Newton's 2nd law of motion governs the motion of the vehicle with respect to a non-accelerating, inertial frame of reference as depicted in Fig. A.1 below and illustrated in detail in [99, 100] by Roskam and McRuer.

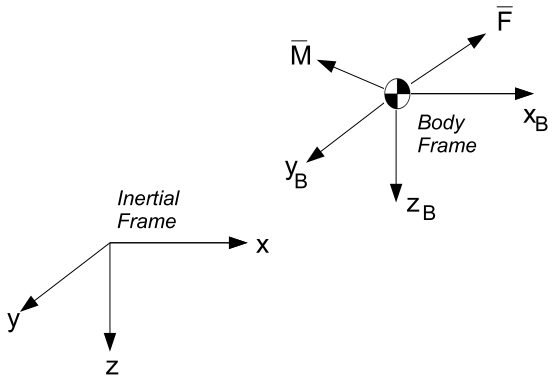


Figure A.1: External Action Vectors Defined with Respect to an Inertial Frame for Newtonian Equations of Motion of a Rigid Body

The external action vectors, acting on the rigid body, are categorized as translational force, $\vec{F}$, and angular torque or moment, $\vec{M}$, and are often conveniently represented (expressed) in the body frame of reference. By the 2nd law of Newton, the time rate of change of the translational momentum, $\vec{P} = mV_c$, equals the sum of the translational forces, $\vec{F}$ acting on the body, where m is the total mass of the body and $\vec{V_c}$ is the inertial velocity vector of the center of mass. Likewise, the time

rate of change of the rotational momentum, $\vec{H} = \mathbf{I}\vec{\omega}$, equals the sum of the rotational moments, $\vec{M}$ acting on the body, where $\mathbf{I}$ is the inertia matrix of the hovercraft.

$$\begin{aligned} \vec{F} = &\quad \frac{d}{dt}(\vec{P}) = \frac{d}{dt}(m\vec{V}_c) \\ \vec{M} = &\quad \frac{d}{dt}(\vec{H}) = \frac{d}{dt}(\mathbf{I}\vec{\omega}) \end{aligned} \tag{A.1}$$

A.1.1 Translational Dynamics

Considering N total infinitesimal elements of the complete rigid body vehicle, the j^{th} mass, δm_j, and resultant force on the element, $\delta\vec{F}_j$, is depicted in Fig. A.2. The total force on the vehicle is the summation over the number of infinitesimal elements as written below.

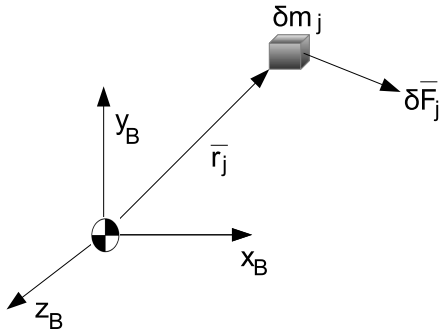


Figure A.2: Infinitesimal Mass Element of Rigid Body

$$\vec{F} = \sum_{j=1}^{N} \delta\vec{F}_j \tag{A.2}$$

Typically, the time rate of change of mass (i.e. $\dot{m}$ due to fuel burn for instance) of an atmospheric vehicle is neglected in a maneuvering and agility design study, since the more relevant stability and control dynamics occur at a significantly higher frequency [99, 100]. Dogan and Blake in [93] provide detailed derivation on the equations of motion for a vehicle with a time varying mass conglomerate, where fuel burn or

addition is accounted with a complex body of multiple components, beyond the scope of the present work. In the case of neglecting mass time variance, and considering only a single rigid body, Eq. (A.1) is further simplified to the regularly encountered Newton form with all vectors defined and expressed in the inertial frame as

$$\vec{F} = m\dot{\vec{V}} \tag{A.3}$$

where m represents the complete mass of the rigid body, and V with the dropped c subscript for brevity represents the inertial velocity vector of the center of mass.

In order to facilitate modeling of the vehicle for simulation and study, the equations are (i) transformed and expressed in the body frame of the vehicle; and (ii) abbreviated into matrix form using skew symmetric operations [127, 128]. The skew symmetric matrix definition $\mathbf{S}(\mathbf{a})$ enables writing $\mathbf{a} \times \mathbf{b} = -\mathbf{S}(\mathbf{a})\mathbf{b}$ and is defined as

$$\mathbf{S}(\mathbf{a}) = \begin{bmatrix} 0 & a_3 & -a_2 \\ -a_3 & 0 & a_1 \\ a_2 & -a_1 & 0 \end{bmatrix} \tag{A.4}$$

where $\mathbf{a} = [a_1 \; a_2 \; a_3]^T \in \Re^3$ is an expression of vector $\vec{a}$ in a coordinate frame of interest. Eq. (A.3) is now rewritten as

$$\mathbf{F} = m\dot{\mathbf{V}} \tag{A.5}$$

with $\mathbf{V} = [\dot{x} \; \dot{y} \; \dot{z}]^T \in \Re^3$ and $\mathbf{F} = [F_x \; F_y \; F_z]^T \in \Re^3$. The dynamics are transformed to a body frame expression using the general Euler equation for relative motion [129]

$$\dot{\mathbf{V}} = \dot{\mathbf{V}}_B - \mathbf{S}(\omega_B)\mathbf{V}_B \tag{A.6}$$

where $\mathbf{V}_B = [u \; v \; w]^T \in \Re^3$, and $\omega_B = [p \; q \; r]^T \in \Re^3$. Substituting into Eq. (A.5),

$$\mathbf{f}_B = m\left[\dot{\mathbf{V}}_B - \mathbf{S}(\omega_B)\mathbf{V}_B\right] \tag{A.7}$$

$\mathbf{f}_B$ represents the inertial force vector in Eq. (A.1) now expressed in the body frame of the vehicle and includes all external forces on the rigid body from aerodynamics, gravity, propulsion, and effectors. For the purpose of this research effort with an application to a distributed propulsive hovercraft, external forces are described as the summation of the inertial gravity vector $\mathbf{g}$ plus the $\mathbf{F}_B$ contribution of the propulsive effectors, $\mathbf{R}m\mathbf{g} + \sum_{j=1}^{N} \mathbf{F}j_B$, where $\mathbf{g} = [0\ 0\ g]^T$, and $\mathbf{R}$ is the 321 Euler Rotation Matrix from the Inertial Frame to the Body Frame as [1]

$$\mathbf{R} = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & \sin\phi\cos\theta \\ \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi & \cos\phi\cos\theta \end{bmatrix}$$

Including this description as the summation of body forces to facilitate an equation for a distributed effector suite, and rearranging to yield acceleration on the left hand side of the equation, the general vector form for the translational dynamics of the rigid body is

$$\dot{\mathbf{V}}_B = \mathbf{R}\mathbf{g} + \mathbf{S}(\omega_B)\mathbf{V}_B + \frac{1}{m}\mathbf{F}_B \tag{A.8}$$

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \mathbf{R}\mathbf{g} + \begin{bmatrix} 0 & r & -q \\ -r & 0 & p \\ q & -p & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} + \frac{1}{m}\sum_{j=1}^{N} \mathbf{F}j_B \tag{A.9}$$

A.1.2 Rotational Dynamics

In a similar approach as depicted for the translational dynamics , the rotational dynamics are expanded for an infinitesimal element mass over the complete rigid body, enabling accommodation for multiple external forces and torques, as detailed in [93, 99, 100]. The time rate of change of the angular momentum vector, $\dot{\vec{H}}$ in

[1] Prior indices are redefined to represent the j^{th} of N total effectors in a distributed effector suite.

Eq. (A.1) is transformed to be expressed in the body frame of the vehicle, again using the general Euler equation for relative motion [129]

$$\mathbf{M}_B = \dot{\mathbf{H}}_B \ - \mathbf{S}(\omega_B)\mathbf{H}_B \tag{A.10}$$

Approximating the body frame hovercraft inertia matrix as time invariant and expanding $\dot{\mathbf{H}}_B$ with the chain rule,

$$\begin{aligned} \dot{\mathbf{H}}_B = & \ \dot{\mathbf{I}}_{\mathbf{B}}\omega_B + \mathbf{I}_{\mathbf{B}}\dot{\omega}_B \\ = & \ \mathbf{I}_{\mathbf{B}}\dot{\omega}_B \end{aligned} \tag{A.11}$$

Substitution into Eq. (A.10) yields

$$\mathbf{M}_B = \mathbf{I}_{\mathbf{B}}\dot{\omega}_B \ - \mathbf{S}(\omega_B)\mathbf{H}_B \tag{A.12}$$

The external moments on the hovercraft are written as the summation of cross products of each engine geometrical lever arm, $\mathbf{d}_j$, with respect to the center of mass with the thrust vector, yielding the inertial torque contribution from each engine, $\tau\mathbf{j}_{\mathbf{B}}$,

$$\mathbf{M}_B = \sum_{j=1}^{N} \left[-\mathbf{S}(\mathbf{d}_j)\mathbf{F}j_B + \tau j_B\right] \tag{A.13}$$

Writing in skew symmetric matrix form, the angular acceleration of the vehicle expressed in the body frame is

$$\dot{\omega}_B = \mathbf{I}_B^{-1}\left\{\mathbf{S}(\omega_B)\mathbf{I}_B\omega_B + \sum_{j=1}^{N}\left[-\mathbf{S}(\mathbf{d}_j)\mathbf{F}j_B + \tau j_B\right]\right\} \tag{A.14}$$

Finally, the expanded equation describes the governing Rotational Dynamics equations of the hovercraft

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \mathbf{I}_B^{-1}\left(\begin{bmatrix} 0 & r & -q \\ -r & 0 & p \\ q & -p & 0 \end{bmatrix}\mathbf{I}_B\begin{bmatrix} p \\ q \\ r \end{bmatrix} + \sum_{j=1}^{N}\left[-\mathbf{S}(\mathbf{d}_j)\mathbf{F}j_B + \tau j_B\right]\right) \tag{A.15}$$

A.1.3 Translational Kinematics

The translational kinematics are written as the body frame represented velocity vector transformed by the Euler 321 rotation matrix transpose from body frame to inertial frame as

$$\mathbf{V} = \mathbf{R}^T \mathbf{V}_B \tag{A.16}$$

A.1.4 Rotational Kinematics

The rotational kinematics of the hovercraft vehicle are defined with respect to the inertial frame as

$$\dot{\mathbf{R}} = \mathbf{S}(\omega_B)\mathbf{R} \tag{A.17}$$

The left hand side of this equation is the derivative with respect to time of the Euler 321 Rotation matrix $\dot{\mathbf{R}} \in \Re^{3x3}$ and yields a set of nine scalar equations which can be algebraically solved for the inertial angular velocity vector (Euler attitude rates).

$$\begin{aligned} \dot{\phi} &= p + \tan\theta(q\sin\theta + r\cos\phi) \\ \dot{\theta} &= q\cos\phi - r\sin\phi \\ \dot{\psi} &= \frac{1}{\cos\phi}(r\cos\phi + q\sin\phi) \end{aligned} \tag{A.18}$$

APPENDIX B

Weighted Least Squares Analytic Solution to L_2 Optimization

B.1 Weighted Least Squares Analytic Solution to L_2 Linear Control Allocation

This section presents the derivation of the analytic solution to the L_2 linear control allocation problem where the mixed optimization objective describes the error with respect to the requested virtual command, and the magnitude of effector position required with respect to a desired position, $\mathbf{u}_d$. This solution is valid for cases that do not exceed actuator constraints. The mixed optimization objective and Euclidean L_2 norm as described in Ch. 3 as [108]

$$L_2 = ||\mathbf{W_u}(\mathbf{u} - \mathbf{u}_d)||_2 + \gamma||\mathbf{W_v}(\mathbf{Bu} - \mathbf{v})||_2 \tag{B.1}$$

can be written in vector and matrix operation format,

$$L_2 = \frac{1}{2}\left[(\mathbf{u} - \mathbf{u}_d)^T\mathbf{W_u}(\mathbf{u} - \mathbf{u}_d) + \gamma(\mathbf{Bu} - \mathbf{v})^T\mathbf{W_v}(\mathbf{Bu} - \mathbf{v})\right] \tag{B.2}$$

Expanding,

$$\begin{aligned} L_2 =& \frac{1}{2}\left(\mathbf{u}^T\mathbf{W_u}\mathbf{u} - \mathbf{u}^T\mathbf{W_u}\mathbf{u}_d - {\mathbf{u}_d}^T\mathbf{W_u}\mathbf{u} + {\mathbf{u}_d}^T\mathbf{W_u}\mathbf{u}_d\right) \\ &+ \frac{1}{2}\gamma\left(\mathbf{u}^T\mathbf{B}^T\mathbf{W_v}\mathbf{Bu} - \mathbf{u}^T\mathbf{B}^T\mathbf{W_v}\mathbf{v} - \mathbf{v}^T\mathbf{W_v}\mathbf{Bu} + \mathbf{v}^T\mathbf{W_v}\mathbf{v}\right) \\ =& \frac{1}{2}\left[\mathbf{u}^T\mathbf{W_u}\mathbf{u} - \mathbf{u}^T\mathbf{W_u}\mathbf{u}_d - \left({\mathbf{u}_d}^T\mathbf{W_u}\mathbf{u}\right)^T + {\mathbf{u}_d}^T\mathbf{W_u}\mathbf{u}_d\right] \\ &+ \frac{1}{2}\gamma\left[\mathbf{u}^T\mathbf{B}^T\mathbf{W_v}\mathbf{Bu} - \mathbf{u}^T\mathbf{B}^T\mathbf{W_v}\mathbf{v} - \left(\mathbf{v}^T\mathbf{W_v}\mathbf{Bu}\right)^T + \mathbf{v}^T\mathbf{W_v}\mathbf{v}\right] \end{aligned} \tag{B.3}$$

Due to the symmetric property of the effector and acceleration penalty matrices, $\mathbf{W_u} = {\mathbf{W_u}}^T$ and $\mathbf{W_v} = {\mathbf{W_v}}^T$,

$$\begin{aligned} L_2 =& \frac{1}{2}\left(\mathbf{u}^T\mathbf{W_u}\mathbf{u} - 2\mathbf{u}^T\mathbf{W_u}\mathbf{u}_d + {\mathbf{u}_d}^T\mathbf{W_u}\mathbf{u}_d\right) \\ &+ \frac{1}{2}\gamma\left(\mathbf{u}^T\mathbf{B}^T\mathbf{W_v}\mathbf{Bu} - 2\mathbf{u}^T\mathbf{B}^T\mathbf{W_v}\mathbf{v} + \mathbf{v}^T\mathbf{W_v}\mathbf{v}\right) \end{aligned} \tag{B.4}$$

As a necessary condition for optimality, $\nabla_u L_2 = 0$. From Eq. (B.4),

$$\begin{aligned} \nabla_u L_2 = 0 =& \mathbf{W_u}\mathbf{u} - \mathbf{W_u}\mathbf{u}_d + \gamma\mathbf{B}^T\mathbf{W_v}\mathbf{Bu} - \gamma\mathbf{B}^T\mathbf{W_v}\mathbf{v} \\ =& \mathbf{W_u}(\mathbf{u} - \mathbf{u}_d) + \gamma\mathbf{B}^T\mathbf{W_v}(\mathbf{Bu} - \mathbf{v}) \end{aligned} \tag{B.5}$$

Solving for the L_2 minimizing control effector solution, $\mathbf{u}$,

$$\mathbf{W_u u}_d + \gamma \mathbf{B}^T \mathbf{W_v B v} = \left[\mathbf{W_u} + \gamma \mathbf{B}^T \mathbf{W_v B}\right] \mathbf{u} \tag{B.6}$$

Therefore,

$$\mathbf{u} = \left[\mathbf{W_u} + \gamma \mathbf{B}^T \mathbf{W_v B}\right]^{-1} \left(\mathbf{W_u u}_d + \gamma \mathbf{B}^T \mathbf{W_v v}\right) \tag{B.7}$$

assuming $\left[\mathbf{W_u} + \gamma \mathbf{B}^T \mathbf{W_v B}\right]$ is invertible. Finally, as a sufficient condition for optimality, ${\nabla_u}^2 L_2$ should be positive definite.

$${\nabla_u}^2 L_2 = \mathbf{W_u} + \gamma \mathbf{B}^T \mathbf{W_v B} \tag{B.8}$$

Given positive definiteness of $\mathbf{W_u}$, γ, $\mathbf{B}$, and $\mathbf{W_v}$, clearly ${\nabla_u}^2 L_2 > 0$.

A notional L_2 objective surface is depicted for a two-effector example in Fig. B.1 with $\mathbf{B} = \begin{bmatrix} 0.6892 & 0.7482 \end{bmatrix}$, $\mathbf{W_u} = I \in \Re^{2\times 2}$, and $W_v = \gamma = 1$. Fig. B.2 presents a contour plot of the L_2 cost with numerical isolines. Given a desired virtual command, $v = 1$, the satisfying solution $u_2 = \frac{1}{B_{12}}(v - B_{11}u_1)$ can be computed exhaustively for the effector domain of interest, $u_1 \in [-10, 10]$. Finally, the analytic solution in Eq. (B.7) depicts the required u_1 and u_2 effector positions to satisfy the desired virtual command request while minimizing the L_2 objective cost.

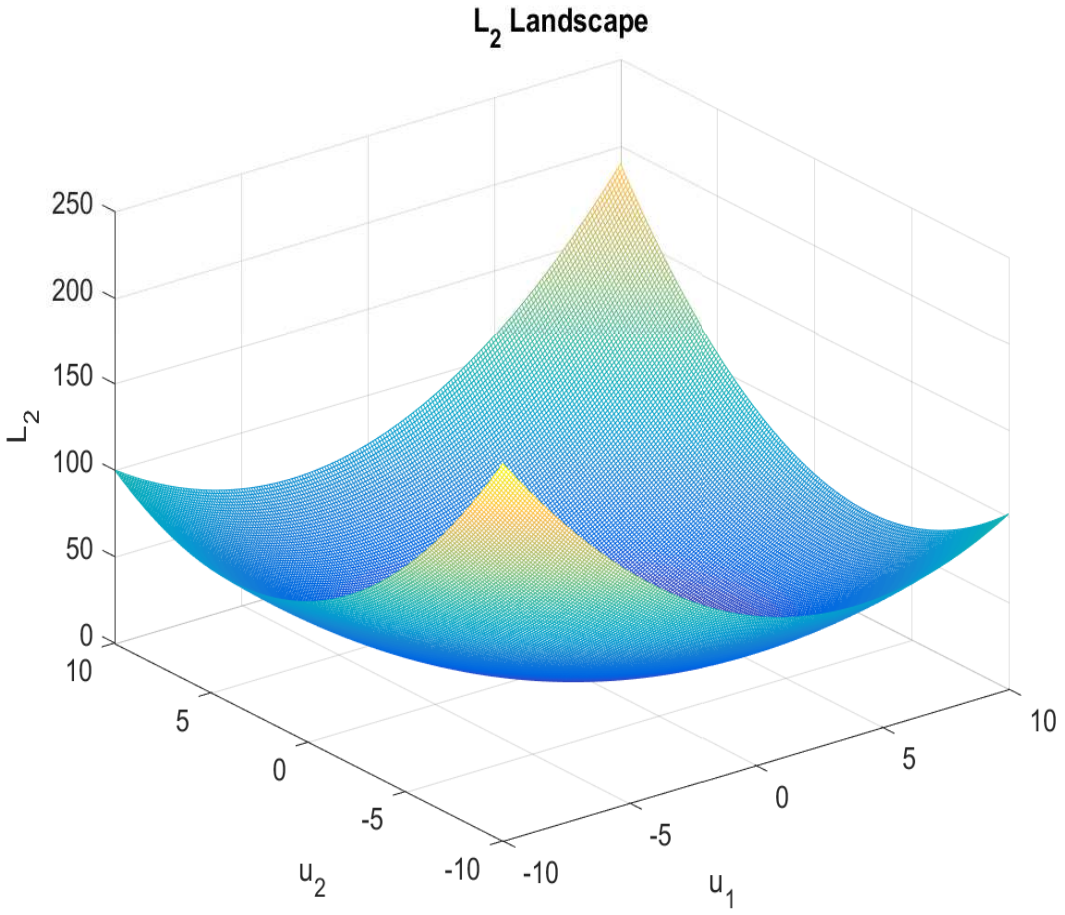


Figure B.1: Notional Two-Effector Example of L_2 Control Allocation Objective

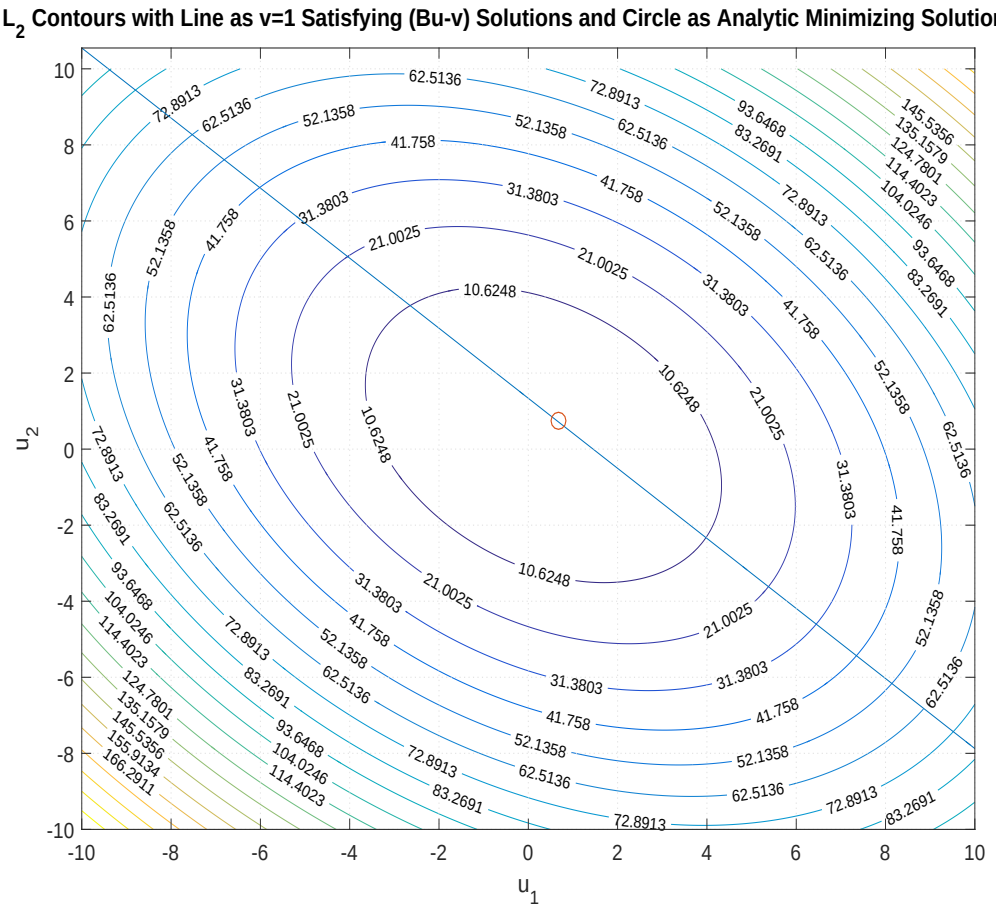


Figure B.2: Notional Two-Effector Example L_2 Contour Plot with Analytic Solution

APPENDIX C

Nearest Newton Second Order Simplex Constrained Gradient Descent

C.1 Nearest Newton Second Order Simplex Constrained Gradient Descent

This section presents the derivation of the Nearest Newton Second Order Simplex Constrained Gradient Descent (NN SOSCGD) update law for the agent probability distribution.

Given the Maxent Lagrangian defined as

$$L(\mathbf{p}, T) = E_{\mathbf{p}}(G) - T[S(\mathbf{p})] \tag{C.1}$$

where

$$E_{\mathbf{p}}(G) \equiv \int_{\mathbf{D}} \mathbf{p}(\mathbf{x}) G(\mathbf{x}) d\mathbf{x} \tag{C.2}$$

denotes the expected value of the objective function G, evaluated over the joint probability distribution $\mathbf{p}$ over a continuous domain Lebesgue integral, $\mathbf{D}$. Writing in discrete form, to facilitate implementation into an embedded software system, the expected value is written as

$$E_{\mathbf{p}}(G) \equiv \sum_{j=1}^{n_{moves}} p(x_j) G(x_j) \tag{C.3}$$

The Shannon relative entropy in Eq. (C.1) is written as

$$S(\mathbf{p}) = -\int_{\mathbf{D}} \mathbf{p}(\mathbf{x}) \ln \mathbf{p}(\mathbf{x}) d\mathbf{x} \tag{C.4}$$

with discrete form,

$$S(p) = -\sum_{j=1}^{n_{moves}} p(x_j) \ln p(x_j) \tag{C.5}$$

For illustration of the gradient with a simple scalar form, for the interim, consider a discrete point sum with $n_{moves} = 2$. The expected value of the probability distribution can then be written as

$$E_{\mathbf{p}}(G) = p(x_1) G(x_2) + p(x_2) G(x_2) \tag{C.6}$$

and the Shannon relative entropy is

$$S(p) = -p(x_1)\ln p(x_1) - p(x_2)\ln p(x_2) \tag{C.7}$$

Substituting Eqs. (C.6-C.7) into Eq. (C.2), the Maxent Lagrangian is written in expanded form as

$$L(\mathbf{p},T) = p(x_1)G(x_1) + p(x_2)G(x_2) - T\left[-p(x_1)\ln p(x_1) - p(x_2)\ln p(x_2)\right] \tag{C.8}$$

Defining $\nabla_{p(x_j)}L(\mathbf{p},T) \equiv \frac{\partial L(\mathbf{p},T)}{\partial p(x_j)}$, and writing the gradient

$$\begin{aligned}\nabla_{p(x_1)}L(\mathbf{p},T) &= G(x_1) - T\left[-\ln p(x_1) - p(x_1)\frac{1}{p(x_1)}\right] \\ &= G(x_1) + T\left[\ln p(x_1) + 1\right]\end{aligned} \tag{C.9}$$

Likewise, the Hessian is defined as $\nabla^2_{p(x_j)}L(\mathbf{p},T) \equiv \frac{\partial^2 L(\mathbf{p},T)}{\partial p(x_j)^2}$

$$\nabla^2_{p(x_1)}L(\mathbf{p},T) = T\frac{1}{p(x_1)} \tag{C.10}$$

Crassidis and Junkins describe the second order Gauss-Newton parameter optimization method employed here in [108] where the Lagrangian $L(\mathbf{p},T)$ is approximated about $p(x_o)$ with a second order Taylor series as

$$L(\mathbf{p},T) \approx L[p(x_o),T] + \Delta p\nabla_{p(x_o)}L(\mathbf{p},T) + \frac{1}{2}\Delta p^2\nabla^2_{p(x_o)}L(\mathbf{p},T) \tag{C.11}$$

where $\Delta p \equiv p^*(x_o) - p(x_o)$. As a necessary condition of optimality, $\nabla_{\Delta p}L(\mathbf{p},T) = 0$,

$$\nabla_{\Delta p}L(\mathbf{p},T) = 0 = \nabla_{p(x_o)}L(\mathbf{p},T) + \Delta p\nabla^2_{p(x_o)}L(\mathbf{p},T) \tag{C.12}$$

and a sufficient condition for a global minimum is a positive definite Hessian $\nabla^2_{p(x_j)}L(\mathbf{p},T) > 0$, which has already been shown in Eq. (C.10) with a positive T and valid probability distribution, $\mathbf{p}$. From Eq. (C.12), the Gauss-Newton update law is derived as

$$\Delta p = -\left[\nabla^2_{p(x_o)}L(\mathbf{p},T)\right]^{-1}\nabla_{p(x_o)}L(\mathbf{p},T) \tag{C.13}$$

Substituting the gradient and Hessian and expanding Δp about $x = x_1$,

$$p^*(x_1) - p(x_1) = -\left[\frac{T}{p(x_1)}\right]^{-1} \{G(x_1) + T\left[\ln p(x_1) + 1\right]\} \tag{C.14}$$

Therefore,

$$\begin{aligned} p^*(x_1) &= p(x_1) - \left[\frac{p(x_1)}{T}\right] \{G(x_1) + T\left[\ln p(x_1) + 1\right]\} \\ &= p(x_1) - p(x_1)\left[\frac{G(x_1)}{T} + \ln p(x_1) + 1\right] \end{aligned} \tag{C.15}$$

Wolpert and Bieniawski in [57, 58, 59, 60] include a proportional step size, α in the update law to throttle the agent from updating the strategy too fast. Also a normalization constraining factor $\lambda = S(\mathbf{p}) - 1 - \frac{1}{T}E_{\mathbf{p}}(G)$ is included to constrain the strategy to simplex as

$$\begin{aligned} p^*(x_1) &= p(x_1) - \alpha p(x_1)\left[\frac{G(x_1)}{T} + \ln p(x_1) + 1 + \lambda\right] \\ &= p(x_1) - \alpha p(x_1)\left[\frac{G(x_1)}{T} + \ln p(x_1) + 1 + S(\mathbf{p}) - 1 - \frac{1}{T}E_{\mathbf{p}}(G)\right] \\ &= p(x_1) - \alpha p(x_1)\left[\frac{1}{T}\{G(x_1) - E_{\mathbf{p}}(G)\} + \ln p(x_1) + S(\mathbf{p})\right] \end{aligned} \tag{C.16}$$

Finally, the update law is written with the minimizing solution to the $KL(\mathbf{p}||\mathbf{q})$ distance in

$$\begin{aligned} KL(\mathbf{p}||\mathbf{q}) =& S(\mathbf{p}||\mathbf{q}) - S(\mathbf{p}) \\ =& -\sum_{i=1}^{n_{agent}} \int_{\mathbf{D}} \mathbf{p}(\mathbf{x}) \ln\left[\mathbf{q_i}(\mathbf{x_i})\right] d\mathbf{x} \end{aligned} \tag{C.17}$$

with $\mathbf{q_i} = \mathbf{p_i}, \forall i$. Therefore, the final second order simplex constrained gradient descent (SOSCGD) probability distribution update law can now be written in general for the i^{th} action of the j^{th} agent,

$$q_j^*(x_i) = q_j(x_i) - \alpha q_j(x_i)\left[\frac{1}{T}\{G(x_i) - E_{\mathbf{q}}(G)\} + \ln q_j(x_i) + S(\mathbf{q_j})\right] \tag{C.18}$$

APPENDIX D

Agent Sampling Interval Refinement

D.1 Agent Sampling Interval Refinement

Kulkarni et al in [61, 82, 83, 84, 85] present a domain shrinking method for refining the sampling interval in a modified DSPC optimization process for improved algorithm performance. Throughout iteration, the i^{th} agent sampling interval is reduced and centered with respect to the best action, x_i^*. Kulkarni defines the sampling domain of the agent as $\psi_\mathbf{i} \in [\psi_i^{lower}, \psi_i^{upper}]$ and introduces a new evaluation step in the baseline DSPC process to update $\psi_\mathbf{i}$ as a function of x_i^* as the following [61]

$$\begin{aligned} \psi_i^{lower} &= x_i^* - \lambda_{down}||\psi_i^{upper} - \psi_i^{lower}||_2 \\ \psi_i^{upper} &= x_i^* + \lambda_{down}||\psi_i^{upper} - \psi_i^{lower}||_2 \\ \psi_\mathbf{i} &\leftarrow [\psi_i^{lower}, \psi_i^{upper}] \end{aligned} \tag{D.1}$$

where $0.0 < \lambda_{down} \leq 1.0$. In this work, the predetermined number of iterations set for DSPC combined with this refinement method essentially locks the final sampling domain $\psi_\mathbf{i}$ to a fixed interval given a specified λ_{down}, regardless of convergence or algorithm performance[1]. As an alternative, a sampling interval refinement scheme accounts for the variance in the strategy as a recursive update. Additionally, the sampling domain is prevented from increasing beyond the original specified search bounds. A proportional gain K_σ is used to refine the level of refinement throughout the optimization as follows

$$\begin{aligned} \psi_i^{lower} &\leftarrow \max\left[x_i^* - K_\sigma \int_D [(x_i - \mu_i)^2 p_i(x_i)]dx_i, \psi_i^{lower}\right] \\ \psi_i^{upper} &\leftarrow \min\left[x_i^* + K_\sigma \int_D [(x_i - \mu_i)^2 p_i(x_i)]dx_i, \psi_i^{upper}\right] \\ \psi_\mathbf{i} &\leftarrow [\psi_i^{lower}, \psi_i^{upper}] \end{aligned} \tag{D.2}$$

[1]This work employs a deterministic number of iterations for DSPC in order to ensure the distributed control allocation method meets an allotted computational processing time for a stability augmentation system loop closure.

For comparison of the two refinement methods, the two dimensional quadric example from Ch. 4 is repeated here with $\min_{\mathbf{x},\mathbf{y}} G(\mathbf{x},\mathbf{y}) = (\mathbf{x}-50)^2 + (\mathbf{y}+50)^2$. DSPC is employed with $T = 0.1$, $\alpha = 0.01$, $\gamma = 0.02$, and with two agents, $n_{agent} = 2$, with $\mathbf{x_1} \equiv \mathbf{x}$, $\mathbf{x_2} \equiv \mathbf{y}$. The stopping condition is set to a maximum step size of 15 iterations, with 5000 bins, $n_{moves_i} = 5000$, defining the agent mixed strategy, $\mathbf{q_i}(\mathbf{x_i})$. 100 samples per iteration ($m = 100$) are drawn from the current strategy. Fig. D.1 depicts the K_σ refinement approach in Eq. (D.2) exhibits an improved sampling domain in this example, given the parameter set and number of allowed iterations.

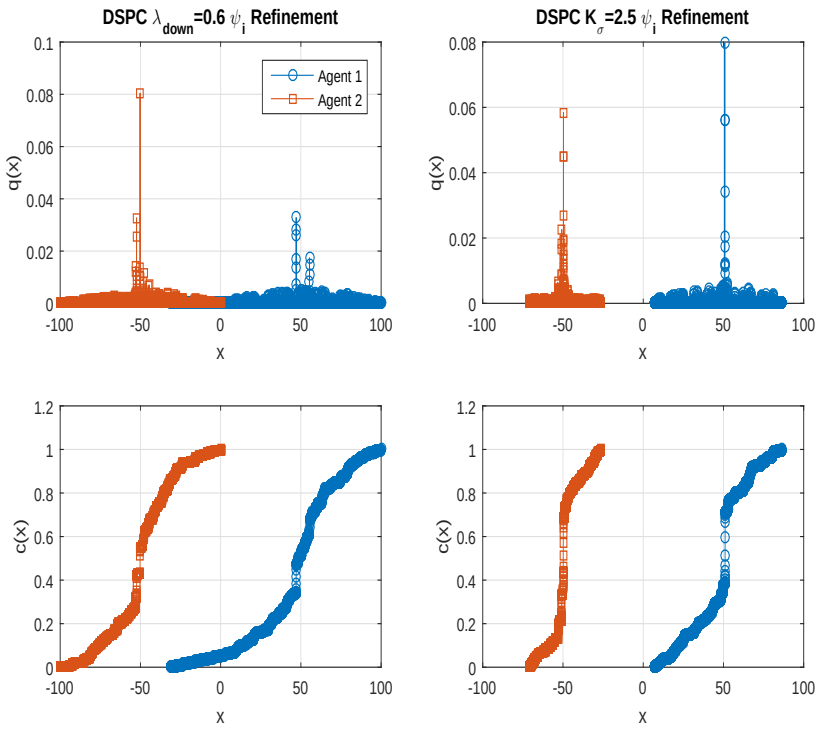


Figure D.1: DSPC Agent Sampling Interval Refinement Comparison for $\lambda_{down} = 0.6$ Method and $K_\sigma = 2.5$ Method

APPENDIX E

Multiple Extrema

E.1 Multiple Extrema

This section demonstrates the delayed sampling probability collectives method on a one dimensional example with two global minima. Consider the one dimensional optimization problem,

$$\min_{\mathbf{x}} G(\mathbf{x}) = (\mathbf{x} - 10)\sin[.65(\mathbf{x} - 10)] \tag{E.1}$$

defined over the domain of $\mathbf{x} \in [0, 20]$. DSPC is employed with $T = 1.0$, $\alpha = 0.2$, $\gamma = 0.3$, and with one agent, $n_{agent} = 1$, with $\mathbf{x_1} \equiv \mathbf{x}$. The stopping condition is set to a maximum step size of 50 iterations, with 3000 bins, $n_{moves_i} = 3000$, defining the agent mixed strategy, $\mathbf{q_i}(\mathbf{x_i})$. 100 samples per iteration ($m = 100$) are drawn from the current strategy[1]. Fig. E.1 depicts the final results of the optimized strategy with the correctly identified extrema x^* equal to 2.47 and 17.5.

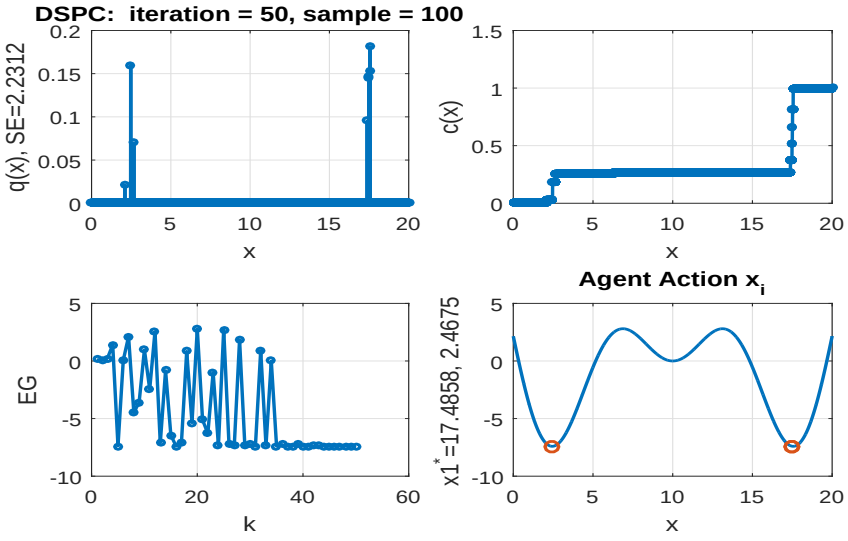


Figure E.1: DSPC for Objectives with Multiple Extrema

[1]In this example, the agent is configured to return samples from the final optimized strategy $\mathbf{q_i}(\mathbf{x_i})$ rather than solely the expected value of the strategy $E[\mathbf{q_i}(\mathbf{x_i})]$. Clearly, returning $E[\mathbf{q_i}(\mathbf{x_i})]$ is an insufficient approach for objectives with multiple extrema.

APPENDIX F

Agent Network Connectivity

F.1 Agent Network Connectivity

In discrete time consensus theory, the state of the i^{th} networked agent updates as a difference equation which is a function of the prior state and the states of the surrounding neighbors,

$$x_i(k+1) = x_i(k) + u_i(k) \tag{F.1}$$

$u_i(k)$ is described as the normalized local voting protocol controlling the consensus of the collective, as given by Lewis et al in [130]. Expanding the local voting protocol,

$$u_i(k) = \frac{1}{1+d_i} \sum_{j \in N_i} a_{ij}[x_j(k) - x_i(k)] \tag{F.2}$$

d_i is the in-degree of node i (i.e. the number of input connections from other agents) with a neighborhood of dependencies N_i. The element of the adjacency matrix, a_{ij}, represents the connection between agent i and agent j, and finally, the $x_i(k)$ and $x_j(k)$ represent the state of agent i and j at discrete time k, respectively. Factoring $x_i(k)$ and rewriting Eq. (F.2),

$$u_i(k) = \frac{1}{1+d_i} \left[-x_i(k) \sum_{j \in N_i} a_{ij} + \sum_{j \in N_i} a_{ij} x_j(k) \right] \tag{F.3}$$

Given the in-degree matrix, $\mathbf{D} \in \Re^{n_{agent} \times n_{agent}}$ of the collective system as

$$\mathbf{D} = \sum_{j \in N_i} a_{ij} \tag{F.4}$$

the normalized local voting protocol is written in matrix form for the collective with adjacency matrix $\mathbf{A} \in \Re^{n_{agent} \times n_{agent}}$ as

$$\begin{aligned} \mathbf{u}(k) &= (\mathbf{I} + \mathbf{D})^{-1} \left[-\mathbf{D}\mathbf{x} + \mathbf{A}\mathbf{x} \right] \\ &= -(\mathbf{I} + \mathbf{D})^{-1} \left[\mathbf{D} - \mathbf{A} \right] \mathbf{x} \\ &= -(\mathbf{I} + \mathbf{D})^{-1} \mathbf{L}\mathbf{x} \end{aligned} \tag{F.5}$$

where $\mathbf{u}, \mathbf{x} \in \Re^{n_{agent}}$ and $\mathbf{L} = \mathbf{D} - \mathbf{A}$ is described as the Laplacian matrix of the network, with eigenvalues describing the consensus stability of the continuous system.

Considering Eq. (F.5) in matrix form with these definitions, and substituting the normalized local voting protocol yields

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{x}(k) + -(\mathbf{I}+\mathbf{D})^{-1}\left[\mathbf{L}\right]\mathbf{x}(k) \\ &= \left[\mathbf{I} - (\mathbf{I}+\mathbf{D})^{-1}\mathbf{L}\right]\mathbf{x}(k) \\ &= (\mathbf{I}+\mathbf{D})^{-1}\left[(\mathbf{I}+\mathbf{D}) - \mathbf{L}\right]\mathbf{x}(k) \\ &= (\mathbf{I}+\mathbf{D})^{-1}\left[(\mathbf{I}+\mathbf{D}) - (\mathbf{D}-\mathbf{A})\right]\mathbf{x}(k) \\ &= (\mathbf{I}+\mathbf{D})^{-1}\left[\mathbf{I}+\mathbf{A}\right]\mathbf{x}(k)\end{aligned} \tag{F.6}$$

Denoting $\mathbf{F} = (\mathbf{I}+\mathbf{D})^{-1}\left[\mathbf{I}+\mathbf{A}\right] \in \Re^{n_{agent}\times n_{agent}}$ as the discrete time synchronization matrix, the global dynamics for the collective are then described by the closed loop equation

$$\mathbf{x}(k+1) = \mathbf{F}\mathbf{x}(k) \tag{F.7}$$

Given solely the network topology with adjacency matrix, $\mathbf{A}$, the consensus dynamics and settling time of a network can be considered by computing the in-degree matrix $\mathbf{D}$, Laplacian matrix $\mathbf{L}$, and discrete time synchronization matrix $\mathbf{F}$. Time to consensus for a network topology is described by the inverse real component of the Fiedler or algebraic connectivity eigenvalue, $\tau = \Re[\lambda_2^{-1}]$, where λ_2 is computed as the second smallest eigenvalue in the normalized Laplacian matrix, $\bar{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A}$ [130][1].

Fig. F.1 presents the complete graph network topology used in this study with 10 distributed agents, $n_{agent} = 10$. The adjacency matrix for this system is given by

[1]The smallest eigenvalue must be at 0.0 in order for the collective to reach steady state consensus, while the Fiedler eigenvalue governs the lowest frequency dynamical response of the system [130].

$$\mathbf{A_{complete}} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Ones represent connectivity between agents where information is shared. For example, row one indicates that the first agent sends the expected value of the local strategy to all other agents. Column one indicates the first agent receives expectations of strategies for all other agents. The zero diagonal terms indicate the agent does not self disagree, also reflected in Eq. (F.3) which equals zero for $j = i$.

For simulation with a notional DSPC implementation, an agent probability update gain is assumed as a multiplicative term on all edges of the adjacency, $\alpha\mathbf{A}$. The adjacency matrix governs the distributed control allocation method in terms of the information flow over the collective, where the information in this case represents the optimal action for all agents to perform in order to minimize the L_2 objective. Consensus results are depicted in Fig. F.2 for the complete graph topology with $\alpha = 0.5$, yielding $\tau = 0.9$ secs. In this example, the system converges to consensus rapidly in less than three iterations, bounding the best case response for distributed control allocation[2].

[2]The consensus analysis techniques in this appendix can be used to explore if the network connectivity of the collective is a limiting factor for the consideration of n_{steps}, α strategy proportional update gain, and other DSPC parameters.

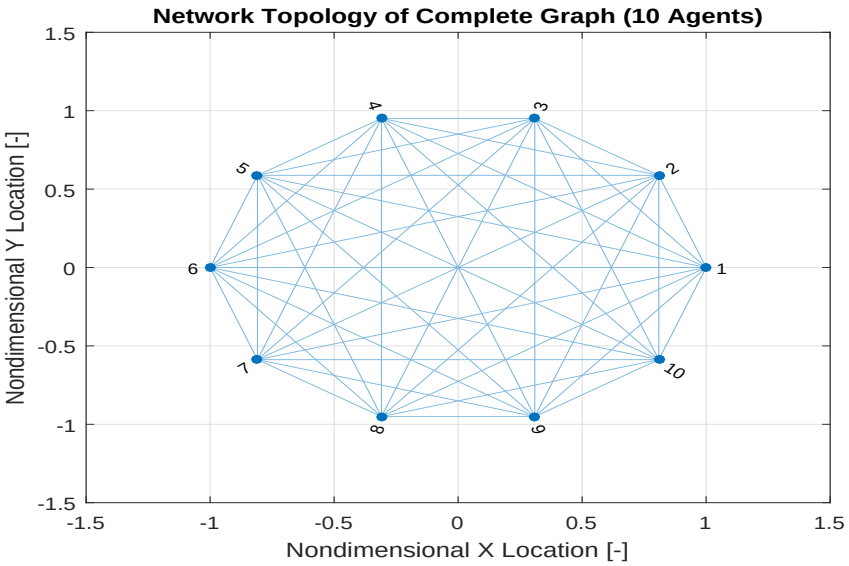


Figure F.1: Network Topology of Complete Graph (10 Agents)

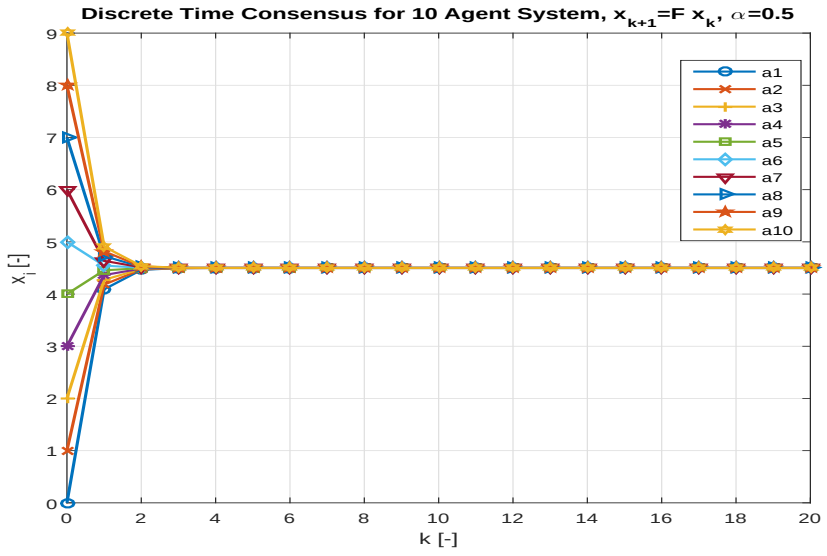


Figure F.2: Discrete Time Consensus Simulation for 10 Agent Complete Graph System

Two additional example topologies are included below to illustrate the potential impact on consensus of a distributed system and the importance of sufficient iteration: (i) spanning tree with adjacency $\mathbf{A_{tree}}$ and $\tau = 5.5$ secs as presented in Figs. F.3-F.4; and (ii) directed gossip ring with adjacency $\mathbf{A_{gossip}}$ and $\tau = 5.3$ secs in Figs. F.5-F.6.

$$\mathbf{A_{tree}} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{A_{gossip}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

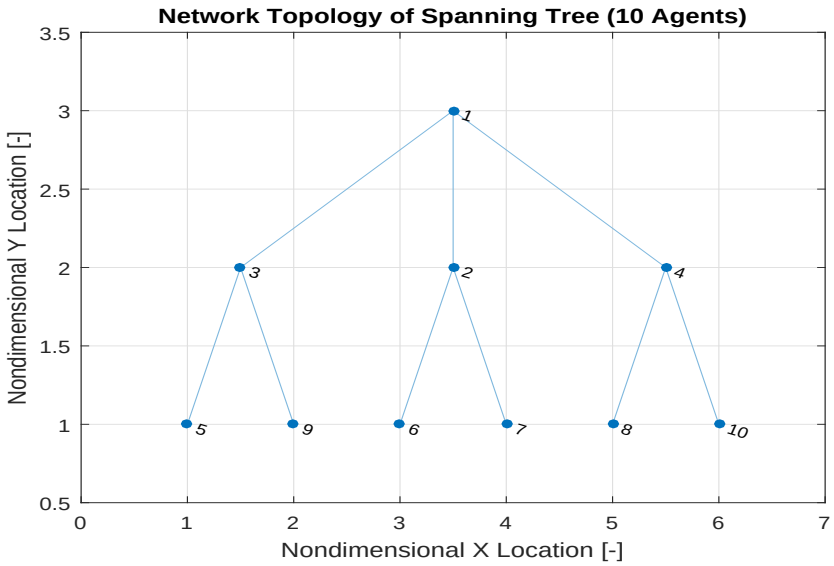


Figure F.3: Network Topology of Spanning Tree Graph (10 Agents)

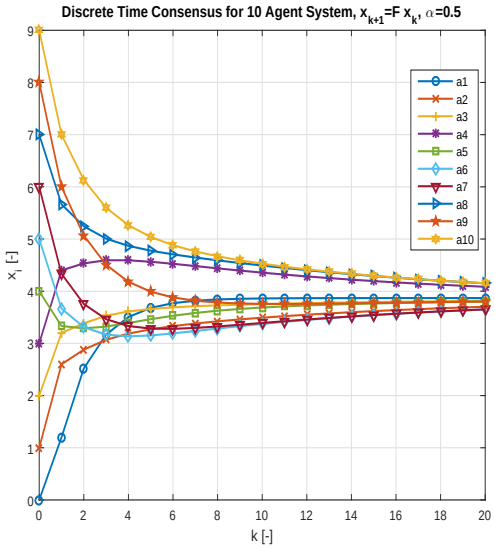


Figure F.4: Discrete Time Consensus Simulation for 10 Agent Spanning Tree Graph System

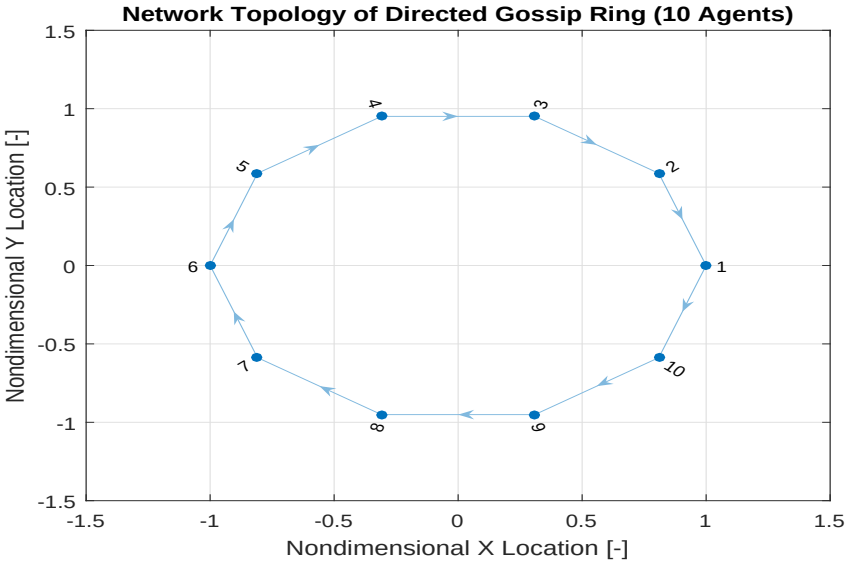


Figure F.5: Network Topology of Directed Gossip Ring Graph (10 Agents)

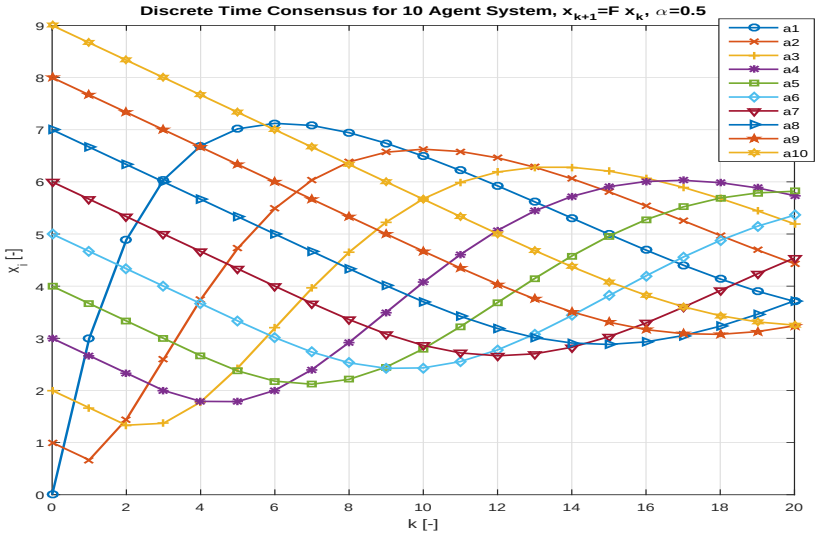


Figure F.6: Discrete Time Consensus Simulation for 10 Agent Directed Gossip Ring Graph System

REFERENCES

[1] P. Likins, "Robert e. roberson: A personal tribute," *Mechanics Based Design of Structures and Machines*, vol. 17, no. 2, pp. 131–133, 1989.

[2] R. W. Longman, "Generalized approach to gravity-gradient stabilization of gyrostat satellites." 1969.

[3] P. T. Kabamba, "Contributions to reduced order control theory," Ph.D. dissertation, Columbia University, 1981.

[4] A. Dogan, "Guidance strategies for microburst escape," Ph.D. dissertation, The University of Michigan, Ann Arbor, MI, 2000.

[5] N. D. S. University. (2015) Mathematics genealogy project. [Online]. Available: http://genealogy.math.ndsu.nodak.edu/mission.php

[6] A. T. Wick, J. R. Hooker, C. J. Hardin, and C. H. Zeune, "Integrated aerodynamic benefits of distributed propulsion," 2015.

[7] R. E. Everly and D. C. Limmer, "Cost-effectiveness analysis of aerial platforms and suitable communication payloads," Ph.D. dissertation, Monterey, California: Naval Postgraduate School, 2014.

[8] J. Rosero, J. Ortega, E. Aldabas, and L. Romeral, "Moving towards a more electric aircraft," *Aerospace and Electronic Systems Magazine, IEEE*, vol. 22, no. 3, pp. 3–9, 2007.

[9] A. S. Gohardani, G. Doulgeris, and R. Singh, "Challenges of future aircraft propulsion: A review of distributed propulsion technology and its potential application for the all electric commercial aircraft," *Progress in Aerospace Sciences*, vol. 47, no. 5, pp. 369–391, 2011.

[10] M. Moore, "Nasa distributed electric propulsion research," Public Presentation, February 2015.

[11] M. Friswell, "The prospects for morphing aircraft," in *Smart Structures and Materials (SMART09), IV ECCOMAS Thematic Conference*, 2009, pp. 175–188.

[12] A. Suleman, A. Costa, P. Moniz, and C. Crawford, "An adaptive aeroelastic wing," in *Smart Structures*. Springer, 2001, pp. 44–54.

[13] L. N. Cattafesta III and M. Sheplak, "Actuators for active flow control," *Annual Review of Fluid Mechanics*, vol. 43, pp. 247–272, 2011.

[14] B. Obradovic, "Modeling and simulation of the flight dynamics of morphing wing aircraft," 2010.

[15] B. L. Stevens and F. L. Lewis, *Aircraft control and simulation*. John Wiley & Sons, 2003.

[16] M. Bodson, "Evaluation of optimization methods for control allocation," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*. Montreal, 2001, pp. 6–9.

[17] K. Bordignon and J. Bessolo, "Control allocation for the x-35b," in *2002 Biennial International Powered Lift Conference and Exhibit*, 2002, pp. 5–7.

[18] O. Härkegård, "Backstepping and control allocation with applications to flight control," 2003.

[19] H. Kim, H. S. Jeong, K. T. Chong, and D. J. Lee, "Dynamic modeling and control techniques for multi-rotor flying robots," *Transactions of the Korean Society of Mechanical Engineers A*, vol. 38, no. 2, pp. 137–148, 2014.

[20] V. Hrishikeshavan, "Experimental investigation of shrouded rotor micro air vehicle in hover and in edgewise gusts," 2011.

[21] I. Kroo, F. Prinz, M. Shantz, P. Kunz, G. Fay, S. Cheng, T. Fabian, and C. Partridge, "The mesicopter: A miniature rotorcraft concept–phase ii interim report," *Stanford University, USA*, 2000.

[22] R. Oung and R. DAndrea, "The distributed flight array," *Mechatronics*, vol. 21, no. 6, pp. 908–917, 2011.

[23] M. Hepperle, "Electric flight–potential and limitations," *NATO-OTAN MP-AVT-209-09*, 2012.

[24] A. Stall, J. Bevirt, M. D. Moore, J. Fredericks, and N. K. Borer, "Drag reduction through distributed electric propulsion," in *14th AIAA Aviation Technology, Integration and Operations Conference*, 2014, pp. 16–20.

[25] J. R. Hooker, A. Wick, C. Zeune, and A. Agelastos, "Over wing nacelle installations for improved energy efficiency," in *31st AIAA Applied Aerodynamics Conference*, 2013, p. 2920.

[26] W. G. Barnwell, "Distributed actuation and sensing on an uninhabited aerial vehicle," 2003.

[27] A. K. Jha and J. N. Kudva, "Morphing aircraft concepts, classifications, and challenges," in *Smart structures and materials*. International Society for Optics and Photonics, 2004, pp. 213–224.

[28] A. Adnan, *Molecular simulations of deformation, failure and fracture of nanostructured materials*. ProQuest, 2008.

[29] B. Holm-Hansen, C. Atkinson, J. Benarek, E. Burnett, L. Nicolai, and H. Youssef, "Envelope expansion of a flexible flying wing by active flutter suppression," *Proceedings of AUVSIs Unmanned Systems North America*, 2010.

[30] J. Beranek, L. Nicolai, M. Buonanno, E. Burnett, C. Atkinson, B. Holm-Hansen, and P. Flick, "Conceptual design of a multi-utility aeroelastic demonstrator," in

13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference, Fort Worth, TX, 2010, pp. 2194–2208.

[31] E. Pendleton, K. E. Griffin, M. W. Kehoe, and B. Perry, "A flight research program for active aeroelastic wing technology," *AIAA Paper*, no. 96-1574, pp. 2263–2273, 1996.

[32] J. Buffington, P. Chandler, and M. Pachter, "On-line system identification for aircraft with distributed control effectors," *International Journal of Robust and Nonlinear Control*, vol. 9, no. 14, pp. 1033–1049, 1999.

[33] A. Das and F. L. Lewis, "Distributed adaptive control for synchronization of unknown nonlinear networked systems," *Automatica*, vol. 46, no. 12, pp. 2014–2021, 2010.

[34] W. Ren and R. W. Beard, *Distributed consensus in multi-vehicle cooperative control*. Springer, 2008.

[35] A. Çela, M. B. Gaid, X.-G. Li, and S.-I. Niculescu, *Optimal Design of Distributed Control and Embedded Systems*. Springer Science & Business Media, 2013.

[36] C. M. Elliott, "Distributed multi-agent systems-a literature survey and inquisitive discussion," in *Aerospace Conference, 2014 IEEE*. IEEE, 2014, pp. 1–7.

[37] D. S. Acharya and N. L. Nath, "Applications of multi agent systems in control engineering: A state-of-the-art survey," 2014.

[38] K. Tagesson, P. Sundstrom, L. Laine, and N. Dela, "Real-time performance of control allocation for actuator coordination in heavy vehicles," in *Intelligent Vehicles Symposium, 2009 IEEE*. IEEE, 2009, pp. 685–690.

[39] M. W. Oppenheimer, D. B. Doman, M. Bolender *et al.*, "Control allocation for over-actuated systems," in *Control and Automation, 2006. MED'06. 14th Mediterranean Conference on*. IEEE, 2006, pp. 1–6.

[40] C. Feng, N.-g. Ding, Y.-l. He, G.-y. Xu, and F. Gao, "Control allocation algorithm for over-actuated electric vehicles," *Journal of Central South University*, vol. 21, pp. 3705–3712, 2014.

[41] T. Johansen, T. Fossen, S. P. Berge *et al.*, "Constrained nonlinear control allocation with singularity avoidance using sequential quadratic programming," *Control Systems Technology, IEEE Transactions on*, vol. 12, no. 1, pp. 211–216, 2004.

[42] T. A. Johansen, "Optimizing nonlinear control allocation," in *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, vol. 4. IEEE, 2004, pp. 3435–3440.

[43] M. A. Bolender and D. B. Doman, "Nonlinear control allocation using piecewise linear functions," *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 6, pp. 1017–1027, 2004.

[44] M. Bodson and S. A. Frost, "Control allocation with load balancing," in *AIAA Guidance, Navigation, and Control Conference & Exhibit*, 2009.

[45] D. G. Luenberger, *Introduction to linear and nonlinear programming*. Addison-Wesley Reading, MA, 1973, vol. 28.

[46] J. J. Craig, *Introduction to robotics: mechanics and control*. Pearson Prentice Hall Upper Saddle River, 2005, vol. 3.

[47] A. Bowling, "Analysis of robotic manipulator dynamic performance: Acceleration and force capablities," Ph.D. dissertation, Stanford University, 1998.

[48] K. A. Bordignon, "Constrained control allocation for systems with redundant control effectors," Ph.D. dissertation, Virginia Polytechnic Institute and State University, 1996.

[49] W. C. Durham, "Constrained control allocation," *Journal of Guidance, Control, and Dynamics*, vol. 16, no. 4, pp. 717–725, 1993.

[50] R. Venkataraman, M. Oppenheimer, and D. Doman, "A new control allocation method that accounts for effector dynamics," in *Aerospace Conference, 2004. Proceedings. 2004 IEEE*, vol. 4. IEEE, 2004, pp. 2710–2715.

[51] M. D. Tandale, K. Subbarao, J. Valasek, and M. R. Akella, "Structured adaptive model inversion control with actuator saturation constraints applied to tracking spacecraft maneuvers," in *American Control Conference, 2004. Proceedings of the 2004*, vol. 2. IEEE, 2004, pp. 1031–1036.

[52] G. Ducard and M.-D. Hua, "Discussion and practical aspects on control allocation for a multi-rotor helicopter," in *Conference on Unmanned Aerial Vehicle in Geomatics*, 2011, pp. 1–6.

[53] H. Tol, C. De Visser, E. Van Kampen, and Q. Chu, "Nonlinear multivariate spline-based control allocation for high-performance aircraft," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 6, pp. 1840–1862, 2014.

[54] J. M. Buffington, M. A. Niestroy, C. M. Ha, P. Wei, and R. Eberhardt, "Robust nonlinear aircraft flight control," DTIC Document, Tech. Rep., 2001.

[55] M. L. Glaze, "The design and implementation of a gui-based control allocation toolbox in the matlab® environment," Ph.D. dissertation, Virginia Polytechnic Institute and State University, 1998.

[56] T. L. Boullion and P. L. Odell, *Generalized inverse matrices*. John Wiley & Sons, 1971.

[57] S. R. Bieniawski, "Distributed optimization and flight control using collectives," Ph.D. dissertation, Stanford University, 2005.

[58] D. H. Wolpert and K. Tumer, "An introduction to collective intelligence," *arXiv preprint cs/9908014*, 1999.

[59] D. H. Wolpert and S. Bieniawski, "Distributed control by lagrangian steepest descent," in *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, vol. 2. IEEE, 2004, pp. 1562–1567.

[60] S. Bieniawski, D. Wolpert, and I. Kroo, "Discrete, continuous, and constrained optimization using collectives," in *Proceedings of 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, New York*, 2004.

[61] A. J. Kulkarni, K. Tai, and A. Abraham, *Probability collectives: a distributed multi-agent system approach for optimization*. Springer, 2014.

[62] D. Rajnarayan, D. Wolpert, and I. Kroo, "Optimization under uncertainty using probability collectives," in *10th AIAA/ISSMO MAO Conf*, 2006.

[63] D. H. Wolpert and D. G. Rajnarayan, "Parametric learning and monte carlo optimization," *arXiv preprint arXiv:0704.1274*, 2007.

[64] R. H. Myers, *Response Surface Methodology*. Edwards Brothers, 1976.

[65] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[66] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi *et al.*, "Optimization by simulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.

[67] E. Aarts and J. Korst, "Simulated annealing and boltzmann machines," 1988.

[68] C.-R. Hwang, "Simulated annealing: theory and applications," *Acta Applicandae Mathematicae*, vol. 12, no. 1, pp. 108–111, 1988.

[69] L. Davis, "Genetic algorithms and simulated annealing," 1987.

[70] I. Szita and A. Lörincz, "Learning tetris using the noisy cross-entropy method," *Neural computation*, vol. 18, no. 12, pp. 2936–2941, 2006.

[71] R. Y. Rubinstein and D. P. Kroese, *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer Science & Business Media, 2013.

[72] D. E. Booth, "The cross-entropy method," *Technometrics*, vol. 50, no. 1, pp. 92–92, 2008.

[73] P. Larranaga and J. A. Lozano, *Estimation of distribution algorithms: A new tool for evolutionary computation*. Springer Science & Business Media, 2002, vol. 2.

[74] J. A. Lozano, R. Sagarna, and P. Larrañaga, "Parallel estimation of distribution algorithms," in *Estimation of Distribution Algorithms*. Springer, 2002, pp. 129–145.

[75] M. Pelikan, M. W. Hauschild, and F. G. Lobo, "Estimation of distribution algorithms," in *Springer Handbook of Computational Intelligence*. Springer, 2015, pp. 899–928.

[76] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Machine learning*, vol. 3, no. 2, pp. 95–99, 1988.

[77] M. Mitchell, *An introduction to genetic algorithms*. MIT press, 1998.

[78] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm intelligence*, vol. 1, no. 1, pp. 33–57, 2007.

[79] M. Clerc, *Particle swarm optimization*. John Wiley & Sons, 2010, vol. 93.

[80] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of machine learning*. Springer, 2011, pp. 760–766.

[81] C. F. Lee and D. H. Wolpert, "Product distribution theory for control of multi-agent systems," in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*. IEEE Computer Society, 2004, pp. 522–529.

[82] A. J. Kulkarni, "A distributed multi-agent system approach for solving constrained optimization problems using probability collectives," Ph.D. dissertation, 2012.

[83] A. J. Kulkarni, I. Kale, K. Tai, and S. K. Azad, "Discrete optimization of truss structure using probability collectives," in *Hybrid Intelligent Systems (HIS), 2012 12th International Conference on.* IEEE, 2012, pp. 225–230.

[84] A. J. Kulkarni, I. R. Kale, and K. Tai, "Probability collectives for solving truss structure problems," in *10th World Congress on Structural and Multidisciplinary Optimization, paper*, no. 5395, 2013.

[85] A. Kulkarni, I. Kale, and K. Tai, "Probability collectives for solving discrete and mixed variable problems," *International Journal of Computer Aided Engineering and Technology*, 2014.

[86] X. Zhang, W. Yu, J. Liang, and B. Liu, "Entropy regularization for coordinated target assignment," in *Proceedings of 3rd IEEE Conference on Computer Science and Information Technology*, 2010, pp. 165–169.

[87] M. H. A. Mehr and B. H. Khalaj, "A distributed probability collectives optimization method for multicast in cdma wireless data networks," in *Wireless Communication Systems, 2007. ISWCS 2007. 4th International Symposium on.* IEEE, 2007, pp. 617–621.

[88] G. S. Ryder and K. Ross, "A probability collectives approach to weighted clustering algorithms for ad hoc networks." in *Communications and Computer Networks*, 2005, pp. 94–99.

[89] D. Wolpert, N. Antoine, S. Bieniawski, and I. Kroo, "Fleet assignment using collective intelligence," in *Proceedings of the 42nd AIAA Aerospace Science Meeting Exhibit*, 2004.

[90] C.-F. Huang and B.-R. Chang, "Adaptive probabilistic learning by collectives in dynamic environments," in *Granular Computing (GrC), 2010 IEEE International Conference on*. IEEE, 2010, pp. 229–234.

[91] ——, "Probability collectives in dynamic environments: A study of controlling the balance between exploration and exploitation of the search," in *Technologies and Applications of Artificial Intelligence (TAAI), 2010 International Conference on*. IEEE, 2010, pp. 187–194.

[92] C.-F. Huang, B.-R. Chang, and D.-W. Cheng, "Tracking extrema in dynamic environments using probability collectives multi-agent systems," in *Machine Learning and Cybernetics (ICMLC), 2011 International Conference on*, vol. 1. IEEE, 2011, pp. 33–39.

[93] A. Dogan and W. Blake, *Flight mechanics of aerial refueling*. AIAA, 2015.

[94] S. A. Erturk, "Performance analysis, dynamic simulation and control of mass-actuated airplane," Ph.D. dissertation, The University of Texas at Arlington, 2016.

[95] K. B. Ariyur and M. Krstic, *Real-time optimization by extremum-seeking control*. John Wiley & Sons, 2003.

[96] H. S. Wolko and J. D. Anderson, "The wright flyer: an engineering perspective," 1983.

[97] M. Spick, *Modern Fighting Aircraft F-14 Tomcat*. Salamander Books Limited, 1985.

[98] J. A. Simpson, E. S. Weiner *et al.*, *The Oxford english dictionary*. Clarendon Press Oxford, 1989, vol. 2.

[99] J. Roskam, *Airplane flight dynamics and automatic flight controls*. DARcorporation, 1995.

[100] D. T. McRuer, D. Graham, and I. Ashkenas, *Aircraft dynamics and automatic control*. Princeton University Press, 2014.

[101] R. S. Garfinkel and G. L. Nemhauser, *Integer programming*. Wiley New York, 1972, vol. 4.

[102] C. Elliot and A. Dogan, "Investigating nonlinear control architecture options for aerial refueling," in *Proceedings of the AIAA Atmospheric Flight Mechanics Conference, Toronto, Canada*, 2010.

[103] H. P. Lee, J. W. Clemens, and H. M. Youssef, "Dynamic inversion flight control design for aircraft with non-minimum phase response," SAE Technical Paper, Tech. Rep., 2011.

[104] A. Snell, "Decoupling of nonminimum phase plants and application to flight control," in *AIAA Guidance, Navigation, and Control Conference and Exhibit, Monterey, California*, 2002.

[105] M. W. Oppenheimer and D. B. Doman, "Control of an unstable, nonminimum phase hypersonic vehicle model," in *Aerospace Conference, 2006 IEEE*. IEEE, 2005, pp. 7–pp.

[106] M. Brodecki and K. Subbarao, "Autonomous formation flight control system using in-flight sweet-spot estimation," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 6, pp. 1083–1096, 2014.

[107] J. Wedershoven, "Analysis of nonlinear dynamic inversion based control law designs," *Unpublished M. Sc. Thesis, Faculty of Aerospace Engineering, Delft University of Technology*, 2010.

[108] J. L. Crassidis and J. L. Junkins, *Optimal estimation of dynamic systems*. CRC press, 2011.

[109] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

[110] E. T. Jaynes, "Information theory and statistical mechanics," *Physical review*, vol. 106, no. 4, p. 620, 1957.

[111] C. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, pp. 379–423,623–656, 1948.

[112] S. S. Haykin, *Neural networks and learning machines*. Pearson Education Upper Saddle River, 2009, vol. 3.

[113] L. Devroye, "Sample-based non-uniform random variate generation," in *Proceedings of the 18th conference on Winter simulation*. ACM, 1986, pp. 260–265.

[114] D. Rajnarayan, I. Kroo, and D. H. Wolpert, "Probability collectives for optimization of computer simulations," in *Proceedings of the AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2007.

[115] J. P. Kleijnen, "Statistical techniques in simulation," 1974.

[116] K. J. DeMars, R. H. Bishop, and M. K. Jah, "Entropy-based approach for uncertainty propagation of nonlinear dynamical systems," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 4, pp. 1047–1057, 2013.

[117] A. Björck, *Numerical methods for least squares problems*. Siam, 1996.

[118] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

[119] I. Chern, "Finite difference methods for solving differential equations," *Department of Mathematics, book, National Taiwan University*, 2009.

[120] S.-P. Han, "A globally convergent method for nonlinear programming," *Journal of optimization theory and applications*, vol. 22, no. 3, pp. 297–309, 1977.

[121] A. Das, K. Subbarao, and F. Lewis, "Dynamic inversion with zero-dynamics stabilisation for quadrotor control," *IET control theory & applications*, vol. 3, no. 3, pp. 303–314, 2009.

[122] L. Sankar, "Helicopter aerodynamics and performance, hover prediction methods," 2008. [Online]. Available: soliton.ae.gatech.edu/people/lsankar/AE6070.Spring2008/Part1.ppt

[123] B. E. Wake and L. N. Sankar, "Solutions of the navier-stokes equations for the flow about a rotor blade," *Journal of the American Helicopter Society*, vol. 34, no. 2, pp. 13–23, 1989.

[124] W. Johnson, *Helicopter theory*. Courier Corporation, 2012.

[125] G. J. Leishman, *Principles of helicopter aerodynamics with CD extra*. Cambridge University Press, 2006.

[126] R. W. Prouty, *Helicopter performance, stability, and control*, 1995.

[127] J. Waishek, A. Dogan, and W. Blake, "Derivation of the dynamics equations of receiver aircraft in aerial refueling," *Journal of guidance, control, and dynamics*, vol. 32, no. 2, pp. 586–598, 2009.

[128] C. Elliott, "Improving receiver station-keeping in aerial refueling by formulating tanker motion as disturbance," Master's thesis, The University of Texas at Arlington, Arlington, TX, December 2009.

[129] A. Bedford, W. L. Fowler, and Y. Ahmad, *Engineering mechanics: dynamics*. Prentice Hall New Jersey, 2005.

[130] F. L. Lewis, H. Zhang, K. Hengster-Movric, and A. Das, *Cooperative control of multi-agent systems: optimal and adaptive design approaches*. Springer Science & Business Media, 2013.

BIOGRAPHICAL STATEMENT

Christopher Michael Elliott was born in Houston, Texas on July 11, 1976. He received a Bachelor of Science degree in Aerospace Engineering from the University of Texas at Austin in December 1998. Concurrent with undergraduate studies, he worked at the Institute of Advanced Technology as a Research Engineer in the Electro-Magnetic Launch Projectiles (EMLP) Dynamics group. After graduation, he worked as a Software Engineer Consultant with the Radx Corporation out of Plano, Texas until accepting a position in October 1999 with the United Space Alliance as an Aerospace Engineer in the International Space Station Program Office at the Johnson Space Center in Houston, Texas. Transferring to the Automation and Robotics NASA directorate in August 2000 with Lockheed Martin Space Operations, he worked in the Multi-use Remote Manipulator Development Facility (MRMDF) until February 2003. During this period, Chris began graduate studies at the University of Houston at Clear Lake in Software Engineering. In February 2003, he transferred to Lockheed Martin Aeronautics Company in Fort Worth, Texas and joined the F16 Control Law Analysis and Design Flight Dynamics team. Along with the career change from the space industry to the atmospheric flight industry, Chris began a new graduate degree path at the University of Texas at Arlington in Aerospace Engineering with an emphasis on Flight Mechanics in the Spring of 2006. Today he continues full time employment with the Lockheed Martin Aeronautics Company with the Skunk Works Flight Controls and the Skunk Works Quantum Science Research teams. He also teaches part time as an Adjunct Professor at the University of Texas at Arlington in the Mechanical and Aerospace Department. He has published multiple technical papers at the American

Institute of Aeronautics and Astronautics (AIAA) as a senior professional member and plans to continue postdoctoral research with Dr. Atilla Dogan subsequent to a PhD graduation in December 2016.