Robust Synthesis Methods for Cooperative Systems

by

BARIS TANER

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 2022

İlham kaynağım olan babam Atilla Taner,

koşullar ne olursa olsun beni yalnız bırakmayan kardeşim Özgür Taner,

hayallerime ulaşmamı mümkün kılan, güçlü kadın, annem Emine Taner

ile bana yeni hayaller veren ve kıymetli varlığıyla beni yücelten sevgili eşim Ruken'e.

Acknowledgements

This dissertation is a result of hard work nourished by the support and guidance of significant individuals. First and most importantly, I thank my advisor, Dr. Kamesh Subbarao, for letting me be a part of a prominent group of engineers and demonstrating authentic leadership and precise guidance. It is a privilege to learn from him, who has excellent knowledge and an even greater interest in sharing it.

I want to thank my committee members Dr. Alan P. Bowling, Dr. Bo P Wang, Dr. Animesh Chakravarthy, and Dr. Yijing Xie, for their valuable advice and inspiration, which makes this dissertation a more valuable one. I am indebted for the support given by the Office of Naval Research (via award no. N00014-18-1-2215) to my Ph.D. research.

I want to thank the administrative staff of the Mechanical and Aerospace Engineering Department, Lanie Gordon, Wendy Ryan, and Ayesha Fatima, for their endless patience, support, and understanding.

I wholeheartedly thank my colleagues, Dr. Rajnish Bhusal, Dr. Diganta Bhattacharjee, Dr. Ameya Godbole, Kamalkumar Bharatkumar Mehta (Kamal), Abel Martinez Martinez, Dr. Katiyayni Balachandran (Kati), Denish Baman, Tsung-Liang Liu (Edward), Jinaykumar Nitinkumar Patel (Jinay), and Cem Kaya for being a second family and making Aerospace Systems Lab (ASL) a home and ELB 213 a welcoming place. Thank you all for sharing your culture, food, and vision and showing impeccable skill and dedication. I want to thank my lab mate Rajnish individually for making everything easier than it looks and giving endless support. Moreover, I would like to thank Kamal for his great support and friendship. I will miss every moment I spent with you all.

Lastly, I want to thank my family for their love, and support in my mother tongue because they deserve to know how much I appreciate their presence in attaining this degree. Beni kendi çocuklarından ayırmayan teyzelerim Müjgan, Mübeccel, ve eniştem Zeynel'e, kendime örnek aldığım ağabeyim Cenk'e, ablam Duygu'ya, ağabeyim Serdar'a ve son olarak dayım Fadıl, eşi Nurgül ablam ve kadeşim Behiç Mert'e verdikleri sonsuz destekler için teşekkür ederim.

December 7, 2022

Abstract

Robust Synthesis Methods for Cooperative Systems

Baris Taner, Ph.D.

The University of Texas at Arlington, 2022

Supervising Professor: Kamesh Subbarao

Cooperative systems, like any other dynamic systems, suffer in performance because of uncertainty yet there is an added layer of uncertainty due to the communication among agents. Therefore, analytic solution to these problems are hard if not impossible. With the advancements in the linear and non-linear methods, i.e. linear matrix inequalities (LMI) and non-linear transformations, robust performance analysis and controller synthesis for cooperative systems can be reformulated as optimization problems with LMI constraints as has been done in the last two decades. Another aspect of the problem becomes visible as the cooperative system grows larger and that necessitates faster solution methodologies to solve the aforementioned problem. Based on this context, the general objectives of this research are to develop computationally efficient analysis and synthesis methods for cooperative systems with uncertainties, which are

- Develop computationally efficient linear parameter varying (LPV) and Linear Time Invariant (LTI) synthesis framework and regarding tailored optimization techniques for cooperative systems, which suffer performance because of uncertainties.
  - develop computationally efficient linear parameter varying controller synthesis method that accommodates uncertainty analysis in a distributed fashion and

provide a framework to synthesize a robust cooperative system starting from a single agent

- develop a robust cooperative system synthesis method that consider uncertainty analysis in edge weight synthesis
- develop a cooperative system synthesis method that is formulated in distributed fashion to improve computational efficiency and suitable for distributed optimization.
- develop implementation strategies for cooperative synthesis methods to state of the art applications.

The methods developed in this dissertation are verified using numerical simulations using the short-period dynamics of an aircraft as an application to benchmark the computational efficiency of the LMI-based methods. On top of that, the cooperative synthesis methodology is implemented on a cooperative docking application and a bipedal locomotion application through Model Predictive Control (MPC).

This dissertation develops a framework for a systematic design of robust controllers to guarantee desired output performance for an interconnected group of multi-input-multi-output dynamic systems. The framework enables the design of a robust Linear Parameter Varying (LPV) controller for all individual vehicles and the interconnected group to account for uncertainties associated with the individual vehicles and the interconnections. In this dissertation, the robust controller design methodology for the individual and cooperative systems is implemented in a nested manner to enhance performance. The application of the nested robust controller is distributed wherein well-known robust performance analysis is adopted and modified. The controller synthesis methodology is developed on top of the same performance condition. Nested robust LPV controllers are synthesized for agents of the cooperative system. Then the cooperative system of these agents is interconnected using a connection topology that suffers time delays. A cooperative controller is designed using two methods described in this work in a nested fashion. Robustness against given

uncertainties is studied with an integral quadratic constraint (IQC) framework. A benchmark is drawn by comparing conventional lumped and the developed distributed method in terms of time efficiency.

Improving the robust performance of a cooperative system using cooperative controllers is a method. However, it increases the complexity of the synthesis due to the added states, and as the number of systems grows higher, the problem becomes intractable. A solution to this is to synthesize a cooperative system to improve its robust performance. Therefore the next problem studied in this work is implementing a non-linear programming method to synthesize edge weights of an adjacency matrix for a cooperative system using bi-linear matrix inequalities, which suffer uncertainties. First, convex-concave decompositions are used on the bi-linear matrix inequality constraints for nominal H_∞ synthesis. Then this method is improved to consider uncertainties using the IQC framework. Agents composing the cooperative system are represented as a linear time-invariant single input, single output systems, which share their output information to achieve consensus. The topology of the cooperative system is predefined, and edge weights are defined as functions of a variable.

Following a synthesis strategy that brings the best local robust performance of a cooperative system without adding the complexity of controllers is valuable, as, without loss of generality, these controllers have at least as many states as the agent. However, more efficient ways exist to extract the best local performance than synthesizing a cooperative system in lumped fashion. A more efficient way is to introduce distributed synthesis methodologies. Based on this context, this work further studies a distributed edge weight synthesis of a cooperative system for a fixed topology to improve H_∞ performance, considering that disturbances are injected at interconnection channels. Performance metrics for lumped and distributed methods are common; therefore, constraints related to performance for both methods are similar. However, the connection between agents of the cooperative system is defined in a distributed fashion in terms of additional synthesis constraints,

which constructs the optimization problem. Then this problem is cast into a linear matrix inequality problem by replacing the original cooperative system with an equivalent ideal cooperative system. Derivations of the method rely on the dissipative system framework. The proposed method provides an upper bound for the induced $\mathcal{L}_2$ norm of the original lumped cooperative system while reducing the computation time. A comparison of computation time illustrates the advantage of the proposed method against the lumped counterpart.

As presented above, the implementation strategy for cooperative synthesis ideology is presented in terms of a fast-slow MPC. The MPC is built with task prioritization to perform docking maneuvers on cooperative systems. The studied method allows agents and a single agent to perform a docking maneuver. In addition, agents give different priorities to a specific subset of shared states. In this way, overall degrees of freedom to achieve the docking task are distributed among various subsets of the task space. Fast-slow model predictive control strategy uses non-linear and linear model predictive control formulations such that docking is handled as a non-linear problem until agents are close enough, where direct transcription is calculated using the Euler discretization method. During this phase generated trajectory is tracked with a linear model predictive control. Then linear model predictive control performs the sensitive close proximity motion to finish docking. The proposed strategy is illustrated in a case study, where quadcopter docks on a non-holonomic rover using a leader-follower topology.

Finally, this dissertation presents a graph theoretic modeling and trajectory optimization technique for a biped robot named ASLB. This method utilizes a cooperative control framework to divide the state propagation and trajectory optimization of the lumped multi-body model of the robot into cooperative multi-bodies. The non-linear robot model is linearized at the current state, and states are propagated using the discrete newton Euler method. In addition, robot dynamics, contact location kinematics, and external forces are represented with respect to the body frame. This allows the cooperative quadratic

optimization method to handle trajectory optimization for ASLB, a highly non-linear system with large degrees of freedom.

Table of Contents

List of Illustrations

List of Tables

Chapter 1

Introduction

Finding an optimally robust cooperative system to uncertainties, especially with varying dynamic properties, is a task that requires the integration of various analytic and numeric frameworks. The frameworks, as mentioned earlier, are based on modeling the varying nature and uncertainty of the system in order to cast it into an optimization problem, where a locally optimal robust solution can be provided. These improved frameworks for general dynamic systems are also suitable for analysis and synthesis for cooperative systems. Based on this context, this research aims to develop analysis and synthesis methods to improve the robust performance of cooperative systems with uncertainties in a computationally efficient way and provide a systematic framework. In addition, this dissertation illustrates application strategies of cooperative synthesis methods for state of the art applications such as bipedal walking and cooperative docking.

1.1 Literature Review

Coordination among a group of agents has been a topic of long-standing interest. Over the past years, considerable work has been devoted to the design of cooperative control strategies among the agents for various applications such as formation flight [3], satellite clustering [4], unmanned vehicle [5, 6] and space applications [7, 8]. Furthermore, CSs are also used in wireless sensor networks, where distributed sensory data is being processed as in [9], and sensory information is merged as in [10]. In addition to that, CSs also fit in infrastructure applications such as AC power grid [11] and water distribution networks [12], where sources and sinks are in large numbers and widely distributed (see

[13–16] and references therein for further applications). Some of these applications cluster a few agents and form a small CS, while the others make large-scale CS, which might be geographically separated and life-critical [17]. As a further note, real-world dynamic systems with cooperating agents occur on weighted Graphs such as neural and biochemical networks [18–20]. Depending on the application, weights can be interpreted as conductance [21] or strength of the communication, where a formation control strategy is given for vertical takeoff and landing vehicles [22].

Recently, the assessment of stability and performance of a CS has been widely done by convex optimization; specifically, semi-definite programming (SDP), where computational algorithms are solving a set of linear matrix inequalities (LMIs) [23, 24]. The existence of LMIs in control theory starts with the Lyapunov theory and Lyapunov inequality [25]. Practical applications of LMIs initially refer to graphical methods [26] yet with the recognition that many LMIs for control problems can be solved by convex optimization and introduction of efficient methods. So such as interior point methods, LMIs have gained popularity since the late 90s up to now [25, 27, 28]. As the SDP emerges more into the control theory, it is recognized that SDP is suitable for more practical applications. These applications can be given in controller and observer synthesis problems [29, 30].

In essence, what is being done is a reformulation of the control theory problems, such as controller/observer synthesis, system identification, and signal processing problems, into LMIs such that a cost function and a constraint represent the goal and restrictions on the intended identity of the resulting entity (a controller, an observer, or identified system, depending on the problem) [31]. This reformulation makes sense if the cost and the constraints are linear because there are efficient solvers, as denoted previously. However, reformulating the control problem mentioned above into LMI-based optimization is not straightforward for many applications, as in the linear feedback control synthesis case. The problem is a bi-linear matrix inequality problem (BMI), a certain type of non-linearity. We

know from the literature that the BMI problem is an NP-hard problem [32–34]. There are multiple ways to address the non-linearity and transform it into an LMI problem, which will be discussed in the following sections, yet a summary of the idea is given here. With the results represented in [31, 35], bi-linear constraints in the optimization (a type of non-linearity) depend non-linearly on decision variables and are transformed into a new set of constraints that are affine functions of new variables. This leads to a non-linear but convex set of inequalities that is easily transformed into LMIs using Schur complement [31, 36]. State-of-the-art algorithms are being studied that directly address the problem as a bi-linear problem [37], which is a more general constraint in control synthesis applications.

1.1.1 Nested Robust Controller Design for Interconnected Linear Parameter Varying Agents

Coordination and cooperation is a high-level problem, where [13] presents the early formulations of control problems using graphs to model information flow, [14] presents optimal guidance laws in a cooperative setting using relative kinematics for an intercept problem. [15] studies the leader-follower flocking control problem wherein the agents follow time-varying reference velocities. The lower-level individual unit is assumed to be sufficiently robust or perform optimally for several high-level formation and cooperative control problems. On the other hand, myriad of control methodologies address just the individual vehicle control problem.

This work considers a nested robust control synthesis problem for a group of uncertain linear parameter varying (LPV) systems under a cooperative system framework. The aforemetioned LPV systems contain uncertainties in individual agent dynamic model as well as in the communication among the agents. Nested controllers can be realized in cooperative systems through a hierarchical structure such that a parent controller organizes the cooperative system and improves the group performance. In contrast, individual controllers deal with stabilizing or enhancing the performance of the individual agents. An example

of that can be given from the work [38], which considers a power distribution network and uses three levels of controllers managing distinct tasks in different time scales. In another work [39], a hierarchical controller shares control action among layers, where lower layers of the hierarchy deal with the positioning and tracking in continuous time while the higher layers focus on coordination, fault identification, and reconfiguration. This work assign tasks for nested controllers such that the lower layer controllers carry out continuous robust tracking. In comparison, the higher layer controllers produce the desired robust cooperative performance under the influence of uncertainties within both individual systems and communication media. Intuitively, cooperative systems' performances are enhanced by cooperative controllers hierarchically working with the controllers at the other layers of the cooperative system.

Improving the robust performance of the cooperative systems can be addressed with various approaches such as μ-synthesis, which is suggested in the literature for dynamic systems and cooperative systems [40,41] and Integral Quadratic Constraints (IQCs), which is described as a filter and a correlator in the literature [42]. This method provides a unified framework by characterizing the input and output behaviors of these multiple uncertainties [43–45]. A significant result for IQCs is provided in [44] emphasizing that the time-domain IQCs that fall into a particular group called "Hard IQCs" are required to satisfy a dissipation inequality for an LPV system with uncertainties [46]. If one considers a stable linear system to be a function of some non-stationary parameter and if this system satisfies certain strict dissipativity conditions under the effect of uncertainty, then one can conclude that the system is asymptotically stable under the effect of uncertainties, thus guaranteeing robustness [47]. For such an LPV system, this analysis can be done using parametrized Linear Matrix Inequalities (LMI) as described in [46]. The idea of "analysis" can also synthesize a controller since this system should satisfy a nominal performance metric. The drawback of this approach is that the feasibility of the LMI is decided by some

decision variable, which grows proportional to the square of the states in the system and the method becomes inefficient if the system size grows too much [48]. Multiple ways to treat this problem exist in the literature which include devising new approaches to solve LMI problems like cutting plane algorithms [48] or novel implementations of interior-point methods [49].

Motivations. Another way of treating the problem of finding a solution to the LMI for large cooperative systems is by using distributed analysis and synthesis methods for the LMI methodology. An example for distributed controller synthesis can be found in [50] which relies on the contractiveness of the linear time-invariant (LTI) cooperative system in addition to the well-posedness. In another work [51], cooperative LPV systems are considered, and a distributed synthesis methodology is proposed. Distributed robust analysis methodologies also exist in the literature that uses the internal stability of the channel from the output of the cooperative system via the graph adjacency matrix back into the cooperative system [43, 52]. The drawback of this method is that it restricts the graph topology, i.e., each agent can have only one input. In addition, a sparse formulation methodology of the cooperative LTI system using IQC framework is described in work [53]; however, this approach does not allow valid time-domain IQC description. Robust analysis and synthesis for LPV systems require valid time-domain IQC definition as discussed in [1]; thus, the approach of [53] is not applicable for LPV cooperative systems. Moreover, the work carried out in [1] for a single-agent system can be extended to LPV cooperative systems with uncertainties in communication. Further, by introducing the distributed robust analysis condition, this method can be extended to a distributed controller synthesis method for LPV cooperative systems suffering uncertainties.

1.1.2 Robust Edge Weight Synthesis for Cooperative Systems Suffering Uncertainties

CSs are investigated under two main subjects, which are denoted as analysis and synthesis. Under the analysis, CSs are provided methods to certify metrics of stability and performance or measure these metrics under certain conditions such as uncertainties. In the synthesis of CSs, however, goal is to improve stability and performance of the system. One way of improving these metrics is done by introducing a controller using methodologies such as optimal and robust control frameworks [54, 55].

Based on the analysis, performance of CSs are revealed to be related to various factors. As shown in [56], stability margin of a cooperative system is quantified for a change in the feedback gain using minimum singular value of the cooperative system. Cooperative system is shown to have an upper bound for the stability margin and this can be achieved by changing the communication topology. In another work [57], a unified analysis for consensus of CSs is provided using the Linear Matrix Inequalities (LMIs). One of the outcomes of this work is the notion of consensus region, which is the region that a CS can achieve consensus, and it is also provided that shape and size of this region is related to a coupling gain that magnifies the feedback gain of the cooperative controller. CSs are also investigated considering the post modern control frameworks such as integral quadratic constraints (IQCs) and robust control [42, 58]. Work given in [59] applies IQCs to employ synchronization analysis to cooperative systems having time-delays. This is a unifying approach of employing uncertainty analysis to verify the consensus of cooperative systems with uncertainties.

As the analysis illustrates, CSs are affected by how the communication between the agents is set and it brings the question of how this can be improved. Numerous robust controller synthesis methodologies are introduced in the literature for CS to achieve a performance criteria under uncertainties [55, 60]. Alternatively, increase in performance is also available if a certain communication topology and edge weighting is selected.

Topology and edge weights of network defines which agents are connected and how strong the connections are. As illustrated in work [61], effective resistance of the overall CS is reduced by optimal allocation of the edge weights. In another work [62], convex optimization is used to allocate edge weights to obtain better dynamic properties for the CS by meeting upper and lower bound constraints on maximum and Fiedler eigenvalues respectively. Before mentioned methods formulate the problems as convex optimization problems and uses semi-definite programming (SDP) to find optimal solution. Alternatively, one can optimize certain objectives for a CS by moving agents along optimal trajectories. This is discussed in [63], where optimal trajectory for agents is calculated to obtain minimum transmitting power by introducing topology constraints on the optimization problem. Design problem can also be formulated as a non-convex quadratically constrained quadratic problem (QCQP) and converted to a rank minimization optimization problem using semi-definite relaxation methods. Thus, original QCQP problem, which is classified as NP-hard, is converted to a linear problem and solved using SDP methods [64].

Formulation of synthesis problem is a difficult task using LMIs. Specifically, synthesis problem for output feedback requires a bilinear matrix inequality (BMI) constraint on the optimization. As illustrated in literature, this can be addressed by a decoupling of the decision variables [65] or convexification via congruence transformation [36]. However, it can also be taken as is and solved as a BMI optimization problem and means to solve this problem are local methods, which does not guarantee global optimum. These methods are augmented Lagrangian method [66], sequential semi-definite programming method [37] and non-smooth methods for BMIs [67].

Motivations. CSs are becoming vastly used systems in recent years and like other systems they are asked to meet certain performance criteria with unmodeled degradations in the communication. This problem is addressed by robust controller synthesis, however, structured controllers are introducing complexities into the CS design. One of these complexities

is that increased system dimensions leads to infeasible calculation times, especially for large systems. On the other hand, communication topology and edge weight design can increase performance of the CS with no-added complexities into CS design. However, to the best of authors' knowledge there is not any work that incorporates the uncertainties in network design either for topology or edge weights. Based on this motivation, this work illustrates a method based on sequential LMI (sLMI) approach for synthesizing connectivity weights of a CS using IQCs. CS used in this work is composed of agents that are defined as linear and strictly proper systems. These systems share their output information within the cooperative system over the prescribed communication grid defined by an adjacency matrix. The topology for this adjacency matrix is initially provided.

1.1.3 Distributed H_∞ Edge Weight Synthesis for Cooperative Systems

Conventional way to approach CS synthesis relies on lumped methods, where CS is modeled as a single system. However, this approach impose a heavy computational load in synthesis process even for relatively small CSs, which motivates the distributed methodologies. Especially in the area of controller synthesis there are various examples of distributed control methods. Some of the recent literature from different frameworks can be given as follows. In [68], an optimal control framework is used to solve multi-objective optimization problem for DC micro-grid using local-neighbour information. In another work, distributed impedance control is synthesized for event-triggered cooperative manipulation under disturbances using Lyapunov stability theory [69]. Finally, a distributed robust controller synthesis methodology from a dissipation perspective is utilized in [55] to consider uncertainties with CS.

As defined in previous section, synthesis of CSs are handled by synthesizing the CS itself by modifying topology or edge weights, which is cast into a semi-definite programming (SDP) [61, 62]. Conventionally, lumped problems are computationally heavy problems

for large scale CS, however, they can also be effectively solved using sparsity promoting methods. In a recent work, topology design of an undirected network is posed as an optimal control problem and solved using proximal gradient and Newton methods [70]. Sparsity of the controller is promoted using l_1 regularization into the H_2 formulation and derivation of a dual problem, which allows application of proximal algorithms. Similar examples can be given from distributed optimization of network systems, where cost function is separable into smaller convex cost functions [71, 72]. Recently, edge weight synthesis of CS is modeled in a distributed fashion and solved using set of linear matrix inequalities (LMI) that satisfy the neutrality between agents and nominal H_∞ performance conditions [73]. Without the relaxations, this problem is NP-hard due to nature of the problem as defined in [74], however, it is given a solution by collecting sources of non-linearity in a synthetic system, which undertakes large scale matrix permutations. The communication grid was parametrized with a single parameter and topology of the CS is predefined and not synthesized.

*Motivations.*To the best of authors knowledge, edge weight synthesis of CSs is not posed to improve H_∞ performance in distributed fashion, which will benefit potentially faster converging algorithms. Method originally defined in [75] and recently implemented for controller synthesis in [55], imposes a constraint on the interconnections by assuming the interconnections are ideal. This assumption results in a symmetric adjacency matrix, which is a limitation on the synthesis. This method is utilized in this research for distributed edge weight synthesis, where performance is measured in terms of induced $\mathcal{L}_2$ norm of the system. Without the relaxations, this problem is NP-hard [74].

1.1.4 Cooperative Model Predictive Control Strategy for Docking with Task Prioritization

Autonomous aerial systems have become essential in numerous practical fields, such as in public safety as a surveillance tool and first response [76–78]. It has also been studied

for the delivery of goods by the industry [79, 80]. A critical task during the operation of an aerial system is docking maneuver, where it might be approaching a stationary or a moving platform [81, 82] to drop/load cargo. Besides that, aerial vehicles require refueling or recharging to extend their workspace [83]. Docking is an intricate maneuver that necessitates the awareness of the docking path's constraints regarding flight safety and the docked platform's attitude [84]. In addition, nonlinear effects such as the wake of the leading agent due to the proximity flight must be considered [85] as well as the ground effect [86]. Thus it is clear that the characteristic of this maneuver requires the agent to handle certain constraints and uncertainties.

A popular control method to handle previously described constraints and calculate control actions is model predictive control (MPC). An MPC reaches the desired control action by minimizing a given objective using linear and non-linear optimization theory as applied in the case of a tracking controller [87]. MPC also allows the integration of secondary tasks into decision-making as in [88], which makes it a unified strategy to handle docking maneuvers without requiring ad-hoc integration of multiple frameworks that increase complexity. Implementing MPC-based docking strategies for space applications exists in case studies, where line of sight constraints are satisfied, energy-saving strategies are pursued, and docking on tumbling objects are executed [89–91].

Autonomous docking for aerial vehicles requires state information of the docked platform, which receives the docking agent. Therefore it certainly requires an on-board or external mechanism to sense and estimate states. On-board sensors such as cameras or lidar are widely implemented solutions (see [92]), yet they are bounded with range limitations. External mechanisms to obtain state information of the docked platform are sensors placed on the agents, except for the docking agent or external observers. As a result, this information is shared over a communication grid as illustrated in [93]. Besides state information, some of the autonomous docking literature can be grouped in terms of

cooperation at the controller level. In one of the works, a central controller calculates lateral and longitudinal velocity commands for both of the docking maneuver devices: uncrewed ground vehicle (UGV) and uncrewed aerial vehicle (UAV). Then these control signals are fed into these agents [94]. In the extension of this work, an MPC strategy is utilized on the same system, where both agents are cooperatively trying to execute docking maneuvers [95]. In another work, a multirotor docks onto a fixed-wing platform, where only the multirotor executes the docking maneuver [82]. The aerial refueling problem is addressed as a docking problem in [83]. However, only the docking platform and the boom are manipulated. Recent work studies an uncrewed sea surface vehicle (USV) landing and designs an MPC controller for a multirotor, where cooperative docking is executed [96]. However, the docking trajectory calculation is calculated on the multirotor and shared back to the USV. Another essential feature of this controller is docking algorithm initiates once the multirotor is in the vicinity of the USV, and another controller handles the approach.

Motivations When multiple agents are communicating with each other and obeying specific rules, i.e., collision avoidance, velocity matching, and staying within the vicinity of the neighbors, the aggregate of these agents are called cooperative agents [97] and the application of these systems are vast due to the advantages. Popular applications are uncrewed vehicle [98], and space applications citeYao2019. The nature of autonomous docking makes the systems performing this maneuver a cooperative system. However, it is not formally addressed as one in the vast majority of the literature since developed algorithms are mostly developed for one of the agents. On the other hand, controllers cooperatively addressing this problem are mainly centralized on a ground controller or in one of the agents, which might suffer performance or even failure due to uncertainties in the communication. Apart from the centralized implementation of cooperative docking, there are a few applications where both agents are taking part in the docking maneuver. Since the cooperative control framework allows various communication topologies, previously

described methods are obtainable with a unified cooperative control framework that is based on MPC with local neighbor state information. Besides the cooperative aspect, a prioritization of the states to track during docking is not studied in the literature. As summarized before, agents that are too far apart from each other first minimize the positional difference, which is done to be in a feasible solution set when the MPC for docking is initiated. Instead of handling this problem as two separate subproblems, an automated prioritization of tracking certain states can be defined so that linear states can be given higher priority over the angular states. Then this single problem can be solved. Therefore, this highlights the motivation of this research.

1.1.5 Graph Theoretic Online Trajectory Optimization and for ASLB Biped Robot

Planning a trajectory for a biped robot is a complex task as free-floating base of the robot is moved by the discontinuous contact forces acting on the feet. This propulsion method requires attention in planning motion of the contact points and the contact forces while considering the dynamic effects on the robot [99]. In addition to that, a straightforward optimal control formulation results in a intractable non-linear programs as stated in the literature [100, 101].

There are multiple causes of complexity in planning the trajectory of a floating base robot. First of all, the pose of the floating base is described with six degrees of freedom (DoF) unactuated base coordinates. Then actuated joint coordinates of legs are added on top of that, which results in a relatively large joint coordinates [99]. Therefore, a trajectory optimization of such a system should find optimal values for all these coordinates. Secondly, contact between feet and the terrain must comply certain contact conditions such as unilateral contact forces [102]. In literature, this problem is handled as a trajectory optimization problem using numerical optimization techniques such as direct optimal control methods [103], indirect optimal control methods [104], dynamic programming [105], and sequential

methods [106]. By this way, trajectory of the robot along with calculated joint coordinates, torques and contact forces are calculated considering the upper level constraints on the task.

Complexity of the problem is reduced in multiple ways. A way to do it is to use low fidelity dynamic models for the robot such as single rigid body dynamics model, where a lumped inertia is attached to the body frame and actuated links are assumed to be moving slowly and having low inertia [100]. Another way to reduce complexity is to split the the optimization into smaller sub-problems and pre-defining some portions of the trajectory such as footholds [107].

However, simplifications in robot model and optimization problem compromise with the complexity of the motion that can be calculated by the trajectory optimization. For this reason means to simplify full-body dynamics without compromising the fidelity of the model is vital. Recently, trajectory optimization problem is distributed into smaller alternating sub-problems, where first one satisfy dynamic constraints of the problem by finding robot momentum and contact forces, then second one finds the leg kinematics that satisfy robot dynamics. This is denoted as centroidal and whole-body model splitting and first introduced in work [106], where a sequential optimal control formulation is represented. In another work based on the same splitting of whole-body motion and centroidal dynamics, locomotion problem is cast into a mathematical framework based on Alternating Direction Method of Multipliers (ADMM) [108]. This paper exploits the natural splitting between centroidal and manipulator dynamics and ensures consensus between these models. Another recent example uses same splitting and introduces an accelerated ADMM algorithm to solve locomotion problem. [101].

Besides centroidal and whole body splitting of the dynamic model of a legged robot, there is obviously another splitting between the portions of the body. This splitting is naturally defined by the design of legged robots and named as body and limbs. Depending on the design, they can contain a single body or a set of multi-bodies and interconnection

between these sets maintained over a connection node, which is a joint. A formal way to describe the cooperation between these sets already exist in the literature within cooperative system framework [109]. This framework defines a relationship between independent agents using communication Graph, which dictates a mapping to the information flow among agents. Cooperative system framework is used in many areas such as increasing performance of a sensory network in localization of data [110], or calculating the communication topology between dynamic agents within the cooperative system [111]. It is also used in designing controllers for agents within a network [112]. All the applications have one thing in common and that is the distributed modelling of the cooperative system and distributed calculation of the variables to reach desired objective.

1.2 Objectives

The objectives of the research are depicted as follows:

1. First objective is to introduce an robust controller for linear parameter varying cooperative systems which suffers uncertainties both in single agent and cooperative system levels. The robust performance metric is depicted as H_∞. This goal is achieved by developing an algorithm for synthesizing nested linear parameter varying robust controller to be implemented on cooperative systems that ensures robust tracking at the individual agent level (lower level) and robust performance at the cooperative system level (higher level), which improves computational efficiency as the uncertainties of each layer are dealt separately. On top of that, the nested robust linear parameter varying controller is decentralized to further improve the computational efficiency.
2. The second objective of this research is to develop a synthesis method for cooperative systems with uncertainties to improve their robust H_∞ performance. This problem is known to be non-convex and requires iteration between uncertainty analysis and

H_∞ performance steps. To achieve this goal, an integral quadratic constraint analysis method is integrated into convex-concave decomposition based sequential linear matrix inequality solution method. By this way, a general algorithm for synthesizing edge weights of a cooperative system to improve robust performance is proposed in lumped fashion.

3. The third objective of the research is to develop the cooperative system synthesis method that incorporates nominal H_∞ synthesis of cooperative systems in a distributed fashion, which includes distributed modeling and optimization. To achieve this goal, the cooperative system is represented by a group of agents and connection between these agents is maintained with constraints in the synthesis. The method solves the problem using dissipative system framework based on neutrality between agents.
4. The final objective of the research is to develop implementation strategies for cooperative sytnthesis methods to improve scope of applications. To satisfy this goal, first a cooperative docking of a quadcopter on a moving rover is deviced, where a task prioritization is embedded into the controller design proces. The aforementioned controller is designed as a cooperative MPC based on local neighbor state feedback. Secondly, **ASLB**, which is a bipedal robot, is designed, and an online trajecotry optimization is introduced using cooperative synthesis methodologies. The trajectory generation strategy for this robot relies on partition of the lumped dynamics of whole robot into subsystems such as floating base, right leg and left leg and treating them as cooperative systems with individual dynamics. Then cooperative conrol framework is applied on **ASLB** to generate an online walking trajectory.

1.3 Contributions of the Proposed Research

This section provides key contributions of the research along with the regarding publications.

1.3.1 Nested Robust Controller Design for Interconnected Linear Parameter Varying Agents

This research utilize robust analysis and synthesis methods in lumped fashion for LPV systems suffering uncertainty and propose a decentralized version of the aforementioned methods to improve its computational time efficiency. The proposed method allows valid time-domain IQC description, which is necessary for IQC analysis LPV systems [44]. This paper introduces an algorithm for synthesizing nested LPV robust controller to be implemented on cooperative systems that ensures robust tracking at the individual agent level (lower level) and robust performance at the cooperative system level (higher level). The main contributions of the paper are as stated below:

1. The paper extends the application of the LPV robust synthesis to cooperative systems in a nested manner to address performance needs of single and cooperative system layers individually.
    - Effectiveness of the method proposed by [1] is projected on LPV cooperative systems.
2. The paper transforms this extended application to a decentralized nested controller synthesis by modifying and using the distributed robust analysis [43, 52] condition within the LPV robust synthesis.
    - Inefficiency of the LPV robust synthesis for cooperative systems is eliminated by the decentralized nested approach.

    Publications based on the research are listed as follows:

- Baris Taner, Kamesh Subbarao, "Nested Robust Controller Design for Interconnected Linear Parameter Varying Aerial Vehicles", Journal of Guidance, Control, and Dynamics, Vol. 44, No. 8, pp. 1454-1468, 2021, https://doi.org/10.2514/1.G005323
- Rajnish Bhusal, Baris Taner, Kamesh Subbarao, "Performance analysis of a team of highly capable individual unmanned aerial systems", AIAA Scitech 2020 Forum, pp. 2070, 2020, https://doi.org/10.2514/6.2020-2070
- Baris Taner, Rajnish Bhusal, Kamesh Subbarao, "A nested robust controller design for interconnected vehicles", AIAA Scitech 2020 Forum, pp. 0602, 2020, https://doi.org/10.2514/6.2020 0602

1.3.2 Robust Edge Weight Synthesis for Cooperative Systems Suffering Uncertainties

This research formulates robust H_∞ performance synthesis using valid time domain IQC description to synthesize edge weights of a cooperative system in a lumped fashion. The formulation of the problem leads to a bi-linear optimization problem and solution to that is given by utilizing convex-concave decomposition and sequential programming. Major contributions of this work are as follows:

1. This work proposes a sequential optimization algorithm that incorporates H_∞ performance analysis to provide nominal H_∞ performance for cooperative systems. The bi-linear matrix inequality constraints are replaced by the linear approximations obtained by convex-concave decomposition.
2. IQC analysis is integrated on the aforementioned method to incorporate various uncertainties that occur in the cooperative system. By this way, a robust H_∞ synthesis methodology is provided for edge weight synthesis. To the best of the author's knowledge, this is the only work that accommodate robust H_∞ technique in edge weight synthesis.

   Publication based on the research is listed as follows:

- Baris Taner and Kamesh Subbarao, "Robust Edge Weight Synthesis for LPV Multi-Agent Systems with Integral Quadratic Constraints", Journal of the Franklin Institute, Submitted in June 2022 (Under Review)

1.3.3 Distributed H_∞ Edge Weight Synthesis for Cooperative Systems

This work resolves limitations on classes of adjacency matrices that can be synthesized by distributed cooperative system definition based on neutrality. Besides, this work also resolves complexity due to NP-hard definition of the problem by redefining the cooperative system. This is done by embedding adjacency matrix into a synthetic system definition and making ideal interconnections between original agents of the cooperative system and the synthetic one. This results in a unique communication topology, which allows distributed cooperative system modeling for any adjacency matrix. In addition, when underlying structure is exploited, this method eliminates the conditions leading to NP-hard problem by promoting sparsity in defining interconnection constraints. On top of that, successive linearization is adopted along with the sequential programming to improve computational efficiency of the problem. Key contributions of this paper are as follows:

- Adopts distributed cooperative system modeling into an edge weight synthesis problem by resolving its limitations on adjacency matrix.
- Resolves complexity due to NP-hardness by introducing an equivalent sparse and ideal cooperative system definition to replace original cooperative system. By this way, problem becomes linear matrix inequality optimization problem, which can be solved using semi-definite programming solvers.
- Separates the non-linearities in the synthesis inequalities and addresses them by successive and convex-concave-decomposition-based-sequential linear approximations. Resulting algorithm can synthesize any class of adjacency matrix within the relative interior.

Publication based on the research is listed as follows:

- Baris Taner, Kamesh Subbarao, "Distributed H_∞ Edge Weight Synthesis for Cooperative Systems," 2021 60th IEEE Conference on Decision and Control (CDC), 2021, pp. 6652-6658, https://doi.org/10.1109/CDC45484.2021.9682966.

1.3.4 Cooperative Model Predictive Control Strategy for Docking with Task Prioritization

Key contributions of this paper are as follows:

- This paper proposes a cooperative control strategy based on MPC for docking. The designed strategy implements a non-linear and a linear MPC for coarse approach (long distance) and delicate docking maneuver (short distance) based on the same objective function with tailored optimization strategies. A leader-follower type of topology is adopted, where the quadcopter docks on the UGV. As a showcase, this controller performs short, long-distance docking of a quadcopter on a UGV.
- Formulation of the MPC includes task prioritization, which is based on a null-space projection of the tasks being ranked. The formulation is adopted from [113] by defining the docking task in terms of the docking agents' Degrees of Freedom (DoF).

Publication based on the research is listed as follows:

- Baris Taner and Kamesh Subbarao,"Model Predictive Control for Cooperative Systems with Task Prioritization applied to Vehicle Rendezvous and Docking", 2023 AIAA SciTech, 01/23-27/2023, National Harbor, MD, USA, submission status: accepted
- Baris Taner and Kamesh Subbarao, "Cooperative Model Predictive Control Strategy for Docking with Task Prioritization", IEEE TCST SI: State-of-the-art Applications of Model Predictive Control, Submission Date: 09/13/2022, Submission Number: 22-0717

1.3.5 Graph Theoretic Online Trajectory Optimization for ASLB Biped Robot

The method introduced in this paper is used to divide the lumped multi-body model of the biped into cooperative multi-bodies and address the trajectory optimization problem using this distributed model. Although the biped robot used in this work has light weight legs compared to the shoulder and the floating base, biped robot will be modeled as three cooperative agents, which are floating base, right shoulder and left shoulder, namely. All agents are defined as multi-bodies as defined in the following sections. The contributions of this paper is summarized as follows,

- This method divides the EoM of the biped robot into smaller cooperative agents, which has simpler EoMs.
- Agents with simpler dynamics that are defined at velocity level results in faster trajectory optimization.
- The quadratic programming formulation given in this paper executes an optimization problem with single objective and single augmented Hamiltonian, however, the problem can be split into multiple objectives and constraints. It should be noted that, distributed solution to this problem is not evaluated in this paper.

Chapter 2

A Nested Robust Controller Design for Interconnected Linear Parameter Variying Aerial Vehicles

2.1 Preliminaries

2.1.1 Linear Parameter Varying Systems

LPV modeling and analysis is very helpful for transforming a nonlinear system into LPV framework by embedding nonlinearities within the exogenous parameters, which enables the extension of linear control techniques to nonlinear systems [114]. The transfer matrix $\boldsymbol{G}(\boldsymbol{\rho})$ of an LPV system with input $\boldsymbol{w} \in \mathbb{R}^{n_w}$ and output $\boldsymbol{z} \in \mathbb{R}^{n_z}$ can be defined over the feasible parameter trajectory $\mathcal{A}(\boldsymbol{\rho})$ as

$$\boldsymbol{G}(\boldsymbol{\rho}) := \left[\begin{array}{c|c} \boldsymbol{A}(\boldsymbol{\rho}) & \boldsymbol{B}(\boldsymbol{\rho}) \\ \hline \boldsymbol{C}(\boldsymbol{\rho}) & \boldsymbol{D}(\boldsymbol{\rho}) \end{array}\right], \quad \mathcal{A}(\boldsymbol{\rho}) := \{\boldsymbol{\rho} \in \mathcal{P} \subset \mathbb{R}^p,\ |\dot{\boldsymbol{\rho}}(t)| < r \geq 0\ \ \forall t \geq 0\} \quad (2.1)$$

where $\boldsymbol{A}(\boldsymbol{\rho}) \in \mathbb{R}^{n_x \times n_x}$, $\boldsymbol{B}(\boldsymbol{\rho}) \in \mathbb{R}^{n_x \times n_w}$, $\boldsymbol{C}(\boldsymbol{\rho}) \in \mathbb{R}^{n_z \times n_w}$, and $\boldsymbol{D}(\boldsymbol{\rho}) \in \mathbb{R}^{n_z \times n_w}$ are the state-space matrices, and $\boldsymbol{\rho}(t) = [\rho_1, \rho_2, \ldots, \rho_p]$ is an exogeneous parameter vector with bounded derivatives.

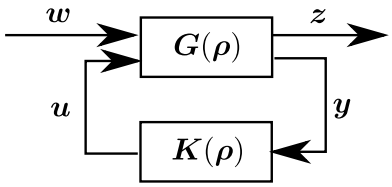


Figure 2.1. LPV system $\boldsymbol{H}(\boldsymbol{\rho})$ maps input $\boldsymbol{w}$ to output $\boldsymbol{z}$..

Let us consider the transfer matrix $\boldsymbol{H}(\boldsymbol{\rho})$ illustrated in Fig. 3.1, which is the lower-fractional transformation (LFT) between $\boldsymbol{G}(\boldsymbol{\rho})$ and the controller $\boldsymbol{K}(\boldsymbol{\rho})$ defined as $\boldsymbol{H}(\boldsymbol{\rho}) :=$

$\mathcal{F}_l(\boldsymbol{G}(\boldsymbol{\rho}), \boldsymbol{K}(\boldsymbol{\rho}))$. The performance of the closed-loop LPV system can be measured in terms of induced $\mathcal{L}_2$ gain of the input/output map with zero initial conditions, and is defined as [45],

$$\|\boldsymbol{H}(\rho)\| := \sup_{\rho \in \mathcal{A}} \sup_{\substack{\boldsymbol{w} \in \mathcal{L}_2 \\ \|\boldsymbol{w}\| \neq 0}} \frac{\|\boldsymbol{z}\|_2}{\|\boldsymbol{w}\|_2} \tag{2.2}$$

The Bounded Real Lemma for linear time-invariant (LTI) systems can be extended to obtain the upper bound of the induced $\mathcal{L}_2$ gain of the LPV system [45]. As discussed in [115, 116], an LPV system $\boldsymbol{H}(\boldsymbol{\rho})$ is exponentially stable over the parameter bounded set $\mathcal{P}$ and $\|\boldsymbol{H}(\boldsymbol{\rho})\| < \gamma$ if there exists a differentiable matrix function $\boldsymbol{X}(\boldsymbol{\rho}) = \boldsymbol{X}^T(\boldsymbol{\rho})$ such that,

$$\boldsymbol{X}(\boldsymbol{\rho}) > \boldsymbol{0}$$

$$\begin{bmatrix} \partial \boldsymbol{X}\left(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}}\right) + \boldsymbol{A}^T(\boldsymbol{\rho})\boldsymbol{X}(\boldsymbol{\rho}) + \boldsymbol{X}(\boldsymbol{\rho})\boldsymbol{A}(\boldsymbol{\rho}) & \boldsymbol{X}(\boldsymbol{\rho})\boldsymbol{B}(\boldsymbol{\rho}) \\ \boldsymbol{B}^T(\boldsymbol{\rho})\boldsymbol{X}(\boldsymbol{\rho}) & \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{C}(\boldsymbol{\rho}) & \boldsymbol{D}(\boldsymbol{\rho}) \end{bmatrix}^T \boldsymbol{P} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{C}(\boldsymbol{\rho}) & \boldsymbol{D}(\boldsymbol{\rho}) \end{bmatrix} \leq \boldsymbol{0} \tag{2.3}$$

where $\boldsymbol{P} = \begin{bmatrix} -\gamma \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & 1/\gamma \boldsymbol{I} \end{bmatrix}$ and $\gamma > 0$. In Eq. (3.3), $\partial \boldsymbol{X}\left(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}}\right)$ is defined as [45]

$$\partial \boldsymbol{X}\left(\boldsymbol{\rho}\left(t\right), \dot{\boldsymbol{\rho}}\left(t\right)\right) \;=\; \frac{d}{dt}\, \boldsymbol{X}\left(\boldsymbol{\rho}\left(t\right)\right) \;=\; \sum_{i=1}^{p} \frac{\partial \boldsymbol{X}\left(\boldsymbol{\rho}\left(t\right)\right)}{\partial \boldsymbol{\rho}_i\left(t\right)} \dot{\boldsymbol{\rho}}_i\left(t\right) \tag{2.4}$$

Equation (3.3) introduces parameter dependent Linear Matrix Inequalities. Moreover, if the system defined in Eq. 3.1 is an affine function on the set $\mathcal{P}$, then $\boldsymbol{X}(\boldsymbol{\rho})$ becomes stationary and $\partial \boldsymbol{X}(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}}) = \boldsymbol{0}$. Eventually, the resulting LMI must be satisfied on the set $\mathcal{P}$, thereby resulting in finite set of LMIs [46]. Dropping the argument $\boldsymbol{\rho}$ from the matrices for the sake of brevity, the LMIs can be written as

$$\boldsymbol{X} > \boldsymbol{0}$$

$$\begin{bmatrix} \boldsymbol{A}^T\boldsymbol{X} + \boldsymbol{X}\boldsymbol{A} & \boldsymbol{X}\boldsymbol{B} \\ \boldsymbol{B}^T\boldsymbol{X} & \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix}^T \boldsymbol{P} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix} \leq \boldsymbol{0} \tag{2.5}$$

2.1.2 Integral Quadratic Constraints

IQCs are widely used tools in analysis of uncertain systems [42, 46] and offer a framework for describing the noisy, uncertain elements of the system with a quadratic constraint on its inputs and outputs. In this context, let us denote uncertainties of a system by a bounded casual operator $\Delta : \mathcal{L}_{2e}^{n_v}[0,\infty) \rightarrow \mathcal{L}_{2e}^{n_d}[0,\infty)$. Also define signals, $\boldsymbol{v} \in \mathcal{L}_{2}^{n_v}[0,\infty)$ and $\boldsymbol{d} \in \mathcal{L}_{2}^{n_d}[0,\infty)$ satisfying $\boldsymbol{d}(t) = \Delta(\boldsymbol{v}(t))$.

Suppose that there exists a bounded rational weighting function

$$\boldsymbol{\Pi} \in \mathbb{RL}_{\infty}^{(n_v+n_d)\times(n_v+n_d)} \tag{2.6}$$

. Then, the signals $\boldsymbol{v}$ and $\boldsymbol{d}$ satisfy the frequency domain IQC defined by $\boldsymbol{\Pi}$, if

$$\int_{-\infty}^{\infty} \begin{bmatrix} \hat{\boldsymbol{v}}(i\omega) \\ \hat{\boldsymbol{d}}(i\omega) \end{bmatrix}^{*} \boldsymbol{\Pi}(i\omega) \begin{bmatrix} \hat{\boldsymbol{v}}(i\omega) \\ \hat{\boldsymbol{d}}(i\omega) \end{bmatrix} d\omega \geq 0 \tag{2.7}$$

where $\hat{\boldsymbol{v}}(i\omega)$ and $\hat{\boldsymbol{d}}(i\omega)$ are the Fourier transforms of signals $\boldsymbol{v}$ and $\boldsymbol{d}$ at frequency ω. Let us factorize $\boldsymbol{\Pi}(i\omega)$ as

$$\boldsymbol{\Pi}(i\omega) = \boldsymbol{\Psi}(i\omega)^{*}\, \boldsymbol{P}\, \boldsymbol{\Psi}(i\omega) \tag{2.8}$$

where $\boldsymbol{\Psi}(i\omega)$ is a dynamic filter satisfying $\boldsymbol{\Psi}^{*}(i\omega) = \boldsymbol{\Psi}(-i\omega)^{\mathrm{T}}$, and $\boldsymbol{P} = \boldsymbol{P}^{\mathrm{T}}$ is a correlator. The dynamic filter $\boldsymbol{\Psi}(i\omega)$ has $[\boldsymbol{d}^{\mathrm{T}} \quad \boldsymbol{v}^{\mathrm{T}}]^{\mathrm{T}}$ as inputs and $\boldsymbol{z}_{\psi}$ as an output, and can be associated to a state-space realization given by

$$\boldsymbol{G}_{\Psi} = \left[\begin{array}{c|c} \boldsymbol{A}_{\psi} & \boldsymbol{B}_{\psi} \\ \hline \boldsymbol{C}_{\psi} & \boldsymbol{D}_{\psi} \end{array}\right], \quad \boldsymbol{z}_{\psi} = \boldsymbol{G}_{\Psi} \begin{bmatrix} \boldsymbol{d} \\ \boldsymbol{v} \end{bmatrix} \tag{2.9}$$

Now, the signals $\boldsymbol{v}$ and $\boldsymbol{d}$ are said to satisfy the time domain IQC defined by $\boldsymbol{\Pi}$ if the following quadratic inequality holds for all $T \geq 0$

$$\int_{0}^{T} \boldsymbol{z}_{\psi}^{T} \boldsymbol{P} \boldsymbol{z}_{\psi} dt \geq 0 \tag{2.10}$$

2.1.3 Modeling Networked Systems

The networked system in this paper is a composition of N LTI dynamic subsystems. Let G^i denotes the i^{th} subsystem with input $\boldsymbol{u}^i \in \mathbb{R}^{n_u}$ and output $\boldsymbol{y}^i \in \mathbb{R}^{n_y}$. The interconnections among the subsystems is modelled in the graph theoretical framework [109]. To that end, each subsystem G^i is represented by a node in the graph, and the two subsystems i and j are connected to each other by an edge with an edge weight a_{ij}, where

$$\{i \in \mathbb{N},\, j \in \mathbb{N} \mid i \neq j \text{ and } 1 \leq i \leq N,\, 1 \leq j \leq N\} \tag{2.11}$$

Let $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{N \times N}$ be the adjacency (or connectivity) matrix of the underlying graph structure. Now, the interconnections among the subsystems can be represented by

$$\boldsymbol{w}^i = \sum_{j=1}^{N} a_{ij}\, \boldsymbol{z}^j, \quad \forall i = 1, 2, \ldots, N \tag{2.12}$$

where $\boldsymbol{z}^i \in \mathbb{R}^{n_z}$ is obtained as a result of duplicating the output $\boldsymbol{y}^i$ of the subsystem G^i.

In this paper, each subsystem G^i is assumed to be uncertain characterized by a multiplicative uncertainty $\Delta_i \in \mathbb{R}^{n_d \times n_v}$ such that

$$\boldsymbol{d}^i = \Delta_i\, \boldsymbol{v}^i, \quad \forall i = 1, 2, \ldots, N \tag{2.13}$$

where Δ_i can be time invariant/varying and assumed to satisfy $\|\Delta_i\| \leq 1$. Here, $\boldsymbol{d}^i \in \mathbb{R}^{n_d}$ is introduced as a signal containing the uncertainty that is extracted from the interconnections among subsystems as defined in [53]. The extraction of the uncertainty can be carried out in three steps, first of which is opening an exogenous input port for passing the signal $\boldsymbol{w}^i$ through the subsystem to have an output for the signal $\boldsymbol{v}^i \in \mathbb{R}^{n_v}$. The second one is opening another input port for signal $\boldsymbol{d}^i$ to put in the effect of uncertainty back in the subsystem. The last step is obtained as the result of duplicating the output $\boldsymbol{y}^i$ of the subsystem in the form of $\boldsymbol{z}^i$ as discussed earlier. The resulting extended subsystem is denoted as $\hat{\mathrm{G}}^i$ which

has $\boldsymbol{w}^i$, $\boldsymbol{d}^i$, and $\boldsymbol{u}^i$ as inputs and $\boldsymbol{z}^i$, $\boldsymbol{v}^i$, and $\boldsymbol{y}^i$ as the outputs. The extended subsystem $\hat{\mathrm{G}}^i$ has following state-space representation

$$\begin{bmatrix} \dot{\boldsymbol{x}}^i \\ \boldsymbol{v}^i \\ \boldsymbol{z}^i \\ \boldsymbol{y}^i \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}^i_{\hat{G}} & \boldsymbol{B}^i_d & \boldsymbol{B}^i_w & \boldsymbol{B}^i_u \\ 0 & 0 & \boldsymbol{I} & 0 \\ \boldsymbol{C}^i_z & \boldsymbol{D}^i_{zd} & \boldsymbol{D}^i_{zw} & \boldsymbol{D}^i_z \\ \boldsymbol{C}^i_x & \boldsymbol{D}^i_{yd} & \boldsymbol{D}^i_{yw} & \boldsymbol{D}^i_u \end{bmatrix} \begin{bmatrix} \boldsymbol{x}^i \\ \boldsymbol{d}^i \\ \boldsymbol{w}^i \\ \boldsymbol{u}^i \end{bmatrix} \tag{2.14}$$

where $\boldsymbol{x}^i \in \mathbb{R}^{n_x}$ is the state and $\boldsymbol{A}^i_{\hat{G}} \in \mathbb{R}^{n_x \times n_x}$, $\boldsymbol{B}^i_d \in \mathbb{R}^{n_x \times n_d}$, $\boldsymbol{B}^i_w \in \mathbb{R}^{n_x \times n_w}$, $\boldsymbol{B}^i_u \in \mathbb{R}^{n_x \times n_u}$, $\boldsymbol{C}^i_z \in \mathbb{R}^{n_z \times n_x}$, $\boldsymbol{D}^i_{zd} \in \mathbb{R}^{n_z \times n_d}$, $\boldsymbol{D}^i_{zw} \in \mathbb{R}^{n_z \times n_w}$, $\boldsymbol{D}^i_z \in \mathbb{R}^{n_z \times n_u}$, $\boldsymbol{C}^i_x \in \mathbb{R}^{n_y \times n_x}$, $\boldsymbol{D}^i_{yd} \in \mathbb{R}^{n_y \times n_d}$, $\boldsymbol{D}^i_{yw} \in \mathbb{R}^{n_y \times n_w}$, and $\boldsymbol{D}^i_u \in \mathbb{R}^{n_y \times n_u}$ are the system matrices of the subsystem $\hat{\mathrm{G}}^i$.

Now, the overall networked system can be modeled in a lumped fashion. With $\boldsymbol{w} = \left[\boldsymbol{w}^{1^{\mathrm{T}}}, \boldsymbol{w}^{2^{\mathrm{T}}}, \ldots, \boldsymbol{w}^{N^{\mathrm{T}}}\right]^{\mathrm{T}} \in \mathbb{R}^{Nn_w}$, $\boldsymbol{z} = \left[\boldsymbol{z}^{1^{\mathrm{T}}}, \boldsymbol{z}^{2^{\mathrm{T}}}, \ldots, \boldsymbol{z}^{N^{\mathrm{T}}}\right]^{\mathrm{T}} \in \mathbb{R}^{Nn_z}$, $\boldsymbol{v} = \left[\boldsymbol{v}^{1^{\mathrm{T}}}, \boldsymbol{v}^{2^{\mathrm{T}}}, \ldots, \boldsymbol{v}^{N^{\mathrm{T}}}\right]^{\mathrm{T}} \in \mathbb{R}^{Nn_v}$ and $\boldsymbol{d} = \left[\boldsymbol{d}^{1^{\mathrm{T}}}, \boldsymbol{d}^{2^{\mathrm{T}}}, \ldots, \boldsymbol{d}^{N^{\mathrm{T}}}\right]^{\mathrm{T}} \in \mathbb{R}^{Nn_d}$ we can rewrite Eqs. (2.12) and (2.13) in the global form as

$$\boldsymbol{w} = \mathbb{A}\boldsymbol{z}, \quad \boldsymbol{d} = \Delta \boldsymbol{v} \tag{2.15}$$

where $\mathbb{A} = \mathcal{A} \otimes \boldsymbol{I}_{n_z}$, $n_w = n_z$, $\Delta = \text{diag}\left\{\Delta_i\right\}|_{i=1}^N$. Here "diag" denotes the block diagonal structure.

The lumped networked system of N subsystems is denoted as $\hat{\mathrm{G}}$ such that $\hat{\mathrm{G}} = \text{diag}\left\{\hat{\boldsymbol{G}}_i\right\}|_{i=1}^N$. Figure 2.2 illustrates the lumped representation of the interconnected system, $\hat{\mathrm{G}}$.

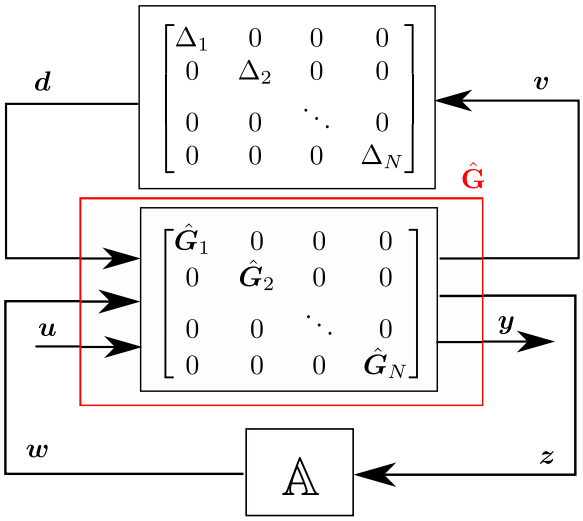


Figure 2.2. Block Diagram Representation of the transformed interconnected system $\hat{\mathrm{G}}$.

2.2 Problem Statement - Nested Robust Controller Synthesis for Interconnected LPV Systems

Consider a group of LPV modeled individual vehicles as in Eq. (3.1) (sec. 3.1.2) connected through an adjacency matrix $\mathbb{A}$ (sec. 2.1.3). The paper seeks to develop a robust control synthesis methodology, that not only guarantees robust performance as in Eq. (3.2) for every single vehicle in the network but also guarantees robust performance as in Eq. (3.2) for the group represented in Eq. (2.14) and shown in Fig. 2.2.

Further, it is noted that the modeling method represented in the previous section (adopted from [117], and [43]) gives a transfer matrix for the cooperative system that is suitable for IQC analysis to be carried out in the next section. *It should be also be noted that, the lumped modeling methodology introduces a greater computational cost in calculation of feasible LMI solutions with the increasing system dimensions such as number of vehicles in the system, number of IQCs introduced, and dimensions of the states of each vehicle* [43, 53].

The problem of increased computational cost for analysis and synthesis for this lumped system will be addressed and mitigated using a distributed analysis and synthesis methodology in the next section.

2.3 Solution Methodology

Vehicles composing a multi-agent system are individually stable systems and stability for individual vehicles and enhancement in performance of the multi-agent system are achieved by robust controllers. In this section, a methodology to do that is provided for lumped and distributed models. Following the results of [1] for designing a robust controller for a single system case, a set of nested robust controllers are synthesized for single-agent and lumped cooperative systems. The nested robust controller proposed in this paper encapsulates the single-agent level robust controller with the robust controller designed for cooperative system. This brings the benefit of distributing uncertainties to their respective levels instead of targeting them at once in a single robust controller synthesis. Moreover, it also improves the computational efficiency as targeting these uncertainties all at once increases the synthesis time dramatically. Then, by adopting a distributed IQC analysis formulation developed in [43] and *providing some modifications to eliminate restrictions on the adjacency matrix definition, a distributed analysis methodology is constructed. Eventually the distributed analysis condition is accommodated in the synthesis step to obtain a controller to increase quadratic performance of the cooperative system.*

2.3.1 Methodology for the Lumped Model

This subsection essentially recaps key developments from [1] which are included here for the sake of completeness and to motivate the subsequent developments in the sections that follow.

Having a system $\hat{\boldsymbol{G}}$ defined as in Eq. (2.16) is useful in representing both the single vehicle and cooperative systems. Here $\boldsymbol{d}$-$\boldsymbol{v}$ and $\boldsymbol{w}$-$\boldsymbol{z}$ are the uncertainty and performance channels while $\boldsymbol{u}$-$\boldsymbol{y}$ is the controller channel of the systems.

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \boldsymbol{v} \\ \boldsymbol{z} \\ \boldsymbol{y} \end{bmatrix} = \left[\begin{array}{c|ccc} \boldsymbol{A}_G(\boldsymbol{\rho}(t)) & \boldsymbol{B}_d(\boldsymbol{\rho}(t)) & \boldsymbol{B}_w(\boldsymbol{\rho}(t)) & \boldsymbol{B}_u(\boldsymbol{\rho}(t)) \\ \hline \boldsymbol{C}_v(\boldsymbol{\rho}(t)) & \boldsymbol{D}_{vd}(\boldsymbol{\rho}(t)) & \boldsymbol{D}_{vw}(\boldsymbol{\rho}(t)) & \boldsymbol{E}_v(\boldsymbol{\rho}(t)) \\ \boldsymbol{C}_z(\boldsymbol{\rho}(t)) & \boldsymbol{D}_{zd}(\boldsymbol{\rho}(t)) & \boldsymbol{D}_{zw}(\boldsymbol{\rho}(t)) & \boldsymbol{E}_z(\boldsymbol{\rho}(t)) \\ \boldsymbol{C}_x(\boldsymbol{\rho}(t)) & \boldsymbol{F}_d(\boldsymbol{\rho}(t)) & \boldsymbol{F}_u(\boldsymbol{\rho}(t)) & \boldsymbol{D}(\boldsymbol{\rho}(t)) \end{array}\right] \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{d} \\ \boldsymbol{w} \\ \boldsymbol{u} \end{bmatrix} \tag{2.16}$$

Consider a parameter-dependent linear feedback controller $\boldsymbol{K}(\boldsymbol{\rho})$ for the system $\hat{\boldsymbol{G}}$ in Eq. (2.16) with the following state-space representation

$$\begin{bmatrix} \dot{\mathbf{x}}_c \\ \mathbf{u} \end{bmatrix} = \left[\begin{array}{c|c} \mathbf{A}_c(\boldsymbol{\rho}(t)) & \mathbf{B}_c(\boldsymbol{\rho}(t)) \\ \hline \mathbf{C}_c(\boldsymbol{\rho}(t)) & \mathbf{D}_c(\boldsymbol{\rho}(t)) \end{array}\right] \begin{bmatrix} \mathbf{x}_c \\ \mathbf{y} \end{bmatrix} \tag{2.17}$$

where $\mathbf{x}_c$ is the state of the controller and $\mathbf{A}_c$, $\mathbf{B}_c$, C_c, and D_c are the parameter dependent system matrices of the controller. Denoting $\hat{\boldsymbol{H}}$ as the closed-loop system, the closed-loop state-space data of the LPV system using parameter dependent controller is given by

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \mathbf{v} \\ \mathbf{z} \end{bmatrix} = \left[\begin{array}{c|cc} \boldsymbol{A}_G^{CL}(\boldsymbol{\rho}(t)) & \boldsymbol{B}_d^{CL}(\boldsymbol{\rho}(t)) & \boldsymbol{B}_w^{CL}(\boldsymbol{\rho}(t)) \\ \hline \boldsymbol{C}_v^{CL}(\boldsymbol{\rho}(t)) & \boldsymbol{D}_{vd}^{CL}(\boldsymbol{\rho}(t)) & \boldsymbol{D}_{vw}^{CL}(\boldsymbol{\rho}(t)) \\ \boldsymbol{C}_z^{CL}(\boldsymbol{\rho}(t)) & \boldsymbol{D}_{zd}^{CL}(\boldsymbol{\rho}(t)) & \boldsymbol{D}_{zw}^{CL}(\boldsymbol{\rho}(t)) \end{array}\right] \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{d} \\ \boldsymbol{w} \end{bmatrix} \tag{2.18}$$

Denote $\Delta \in \boldsymbol{\Delta}(\Pi_1, \cdots, \Pi_M)$ as the multiplicative uncertainty for the system in Eq. (2.18) that satisfies a collection of frequency domain IQCs defined by $\{\boldsymbol{\Pi}_k\}_{k=1}^{M} \subset \mathbb{RL}_{\infty}^{(n_v+n_d)\times(n_v+n_d)}$. We construct the factorization as in Eq. 2.8 as $(\boldsymbol{\Psi}_k, \boldsymbol{P}_k)$, where $\boldsymbol{\Psi}_k$ is stable and satisfies the time-domain IQC description given in Eq. (3.18).

As discussed in [1], in order to define the robust performance of a system, we need to consider the notion of scaled uncertainty set. Following the notation in [1], let

$\Delta_{1/\gamma}(\Pi_1, \cdots, \Pi_M)$ be the scaled uncertainty set that satisfy the frequency domain IQCs defined by $\boldsymbol{S}_{1/\gamma}\Pi_k\boldsymbol{S}_{1/\gamma}$ for all $k = 1, \cdots, M$ where $\boldsymbol{S}_{1/\gamma}$ is defined in Eq. (2.20). Now, the system is said to achieve robust performance of level γ if the induced $\mathcal{L}_2$ gain of the system is upper bounded by γ over all the uncertainties in the set $\Delta_{1/\gamma}(\Pi_1, \cdots, \Pi_M)$. Next, we consider the factorization of the scaled multipliers $\boldsymbol{S}_{1/\gamma}\Pi_k\boldsymbol{S}_{1/\gamma}$ for all $k = 1, \cdots, M$ as $(\boldsymbol{\Psi}_k\boldsymbol{S}_{1/\gamma},\ \boldsymbol{P})$. The scaled filter $\boldsymbol{\Psi}_k\boldsymbol{S}_{1/\gamma}$ has $\boldsymbol{z}^k_{\psi_{1/\gamma_k}}$ as output and $[\boldsymbol{v}^{\mathrm{T}}\ \boldsymbol{w}^{\mathrm{T}}]^{\mathrm{T}}$ as input. Further, the uncertainty set is said to have a valid time-domain IQC for the factorization $(\boldsymbol{\Psi}_k\boldsymbol{S}_{1/\gamma},\ \boldsymbol{P}_k)$, if the following inequality holds true for all $k = 1, \ldots, M$

$$\int_0^T \boldsymbol{z}^{k^{\mathrm{T}}}_{\psi_{1/\gamma}}(t)\ \boldsymbol{S}_{1/\gamma}\ \boldsymbol{P}_k\ \boldsymbol{S}_{1/\gamma}\ \boldsymbol{z}^k_{\psi_{1/\gamma}}(t)\ dt \geq 0 \tag{2.19}$$

The state-space representation of the scaled filter $\boldsymbol{\Psi}_k\boldsymbol{S}_{1/\gamma}$ for all $k = 1, \ldots, M$ can be written as

$$\boldsymbol{\Psi}_k\boldsymbol{S}_{1/\gamma} \ = \ \left[\begin{array}{c|cc} \boldsymbol{A}_\psi & \boldsymbol{B}_{\psi v} & \boldsymbol{B}_{\psi d} \\ \hline \boldsymbol{C}_{\psi_k} & \boldsymbol{D}_{\psi_k v} & \boldsymbol{D}_{\psi_k d} \end{array}\right] \left[\begin{array}{c|cc} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \frac{1}{\gamma}\boldsymbol{I}_{nv} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_{nd} \end{array}\right], \quad \forall k = 1, \cdots, M \tag{2.20}$$

Combinations of the state-space (SS) realizations given in Eqs. (2.18) and (2.20) along with supplying the input $\boldsymbol{w}$ as an output, yields the system, $\tilde{\boldsymbol{H}}$, which is represented by detailed and minimal transfer matrices given by

$$\tilde{\boldsymbol{H}} := \left[\begin{array}{cc|cc} \boldsymbol{A}_G^{CL} & \boldsymbol{0} & \boldsymbol{B}_d^{CL} & \boldsymbol{B}_w^{CL} \\ \frac{1}{\gamma}\boldsymbol{B}_{\psi v}\boldsymbol{C}_v^{CL} & \boldsymbol{A}_\psi & \frac{1}{\gamma}\boldsymbol{B}_{\psi v}\boldsymbol{D}_{vd}^{CL} + \boldsymbol{B}_{\psi d} & \frac{1}{\gamma}\boldsymbol{B}_{\psi v}\boldsymbol{D}_{vw}^{CL} \\ \hline \frac{1}{\gamma}\boldsymbol{D}_{\psi_k v}\boldsymbol{C}_v^{CL} & \boldsymbol{C}_{\psi_k} & \frac{1}{\gamma}\boldsymbol{D}_{\psi_k v}\boldsymbol{D}_{vd}^{CL} + \boldsymbol{D}_{\psi_k d} & \frac{1}{\gamma}\boldsymbol{D}_{\psi_k v}\boldsymbol{D}_{vw}^{CL} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{C}_{zd}^{CL} & \boldsymbol{0} & \boldsymbol{D}_{zd}^{CL} & \boldsymbol{D}_{zw}^{CL} \end{array}\right] = \left[\begin{array}{c|cc} \widetilde{\boldsymbol{A}} & \tilde{\boldsymbol{B}}_d & \tilde{\boldsymbol{B}}_w \\ \hline \tilde{\boldsymbol{C}}_{z_\psi} & \tilde{\boldsymbol{D}}_{\psi_k d} & \tilde{\boldsymbol{D}}_{\psi_k w} \\ \tilde{\boldsymbol{C}}_w & \tilde{\boldsymbol{D}}_{wd} & \tilde{\boldsymbol{D}}_{ww} \\ \tilde{\boldsymbol{C}}_z & \tilde{\boldsymbol{D}}_{zd} & \tilde{\boldsymbol{D}}_{zw} \end{array}\right] \tag{2.21}$$

On the other hand, the filter $\left\{\boldsymbol{\Psi}_k\boldsymbol{S}_{1/\gamma}\right\}_{k=1}^M$ can be assembled into a single filter $\boldsymbol{\Psi}_{1/\gamma}$ with output $\boldsymbol{z}_{\psi_{1/\gamma}} = [\boldsymbol{z}^{1^{\mathrm{T}}}_{\psi_{1/\gamma}}, \cdots, \boldsymbol{z}^{M^{\mathrm{T}}}_{\psi_{1/\gamma}}]^{\mathrm{T}}$. Moreover, as discussed in [1], a connection is

required between a combined multiplier, Π_λ and $\boldsymbol{\Psi}_{1/\gamma}$ for robust performance analysis. Here, the multiplier Π_λ is provided as follows

$$\Pi_\lambda \;=\; \sum_{k=1}^{M} \lambda_k \boldsymbol{S}_{1/\gamma} \Pi_k \boldsymbol{S}_{1/\gamma} \tag{2.22}$$

where $\lambda_1 > 0$ and $\lambda_k \geq 0$ for $k = 2, \cdots, M$. Further, if Π_λ has a J-spectral factorization $(\boldsymbol{\Psi}, \boldsymbol{P})$, then one can also establish another factorization given by

$$J(\Pi_\lambda) := (\boldsymbol{\Psi}_\lambda,\ \boldsymbol{P}_\lambda) = (\boldsymbol{S}_\gamma\ \boldsymbol{\Psi},\ \boldsymbol{S}_{1/\gamma}\ \boldsymbol{P}\ \boldsymbol{S}_{1/\gamma}), \quad \boldsymbol{P} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \tag{2.23}$$

where $\boldsymbol{P}_\lambda = \begin{bmatrix} \frac{1}{\gamma^2}\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{I} \end{bmatrix}$. Besides, the filter $\boldsymbol{\Psi}_\lambda$ can be associated with a state-space representation given by

$$\begin{bmatrix} \dot{\boldsymbol{x}}_\psi \\ \boldsymbol{z}_{\psi_\lambda} \end{bmatrix} = \left[\begin{array}{c|cc} \boldsymbol{A}_\psi & \frac{1}{\gamma}\boldsymbol{B}_{\psi v} & \boldsymbol{B}_{\psi d} \\ \hline \boldsymbol{C}_{\psi_\lambda} & \boldsymbol{D}_{\psi_\lambda v} & \boldsymbol{D}_{\psi_\lambda d} \end{array}\right] \begin{bmatrix} \boldsymbol{x}_\psi \\ \boldsymbol{v} \\ \boldsymbol{d} \end{bmatrix} \tag{2.24}$$

Combining the state-space representation of $\boldsymbol{\Psi}_\lambda$ in Eq. (2.24) with the transfer matrix in Eq. (2.18) results in a new extended system $\tilde{\tilde{\boldsymbol{H}}}$ given by

$$\tilde{\tilde{\boldsymbol{H}}} := \left[\begin{array}{cc|cc} \boldsymbol{A}_G^{CL} & \boldsymbol{0} & \boldsymbol{B}_d^{CL} & \boldsymbol{B}_w^{CL} \\ \frac{1}{\gamma}\boldsymbol{B}_{\psi v}\boldsymbol{C}_v^{CL} & \boldsymbol{A}_\psi & \frac{1}{\gamma}\boldsymbol{B}_{\psi v}\boldsymbol{D}_{vd}^{CL} + \boldsymbol{B}_{\psi d} & \frac{1}{\gamma}\boldsymbol{B}_{\psi v}\boldsymbol{D}_{vw}^{CL} \\ \hline \boldsymbol{D}_{\Psi_\lambda v}\boldsymbol{C}_v^{CL} & \boldsymbol{C}_{\psi_\lambda} & \boldsymbol{D}_{\Psi_\lambda v}\boldsymbol{D}_{vd}^{CL} + \boldsymbol{D}_{\Psi_\lambda d} & \boldsymbol{D}_{\Psi_\lambda v}\boldsymbol{D}_{vw}^{CL} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{C}_{zd}^{CL} & \boldsymbol{0} & \boldsymbol{D}_{zd}^{CL} & \boldsymbol{D}_{zw}^{CL} \end{array}\right] = \left[\begin{array}{c|cc} \tilde{\tilde{\boldsymbol{A}}} & \tilde{\tilde{\boldsymbol{B}}}_d & \tilde{\tilde{\boldsymbol{B}}}_w \\ \hline \tilde{\tilde{\boldsymbol{C}}}_{z_\lambda} & \tilde{\tilde{\boldsymbol{D}}}_{\Psi_\lambda d} & \tilde{\tilde{\boldsymbol{D}}}_{\Psi_\lambda w} \\ \tilde{\tilde{\boldsymbol{C}}}_w & \tilde{\tilde{\boldsymbol{D}}}_{wd} & \tilde{\tilde{\boldsymbol{D}}}_{ww} \\ \tilde{\tilde{\boldsymbol{C}}}_z & \tilde{\tilde{\boldsymbol{D}}}_{zd} & \tilde{\tilde{\boldsymbol{D}}}_{zw} \end{array}\right] \tag{2.25}$$

Following the main results of the work [1], extended systems $\tilde{\boldsymbol{H}}$ and $\tilde{\tilde{\boldsymbol{H}}}$ are used in obtaining following LMI conditions

$$\begin{aligned} &\begin{bmatrix} \tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{X}} + \tilde{\boldsymbol{X}}\tilde{\boldsymbol{A}} & \tilde{\boldsymbol{X}}\tilde{\boldsymbol{B}}_d & \tilde{\boldsymbol{X}}\tilde{\boldsymbol{B}}_w \\ \tilde{\boldsymbol{B}}_d^T\tilde{\boldsymbol{X}} & \boldsymbol{0} & \boldsymbol{0} \\ \tilde{\boldsymbol{B}}_w^T\tilde{\boldsymbol{X}} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \tilde{\boldsymbol{C}}_w & \tilde{\boldsymbol{D}}_{wd} & \tilde{\boldsymbol{D}}_{ww} \\ \tilde{\boldsymbol{C}}_z & \tilde{\boldsymbol{D}}_{zd} & \tilde{\boldsymbol{D}}_{zw} \end{bmatrix}^T \boldsymbol{P}_p \begin{bmatrix} \tilde{\boldsymbol{C}}_w & \tilde{\boldsymbol{D}}_{wd} & \tilde{\boldsymbol{D}}_{ww} \\ \tilde{\boldsymbol{C}}_z & \tilde{\boldsymbol{D}}_{zd} & \tilde{\boldsymbol{D}}_{zw} \end{bmatrix} \\ &\qquad + \sum_{k=1}^{M} \lambda_k \begin{bmatrix} \tilde{\boldsymbol{C}}_{\psi_k} & \tilde{\boldsymbol{D}}_{\psi_k d} & \tilde{\boldsymbol{D}}_{\psi_k w} \end{bmatrix}^T \boldsymbol{P}_k \begin{bmatrix} \tilde{\boldsymbol{C}}_{\psi_k} & \tilde{\boldsymbol{D}}_{\psi_k d} & \tilde{\boldsymbol{D}}_{\psi_k w} \end{bmatrix} < 0 \end{aligned} \tag{2.26}$$

$$\begin{bmatrix} \tilde{\tilde{\boldsymbol{A}}}^T\tilde{\tilde{\boldsymbol{X}}}+\tilde{\tilde{\boldsymbol{X}}}\tilde{\tilde{\boldsymbol{A}}} & \tilde{\tilde{\boldsymbol{X}}}\tilde{\tilde{\boldsymbol{B}}}_d & \tilde{\tilde{\boldsymbol{X}}}\tilde{\tilde{\boldsymbol{B}}}_w \\ \tilde{\tilde{\boldsymbol{B}}}_d^T\tilde{\tilde{\boldsymbol{X}}} & \boldsymbol{0} & \boldsymbol{0} \\ \tilde{\tilde{\boldsymbol{B}}}_w^T\tilde{\tilde{\boldsymbol{X}}} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \tilde{\tilde{\boldsymbol{C}}}_{\psi_\lambda} & \tilde{\tilde{\boldsymbol{D}}}_{\psi_\lambda d} & \tilde{\tilde{\boldsymbol{D}}}_{\psi_\lambda w} \\ \tilde{\tilde{\boldsymbol{C}}}_w & \tilde{\tilde{\boldsymbol{D}}}_{wd} & \tilde{\tilde{\boldsymbol{D}}}_{ww} \\ \tilde{\tilde{\boldsymbol{C}}}_z & \tilde{\tilde{\boldsymbol{D}}}_{zd} & \tilde{\tilde{\boldsymbol{D}}}_{zw} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{P}_\lambda & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{P}_p \end{bmatrix} \begin{bmatrix} \tilde{\tilde{\boldsymbol{C}}}_{\psi_\lambda} & \tilde{\tilde{\boldsymbol{D}}}_{\psi_\lambda d} & \tilde{\tilde{\boldsymbol{D}}}_{\psi_\lambda w} \\ \tilde{\tilde{\boldsymbol{C}}}_w & \tilde{\tilde{\boldsymbol{D}}}_{wd} & \tilde{\tilde{\boldsymbol{D}}}_{ww} \\ \tilde{\tilde{\boldsymbol{C}}}_z & \tilde{\tilde{\boldsymbol{D}}}_{zd} & \tilde{\tilde{\boldsymbol{D}}}_{zw} \end{bmatrix} < 0 \quad (2.27)$$

where

$$\boldsymbol{P}_k = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{I} \end{bmatrix}, \quad \boldsymbol{P}_\lambda = \begin{bmatrix} \frac{1}{\gamma^2}\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{I} \end{bmatrix}, \quad \boldsymbol{P}_p = \begin{bmatrix} -\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \frac{1}{\gamma^2}\boldsymbol{I} \end{bmatrix}$$

LMIs in Eqs. (2.26) and (2.27) have feasible solutions, $\tilde{\boldsymbol{X}}$ and $\tilde{\tilde{\boldsymbol{X}}}$, respectively under the assumptions in [1]. The assumptions guarantee the sufficient conditions so that the LMI in Eq. (2.26) and LMI in Eq. (2.27) provides the same solution that is, the γ robust performance obtained from the system $\|\mathcal{F}_u\left(\hat{\boldsymbol{H}}(\boldsymbol{\rho}), \Delta\right)\|$ for all $\Delta \in \Delta_{1/\gamma}$ is equal to that obtained from $\|\mathcal{F}_l\left(\hat{\boldsymbol{G}}_{scl}, \boldsymbol{K}(\boldsymbol{\rho})\right)$. Therefore satisfying LMI in Eq. (2.26) necessarily implies $\|\mathcal{F}_l\left(\hat{\boldsymbol{G}}_{scl}, \boldsymbol{K}(\boldsymbol{\rho})\right)\| \leq \gamma$.

Pre-multiplying and post-multiplying LMI condition in Eq. (2.27) with $[\boldsymbol{x}\ \boldsymbol{x}_\psi\ \boldsymbol{d}\ \boldsymbol{w}]$ and $\left[\boldsymbol{x}^T\ \boldsymbol{x}_\psi^T\ \boldsymbol{d}^T\ \boldsymbol{w}^T\right]^T$ will yield a dissipation inequality as follows

$$\dot{V} + \frac{1}{\gamma^2}\boldsymbol{v}_\lambda^T\boldsymbol{v}_\lambda - \boldsymbol{d}_\lambda^T\boldsymbol{d}_\lambda - \boldsymbol{w}^T\boldsymbol{w} + \frac{1}{\gamma^2}\boldsymbol{z}^T\boldsymbol{z} \leq 0 \quad (2.28)$$

$$\dot{V} \leq \boldsymbol{w}^T\boldsymbol{w} - \frac{1}{\gamma^2}\boldsymbol{z}^T\boldsymbol{z} + \boldsymbol{d}_\lambda^T\boldsymbol{d}_\lambda - \frac{1}{\gamma^2}\boldsymbol{v}_\lambda^T\boldsymbol{v}_\lambda \quad (2.29)$$

Using the SS representation of the filter $\boldsymbol{\Psi}_\lambda$ and manipulating the signal channel $\boldsymbol{d}$ - $\boldsymbol{d}_\lambda$, a new filter is obtained and called as $\boldsymbol{\Psi}^+$. This filter gives a system called $\hat{\boldsymbol{G}}_{scl}$, when an upper LFT connection defined as $\mathcal{F}_u(\hat{\boldsymbol{G}}, \boldsymbol{\Psi}^+)$ is made (see Eq. (2.16) for $\hat{\boldsymbol{G}}$ and Eq. (2.30) for $\boldsymbol{\Psi}^+$). Denoting $\boldsymbol{z}_{\psi_\lambda} = \left[\boldsymbol{v}_\lambda^T\ ,\ \boldsymbol{d}_\lambda^T\right]^T$, the transfer matrix of this new filter is provided as follows [1]

$$\begin{bmatrix} \dot{\boldsymbol{x}}_\psi \\ \boldsymbol{v}_\lambda \\ \boldsymbol{d} \end{bmatrix} = \left[\begin{array}{c|cc} \boldsymbol{A}_\psi & \frac{1}{\gamma}\boldsymbol{B}_{\psi v} & \boldsymbol{B}_{\psi d} \\ \hline \boldsymbol{C}_{v_\lambda} & \boldsymbol{D}_{v_\lambda v} & \boldsymbol{D}_{v_\lambda d} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{array}\right] \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{D}_{d_\lambda d}^{-1}\boldsymbol{C}_{d_\lambda} & \boldsymbol{D}_{d_\lambda d}^{-1} & -\boldsymbol{D}_{d_\lambda d}^{-1}\boldsymbol{D}_{d_\lambda w} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_\psi \\ \boldsymbol{d}_\lambda \\ \boldsymbol{v} \end{bmatrix} \quad (2.30)$$

Both upper and lower LFT interconnections are graphically illustrated in Fig. 2.3. Here, $\hat{\boldsymbol{G}}_{scl}$ represents a system obtained from the interpretation of the dissipation inequality and manipulation of the transfer matrix of filter $\boldsymbol{\Psi}_\lambda$ in Eq. (2.28) and Eq. (2.24), respectively.

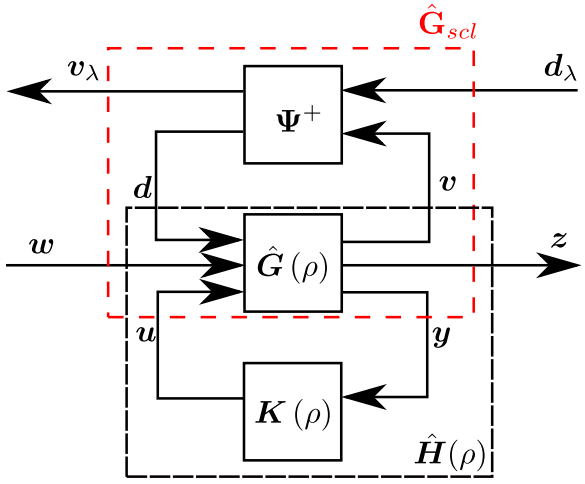


Figure 2.3. Sub-systems $\hat{\boldsymbol{H}}(\boldsymbol{\rho})$ and $\hat{\boldsymbol{G}}_{scl}$ illustrated with the interconnections to construct required extended systems..

2.3.2 Methodology for the Distributed Model

Maintaining the alternative synthesis and analysis approach for the distributed case needs a distributed robust performance definition first. A definition does exist in the literature to prove stability in a distributed fashion (see [43]) for some restricted communication graph. The existing definition verifies the internal stability of the communication channel of a cooperative system to ensure the robust stability of the overall cooperative system for arbitrarily large γ. This method, however, does not provide a γ value. In this subsection, the LMI condition to calculate robust performance of the scaled system will be given based on the approach pursued by [43]. Following that, the robust performance condition and the idea adopted from work [1] in the previous section for introducing a scaled filter to

accommodate IQC analysis is combined to obtain a distributed synthesis condition that is suitable for simultaneous synthesis and analysis.

2.3.2.1 Robustness Analysis Formulation

While lumped modelling-based approach requires high computational effort, a robustness analysis, which exploits some properties and structure of the connectivity (adjacency) matrix can be employed to reduce the computational burden. This idea is proposed for adjacency matrices that are composed of rows with only one non-zero entry in [43, 52]. In contrast, we develop a framework to carry out distributed robustness analysis for a more generic adjacency matrix.

Lemma 2.3.1 *([52], proof in Appendix A, pg 1599) Assuming networked system $\mathcal{F}_l\{\hat{\boldsymbol{H}}, \mathbb{A}\}$ is well-posed, which means $\left(\boldsymbol{I} - \boldsymbol{D}_{zw}^{CL}\mathbb{A}\right)^{-1}$ is regular. In addition, admissible trajectory for varying parameter $\boldsymbol{\rho}$ is on finite subset of $\mathcal{P}$, where parameters are frozen, then a transfer function for $\hat{\boldsymbol{H}}_{zw}(\lambda)$ at every frozen parameter is defined as $\hat{\boldsymbol{H}}_{zw}(\lambda) = \boldsymbol{D}_{zw}^{CL} + \boldsymbol{C}_z^{CL}\left(\lambda\boldsymbol{I} - \boldsymbol{A}_G^{CL}\right)^{-1}\boldsymbol{B}_w^{CL}$. Then λ is not a pole of system $\mathcal{F}_l\{\hat{\boldsymbol{H}}, \mathbb{A}\}$ if and only if $|\boldsymbol{I} - \mathbb{A}\hat{\boldsymbol{H}}_{zw}(\lambda)| \neq 0$.*

In order to carry out the robustness analysis of the cooperative system, we first provide some significant results from [52]. Let $\tilde{\mathcal{A}}$ be an adjacency matrix of the graph such that each row of $\tilde{\mathcal{A}}$ has only one non-zero element as discussed in [52]. A matrix Γ_i is constructed such that

$$\tilde{\mathbb{A}}^T\tilde{\mathbb{A}} = \text{diag}\left\{\Gamma_i^2\right\}|_{i=1}^N \tag{2.31}$$

where $\tilde{\mathbb{A}} = \tilde{\mathcal{A}} \otimes \boldsymbol{I}_{n_z}$.

Lemma 2.3.2 *([52], proof in Appendix B, pg 1599) Denote the transfer function matrix $\boldsymbol{D}_{zw}^{CL^i} + \boldsymbol{C}_z^{CL^i}\left(\lambda\boldsymbol{I} - \boldsymbol{A}_G^{CL^i}\right)^{-1}\boldsymbol{B}_w^{CL^i}$ by $\hat{\boldsymbol{H}}_{zw}^i(\lambda)$ such that $\hat{\boldsymbol{H}}_{zw}(\lambda) = \text{diag}\{\hat{\boldsymbol{H}}_{zw}^i(\lambda)|_{i=1}^N\}$.*

Assume $\|\Gamma_i \hat{\boldsymbol{H}}^i_{zw}(\lambda)\|_\infty < 1$ *for each subsystem* $i = 1, \cdots, N$. *Then the system* $\mathcal{F}_l\{\hat{\boldsymbol{H}}, \tilde{\mathbb{A}}\}$ *is stable.*

Remark 2.3.3 *As each row of* $\tilde{\mathcal{A}}$ *has only one non-zero element, we can rewrite Eq.* (2.31) *as* $\tilde{\mathbb{A}}^T\tilde{\mathbb{A}} = diag\{s_i^2\mathbf{I}_{n_z}\}\ |_{i=1}^N$ *such that* $\Gamma_i = s_i\mathbf{I}_{n_z}$. *Define a diagonal matrix* $\boldsymbol{S}$ *such that* $\boldsymbol{S} = diag\{s_1, s_2, \cdots, s_N\}$. *For the diagonal matrix* $\boldsymbol{S}$, *we can write* $\max\{s_i\}_{i=1}^N = \|S\|_\infty = \|S\|_2$. *With this,* $\|\Gamma_i \hat{\boldsymbol{H}}^i_{zw}(\lambda)\|_\infty = \|s_i \hat{\boldsymbol{H}}^i_{zw}(\lambda)\|_\infty$ *for all* $i = 1, 2, \ldots, N$. *Thus, if* $\max\{s_i\} \parallel \hat{\boldsymbol{H}}^i_{zw}(\lambda)\|_\infty < 1$ *for each subsystem* $i = 1, \cdots, N$. *Then the system* $\mathcal{F}_l\{\hat{\boldsymbol{H}}, \tilde{\mathbb{A}}\}$ *is stable.*

In the following result, the restrictions imposed on adjacency matrix in [52] are eliminated and the conditions for the stability of $\mathcal{F}_l\{\hat{\boldsymbol{H}}, \mathbb{A}\}$ provided in Lemma 2.3.2 are reformulated by defining a new scalar κ. The key purpose of redefining the following stability condition is to carry out robust distributed performance analysis for a general weighted adjacency matrix.

Lemma 2.3.4 *Denote the transfer function matrix* $\boldsymbol{D}^{CL^i}_{zw} + \boldsymbol{C}^{CL^i}_z \left(\lambda \boldsymbol{I} - \boldsymbol{A}^{CL^i}_G\right)^{-1} \boldsymbol{B}^{CL^i}_w$ *by* $\hat{\boldsymbol{H}}^i_{zw}(\lambda)$. *If* $\mathcal{A} \in \mathbb{R}^{N\times N}$ *is the weighted adjacency matrix of the graph for the cooperative system* $\mathcal{F}_l\{\hat{\boldsymbol{H}}, \mathbb{A}\}$, *then the system* $\mathcal{F}_l\{\hat{\boldsymbol{H}}, \mathbb{A}\}$ *is stable if* $\kappa \|\hat{\boldsymbol{H}}^i_{zw}(\lambda)\|_\infty < 1$ *for each subsystem* $i = 1, \cdots, N$ *where* κ *is defined as*

$$\kappa^2 := \sqrt{Nn_z}\, \|\mathbb{A}^T\mathbb{A}\|_\infty \tag{2.32}$$

and $\mathbb{A} = \mathcal{A} \otimes \boldsymbol{I}_{n_z}$.

Using equivalence of norms between $\|\mathbb{A}^T\mathbb{A}\|_2$ and $\|\mathbb{A}^T\mathbb{A}\|_\infty$ [118], one can show that κ provides an upper bound to $\max\{s_i\}_{i=1}^N$ (in Remark 2.3.3). This condition along with Lemma 2.3.2 and Remark 2.3.3 guarantees the stability condition described in Lemma 2.3.4. The complete proof has been omitted for the brevity of the paper.

As already described in [43], this decoupled stability condition ensures internal stability for a cooperative system, which equivalently means that cooperative system is robustly stable. Conversely, this is not necessarily true, besides, robustly stable cooperative systems have arbitrarily large $\kappa\|\hat{\boldsymbol{H}}_{zw}(\lambda)\|_\infty$ values [119]. In fact, system $\hat{\boldsymbol{G}}$ is composed of robustly stable individual agents because of the progression of the nested algorithm proposed in this work. Thus, $\hat{\boldsymbol{G}}$ is robustly stable. Although the need for proving stability is not required, still this channel can be used to improve cooperative performance. To calculate robust performance level while including the uncertainties, a new system is created and called as $\bar{\boldsymbol{H}}^i$. We utilize the same notation as in the original work [43] in Eq. (2.33). Referring to the work [43], an upper LFT defined as $\mathcal{F}_u\{\bar{\boldsymbol{H}}^i, \Delta_i\}$ is seen to be the same as $\kappa\hat{\boldsymbol{H}}^i_{zw}$, therefore $\|\mathcal{F}_u\{\bar{\boldsymbol{H}}^i, \Delta_i\} := \kappa\hat{\boldsymbol{H}}^i_{zw}\|_\infty < \gamma$.

$$\begin{bmatrix} \dot{\boldsymbol{x}}^i \\ \boldsymbol{v}^i \\ \boldsymbol{z}^i \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_G^{CL^i} & \boldsymbol{B}_d^{CL^i} & \boldsymbol{B}_w^{CL^i} \\ 0 & 0 & I \\ \kappa\boldsymbol{C}_z^{CL^i} & \kappa\boldsymbol{D}_{zd}^{CL^i} & \kappa\boldsymbol{D}_{zw}^{CL^i} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}^i \\ \boldsymbol{d}^i \\ \bar{\boldsymbol{w}}^i \end{bmatrix} \tag{2.33}$$

Each vehicle has a transfer function $\bar{\boldsymbol{H}}^i$ and is represented with the following minimal realization (2.34)

$$\bar{\boldsymbol{H}}^i := \left[\begin{array}{c|c} \bar{\boldsymbol{H}}^i_{vd} & \bar{\boldsymbol{H}}^i_{vw} \\ \hline \bar{\boldsymbol{H}}^i_{zd} & \bar{\boldsymbol{H}}^i_{zw} \end{array}\right] = \left[\begin{array}{c|cc} \bar{\boldsymbol{A}}_{\bar{H}}^{CL^i} & \bar{\boldsymbol{B}}_d^{CL^i} & \bar{\boldsymbol{B}}_w^{CL^i} \\ \hline \bar{\boldsymbol{C}}_v^{CL^i} & \bar{\boldsymbol{D}}_{vd}^{CL^i} & \bar{\boldsymbol{D}}_{vw}^{CL^i} \\ \bar{\boldsymbol{C}}_z^{CL^i} & \bar{\boldsymbol{D}}_{zd}^{CL^i} & \bar{\boldsymbol{D}}_{zw}^{CL^i} \end{array}\right] \tag{2.34}$$

Different than the lumped analysis case, here multiplier is defined as $\boldsymbol{\Pi}^i \in \mathbb{RL}_\infty^{(n_{vi}+n_{di})\times(n_{vi}+n_{di})}$, which has different size, therefore factorization and scaling of this multiplier yields a filter $\boldsymbol{\Psi}^i_{1/\gamma}$ with relevant size and has the following minimal realization.

$$\boldsymbol{\Psi}^i_{1/\gamma} \;:=\; \boldsymbol{\Psi}^i_k \boldsymbol{S}_{1/\gamma} \;=\; \left[\begin{array}{c|cc} \boldsymbol{A}^i_\psi & \boldsymbol{B}^i_{\psi v} & \boldsymbol{B}^i_{\psi d} \\ \hline \boldsymbol{C}^i_{\psi_k} & \boldsymbol{D}^i_{\psi_k v} & \boldsymbol{D}^i_{\psi_k d} \end{array}\right] \left[\begin{array}{c|cc} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \frac{1}{\gamma}\boldsymbol{I}_{n_{vi}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_{n_{di}} \end{array}\right] \tag{2.35}$$

The output of the filter $\boldsymbol{\Psi}^i_{1/\gamma}$ denoted as $\boldsymbol{z}^i_\psi$ is given by

$$\boldsymbol{z}^i_\psi = \boldsymbol{\Psi}^i_{1/\gamma} \left[\begin{array}{cc} \boldsymbol{I} & \boldsymbol{0} \\ \bar{\boldsymbol{H}}^i_{vd} & \bar{\boldsymbol{H}}^i_{vw} \end{array}\right] \left[\begin{array}{c} \boldsymbol{d}^i \\ \boldsymbol{w}^i \end{array}\right] \tag{2.36}$$

For the state-space characterization of the LMI, SS realizations given in Eqs. (2.34) and (2.35) and are used to build an extended system. The extended system's transfer matrix and minimal representation are introduced in Eqs. (2.37) and (2.38) respectively. It should be noted that subscript p in Eq. (2.38) stands for the signal $\boldsymbol{p}$ and collects all the performance outputs of the extended system i.e. $\boldsymbol{p} = \left[\boldsymbol{w}^T \; \boldsymbol{z}^T\right]^T$. Then for this system, an LMI characterization is given in *Lemma* 2.3.5 with Eq. (2.39).

$$\left[\begin{array}{c} \dot{\boldsymbol{x}}^i_{\bar{H}} \\ \dot{\boldsymbol{x}}^i_\psi \\ \boldsymbol{z}^i_\psi \\ \boldsymbol{w}^i \\ \boldsymbol{z}^i \end{array}\right] = \left[\begin{array}{cc|cc} \bar{\boldsymbol{A}}^{CL^i}_{\bar{H}} & \boldsymbol{0} & \bar{\boldsymbol{B}}^{CL^i}_d & \bar{\boldsymbol{B}}^{CL^i}_w \\ \frac{1}{\gamma}\boldsymbol{B}^i_{\psi v}\bar{\boldsymbol{C}}^{CL^i}_v & \boldsymbol{A}^i_\psi & \frac{1}{\gamma}\boldsymbol{B}^i_{\psi v}\bar{\boldsymbol{D}}^{CL^i}_{vd} + \boldsymbol{B}^i_{\psi d} & \frac{1}{\gamma}\boldsymbol{B}^i_{\psi v}\bar{\boldsymbol{D}}^{CL^i}_{vw} \\ \hline \frac{1}{\gamma}\boldsymbol{D}^i_{\psi_k v}\bar{\boldsymbol{C}}^{CL^i}_v & \boldsymbol{C}^i_{\psi_k} & \frac{1}{\gamma}\boldsymbol{D}^i_{\psi_k v}\bar{\boldsymbol{D}}^{CL^i}_{vd} + \boldsymbol{D}^i_{\psi_k d} & \frac{1}{\gamma}\boldsymbol{D}^i_{\psi_k v}\bar{\boldsymbol{D}}^{CL^i}_{vw} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \\ \bar{\boldsymbol{C}}^{CL^i}_z & \boldsymbol{0} & \bar{\boldsymbol{D}}^{CL^i}_{zd} & \bar{\boldsymbol{D}}^{CL^i}_{zw} \end{array}\right] \left[\begin{array}{c} \boldsymbol{x}^i_{\bar{H}} \\ \boldsymbol{x}^i_\psi \\ \boldsymbol{d}^i \\ \boldsymbol{w}^i \end{array}\right] \tag{2.37}$$

$$\mathcal{H}^i := \left[\begin{array}{c|c} \mathcal{H}^i_{\psi d} & \mathcal{H}^i_{\psi w} \\ \hline \mathcal{H}^i_{pd} & \mathcal{H}^i_{pw} \end{array}\right] = \left[\begin{array}{c|cc} \mathcal{A}^i & \mathcal{B}^i_d & \mathcal{B}^i_w \\ \hline \mathcal{C}^i_\psi & \mathcal{D}^i_{\psi d} & \mathcal{D}^i_{\psi w} \\ \mathcal{C}^i_w & \mathcal{D}^i_{wd} & \mathcal{D}^i_{ww} \\ \mathcal{C}^i_z & \mathcal{D}^i_{zd} & \mathcal{D}^i_{zw} \end{array}\right] \tag{2.38}$$

Lemma 2.3.5 *([46], proof in Proposition 3.8, pg 54) For the system represented by $\mathcal{H}^i$ in Eq.* (2.38)*, if there exists a symmetric solution of $\mathcal{X}$ of the LMI in Eq.* (2.39)*, then system is strictly dissipative (see Definition 2.3.1) with respect to the supply function $s([\boldsymbol{d}^i\ ,\ \boldsymbol{w}^i]^T, [\boldsymbol{z}^i_\psi\ ,\ \boldsymbol{z}]^T) = \gamma^2\|[\boldsymbol{d}^i\ ,\ \boldsymbol{w}^i]^T\|^2 - \|[\boldsymbol{z}^i_\psi\ ,\ \boldsymbol{z}]^T\|^2$, which equivalently means system is asymptotically stable and $\|\mathcal{H}\|_\infty < \gamma$.*

$$\begin{bmatrix} \mathcal{A}^{iT}\mathcal{X}^i + \mathcal{X}^i\mathcal{A}^i & \mathcal{X}^i\mathcal{B}^i_d & \mathcal{X}^i\mathcal{B}^i_w \\ \mathcal{B}^{iT}_d\mathcal{X}^i & \mathbf{0} & \mathbf{0} \\ \mathcal{B}^{iT}_w\mathcal{X}^i & \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathcal{C}^i_\psi & \mathcal{D}^i_{\psi d} & \mathcal{D}^i_{\psi w} \\ \mathcal{C}^i_w & \mathcal{D}^i_{wd} & \mathcal{D}^i_{ww} \\ \mathcal{C}^i_z & \mathcal{D}^i_{zd} & \mathcal{D}^i_{zw} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{P} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{P}_p \end{bmatrix} \begin{bmatrix} \mathcal{C}^i_\psi & \mathcal{D}^i_{\psi d} & \mathcal{D}^i_{\psi w} \\ \mathcal{C}^i_w & \mathcal{D}^i_{wd} & \mathcal{D}^i_{ww} \\ \mathcal{C}^i_z & \mathcal{D}^i_{zd} & \mathcal{D}^i_{zw} \end{bmatrix} < 0 \tag{2.39}$$

Strict dissipativity is introduced in the following definition.

Definition 2.3.1 *(Strict Dissipativity) [46] A system* $\mathbf{G}$*, which has $\boldsymbol{w}$ as input, $\boldsymbol{z}$ as output and $\boldsymbol{x}$ as state, with supply rate $s(\boldsymbol{w}, \boldsymbol{z})$ is said to be strictly dissipative if there exists a non-negative $V : X \to \mathbb{R}$ which attains a strong global minimum over trajectories of $(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{x})$ and an $\epsilon > 0$ s.t.*

$$V(\boldsymbol{x}(t_0)) + \int_{t_0}^{t_1} \left(s(\boldsymbol{w}(t), \boldsymbol{z}(t)) - \epsilon^2\|\boldsymbol{w}(t)\|^2\right) dt \geq V(\boldsymbol{x}(t_1)), \quad \forall t_0 \leq t_1 \tag{2.40}$$

If $V(\boldsymbol{x}(\cdot))$ is differentiable then Eq. (2.40) *can be written as*

$$\dot{V}(t) \leq s(t) \tag{2.41}$$

To show strict dissipativity for the system given in Eq. (2.38), using the previously provided information we need to pre-multiply and post-multiply the LMI condition given in Eq. (2.39) with the vector $\begin{bmatrix}\boldsymbol{x}^i_{\bar{H}} & \boldsymbol{x}^i_\psi & \boldsymbol{d}^i & \boldsymbol{w}^i\end{bmatrix}$ and $\begin{bmatrix}\boldsymbol{x}^{i\ T}_{\bar{H}} & \boldsymbol{x}^{i\ T}_\psi & \boldsymbol{d}^{iT} & \boldsymbol{w}^{iT}\end{bmatrix}^T$, respectively. This will yield Eq. (2.41), and ensures strict dissipativity according to the *Definition* 2.3.1.

We can conclude that system $\bar{\boldsymbol{H}}^i$ with performance channel $\boldsymbol{w} \rightarrow \boldsymbol{z}$ and an uncertainty channel $\boldsymbol{d} \rightarrow \boldsymbol{v}$, is asymptotically stable.

2.3.2.2 Robust Synthesis Formulation

Describing system definitions, without considering the dependency on $\boldsymbol{\rho}$, for open loop system, $\bar{\boldsymbol{G}}^i$, scaled filter, $\boldsymbol{\Psi}^{+i}$, and the controller, $\boldsymbol{K}^i$ in an individual fashion, we obtain SS definitions given in Eq. (2.42),(2.43) and (2.44). To accommodate distributed robustness condition described in Section 2.3.2.1, a magnification of κ is applied to channel $\boldsymbol{z}$ to the system $\hat{\boldsymbol{G}}^i$ and $\bar{\boldsymbol{G}}^i$ is generated. Having $\boldsymbol{\Psi}^i_{1/\gamma}$, one can find $\boldsymbol{\Psi}^{i+}$ in the same fashion described in Section 2.3.1. Then $\bar{\boldsymbol{G}}^i$ and $\boldsymbol{\Psi}^{i+}$ is combined such that a scaled individual system is constructed and subject to both synthesis and analysis steps alternatively as described previously in Section 2.3.1.This scaled system is desired to have inputs $\boldsymbol{d}^i_\lambda$, $\boldsymbol{w}^i$, $\boldsymbol{u}^i$ and outputs $\boldsymbol{v}^i_\lambda$, $\boldsymbol{z}^i$, $\boldsymbol{y}^i$. In the synthesis step, the objective is to obtain a stabilizing controller, which minimizes induced $\mathcal{L}_2$ norm of the system having input signals $\left[\boldsymbol{d}^{i^T}_\lambda \ \boldsymbol{w}^{i^T}\right]^T$ and output signals $\left[\boldsymbol{v}^{i^T}_\lambda \ \boldsymbol{z}^{i^T}\right]^T$, respectively and in the following text this will be described.

$$\begin{bmatrix} \dot{\boldsymbol{x}}^i \\ \boldsymbol{v}^i \\ \boldsymbol{z}^i \\ \boldsymbol{y}^i \end{bmatrix} = \left[\begin{array}{c|ccc} \boldsymbol{A}^i_G & \boldsymbol{B}^i_d & \boldsymbol{B}^i_w & \boldsymbol{B}^i_u \\ \hline \boldsymbol{C}^i_v & \boldsymbol{D}^i_{vd} & \boldsymbol{D}^i_{vw} & \boldsymbol{E}^i_v \\ \kappa\boldsymbol{C}^i_z & \kappa\boldsymbol{D}^i_{zd} & \kappa\boldsymbol{D}^i_{zw} & \kappa\boldsymbol{E}^i_z \\ \boldsymbol{C}^i_x & \boldsymbol{F}^i_{yd} & \boldsymbol{F}^i_{yw} & \boldsymbol{0} \end{array}\right] \begin{bmatrix} \boldsymbol{x}^i \\ \boldsymbol{d}^i \\ \boldsymbol{w}^i \\ \boldsymbol{u}^i \end{bmatrix} = \left[\begin{array}{c|ccc} \bar{\boldsymbol{A}}^i_G & \bar{\boldsymbol{B}}^i_d & \bar{\boldsymbol{B}}^i_w & \bar{\boldsymbol{B}}^i_u \\ \hline \bar{\boldsymbol{C}}^i_v & \bar{\boldsymbol{D}}^i_{vd} & \bar{\boldsymbol{D}}^i_{vw} & \bar{\boldsymbol{E}}^i_v \\ \bar{\boldsymbol{C}}^i_z & \bar{\boldsymbol{D}}^i_{zd} & \bar{\boldsymbol{D}}^i_{zw} & \bar{\boldsymbol{E}}^i_z \\ \bar{\boldsymbol{C}}^i_x & \bar{\boldsymbol{F}}^i_{yd} & \bar{\boldsymbol{F}}^i_{yw} & \boldsymbol{0} \end{array}\right] \begin{bmatrix} \boldsymbol{x}^i \\ \boldsymbol{d}^i \\ \boldsymbol{w}^i \\ \boldsymbol{u}^i \end{bmatrix} \tag{2.42}$$

$$\begin{bmatrix} \dot{\boldsymbol{x}}^i_c \\ \boldsymbol{u}^i \end{bmatrix} = \left[\begin{array}{c|c} \boldsymbol{A}^i_c & \boldsymbol{B}^i_c \\ \hline \boldsymbol{C}^i_c & \boldsymbol{D}^i_c \end{array}\right] \begin{bmatrix} \boldsymbol{x}^i_c \\ \boldsymbol{y}^i \end{bmatrix} \tag{2.43}$$

$$\begin{bmatrix} \dot{\boldsymbol{x}}^i_\psi \\ \boldsymbol{v}^i_\lambda \\ \boldsymbol{d}^i \end{bmatrix} = \left[\begin{array}{c|cc} \boldsymbol{A}^{+i} & \boldsymbol{B}^{+i}_{d_\lambda} & \boldsymbol{B}^{+i}_{v} \\ \hline \boldsymbol{C}^{+i}_{v_\lambda} & \boldsymbol{D}^{+i}_{v_\lambda d_\lambda} & \boldsymbol{D}^{+i}_{v_\lambda v} \\ \boldsymbol{C}^{+i} & \boldsymbol{D}^{+i}_{dd_\lambda} & \boldsymbol{D}^{+i}_{dv} \end{array}\right] \begin{bmatrix} \boldsymbol{x}^i_\psi \\ \boldsymbol{d}^i_\lambda \\ \boldsymbol{v}^i \end{bmatrix} \tag{2.44}$$

The upper LFT connection $\mathcal{F}_u\left(\bar{\boldsymbol{G}}^i, \boldsymbol{\Psi}^{i^+}\right)$ is nothing but recreation of $\hat{\boldsymbol{G}}_{scl}$ for individual vehicle in the cooperative system with a magnification in $\boldsymbol{z}$ channel, which is denoted by $\bar{\boldsymbol{G}}^i_{scl}$, and yields a transfer matrix as in Eq. (2.45).

$$\begin{aligned} \begin{bmatrix} \dot{\boldsymbol{x}}^i \\ \dot{\boldsymbol{x}}^i_\psi \\ \boldsymbol{v}^i_\lambda \\ \boldsymbol{z}^i \\ \boldsymbol{y}^i \end{bmatrix} &= \left[\begin{array}{cc|cc:c} \bar{\boldsymbol{A}}^i_G & \bar{\boldsymbol{B}}^i_d\boldsymbol{C}^{+i} & \bar{\boldsymbol{B}}^i_d\boldsymbol{D}^{+i}_{dd_\lambda} & \bar{\boldsymbol{B}}^i_d\boldsymbol{D}^{+i}_{dv} + \bar{\boldsymbol{B}}^i_w & \bar{\boldsymbol{B}}^i_u \\ \boldsymbol{0} & \boldsymbol{A}^{+i} & \boldsymbol{B}^{+i}_{d_\lambda} & \boldsymbol{B}^{+i}_{v} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{C}^{+i}_{v_\lambda} & \boldsymbol{D}^{+i}_{v_\lambda d_\lambda} & \boldsymbol{D}^{+i}_{v_\lambda v} & \boldsymbol{0} \\ \bar{\boldsymbol{C}}^i_z & \bar{\boldsymbol{D}}^i_{zd}\boldsymbol{C}^{+i} & \bar{\boldsymbol{D}}^i_{zd}\boldsymbol{D}^{+i}_{dd_\lambda} & \bar{\boldsymbol{D}}^i_{zd}\boldsymbol{D}^{+i}_{dv} + \bar{\boldsymbol{D}}^i_{zw} & \bar{\boldsymbol{E}}^i_z \\ \hdashline \bar{\boldsymbol{C}}^i_x & \bar{\boldsymbol{F}}^i_{yd}\boldsymbol{C}^{+i} & \bar{\boldsymbol{F}}^i_{yd}\boldsymbol{D}^{+i}_{dd_\lambda} & \bar{\boldsymbol{F}}^i_{yd}\boldsymbol{D}^{+i}_{dv} + \bar{\boldsymbol{F}}^i_{yw} & \boldsymbol{0} \end{array}\right] \begin{bmatrix} \boldsymbol{x}^i \\ \boldsymbol{x}^i_\psi \\ \boldsymbol{d}^i_\lambda \\ \boldsymbol{w}^i \\ \boldsymbol{u}^i \end{bmatrix} \\ &= \left[\begin{array}{c|c:c} \bar{\boldsymbol{A}}^i_{scl} & \bar{\boldsymbol{B}}^i_{1_{scl}} & \bar{\boldsymbol{B}}^i_{scl} \\ \hline \bar{\boldsymbol{C}}^i_{1_{scl}} & \bar{\boldsymbol{D}}^i_{1_{scl}} & \bar{\boldsymbol{E}}^i_{1_{scl}} \\ \hdashline \bar{\boldsymbol{C}}^i_{scl} & \bar{\boldsymbol{F}}^i_{1_{scl}} & \boldsymbol{0} \end{array}\right] \begin{bmatrix} \boldsymbol{x}^i \\ \boldsymbol{x}^i_\psi \\ \boldsymbol{d}^i_\lambda \\ \boldsymbol{w}^i \\ \boldsymbol{u}^i \end{bmatrix} \end{aligned} \tag{2.45}$$

Assuming there exists a stabilizing controller $\boldsymbol{K}^i$, then lower LFT $\mathcal{F}_l\left(\bar{\boldsymbol{G}}^i_{scl}, \boldsymbol{K}^i\right)$ provides a stable system $\mathcal{H}^i_{scl}$ that is affine function of controller parameters as illustrated in Eq. (2.46).

$$\mathcal{H}^i_{scl} := \left[\begin{array}{c|c} \mathcal{A}^i_{scl} & \mathcal{B}^i_{scl} \\ \hline \mathcal{C}^i_{scl} & \mathcal{D}^i_{scl} \end{array}\right] = \left[\begin{array}{cc|c} \bar{\boldsymbol{A}}^i_{scl} & \boldsymbol{0} & \bar{\boldsymbol{B}}^i_{1_{scl}} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \hline \bar{\boldsymbol{C}}^i_{1_{scl}} & \boldsymbol{0} & \bar{\boldsymbol{D}}^i_{1_{scl}} \end{array}\right] + \left[\begin{array}{cc|c} \boldsymbol{0} & \bar{\boldsymbol{B}}^i_{scl} & \\ \boldsymbol{I} & \boldsymbol{0} & \\ \hline \boldsymbol{0} & \bar{\boldsymbol{E}}^i_{1_{scl}} & \end{array}\right] \left[\begin{array}{c|c} \mathbf{A}^i_c & \mathbf{B}^i_c \\ \hline \mathbf{C}^i_c & \mathbf{D}^i_c \end{array}\right] \left[\begin{array}{cc|c} \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \bar{\boldsymbol{C}}^i_{scl} & \boldsymbol{0} & \bar{\boldsymbol{F}}^i_{1_{scl}} \end{array}\right] \tag{2.46}$$

For this system, a matrix inequality condition can be written as described in Eq. (2.47) and this condition defines the quadratic performance synthesis problem, whose nature is contained within the quadratic performance matrix $\boldsymbol{P}_{scl}$, and objective of this problem is to

find controller parameters $\boldsymbol{A}_c^i$, $\boldsymbol{B}_c^i$, $\boldsymbol{C}_c^i$, $\boldsymbol{D}_c^i$ and $\mathcal{X} > 0$. Since $\mathcal{A}_{scl}^i$ depends on controller parameters, $\mathcal{X}\mathcal{A}_{scl}^i$ is nonlinear. Note that $\boldsymbol{P}_{scl} := \text{diag}\left\{\boldsymbol{P}_\lambda\ ,\ \boldsymbol{P}_p\right\}$ with compatible sizes.

$$\mathcal{X} > 0, \quad \begin{bmatrix} \mathcal{A}_{scl}^{i^T}\mathcal{X} + \mathcal{X}\mathcal{A}_{scl}^i & \mathcal{X}\mathcal{B}_{scl}^i \\ \mathcal{B}_{scl}^{i^T}\mathcal{X} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \boldsymbol{I} \\ \mathcal{C}_{scl}^i & \mathcal{D}_{scl}^i \end{bmatrix}^T \boldsymbol{P}_{scl} \begin{bmatrix} \mathbf{0} & \boldsymbol{I} \\ \mathcal{C}_{scl}^i & \mathcal{D}_{scl}^i \end{bmatrix} < 0 \tag{2.47}$$

The problem of nonlinearity already has a solution by a nonlinear transformation described in [35, 46]. After this transformation nonlinear dependency on $\mathcal{X}\mathcal{A}_{scl}^i$ is transformed to a affine dependency on parameter v as illustrated in Eq. (2.48).

$$\mathbf{X}(v) > 0,$$

$$\begin{bmatrix} \mathbf{A}_{scl}^{i^T}(v) + \mathbf{A}_{scl}^i(v) & \mathbf{X}(v)\mathbf{B}_{1_{scl}}^i(v) \\ \mathbf{B}_{1_{scl}}^{i^T}(v)\mathbf{X}(v) & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \boldsymbol{I} \\ \mathbf{C}_{1_{scl}}^i(v) & \mathbf{D}_{1_{scl}}^i(v) \end{bmatrix}^T \boldsymbol{P}_1 \begin{bmatrix} \mathbf{0} & \boldsymbol{I} \\ \mathbf{C}_{1_{scl}}^i(v) & \mathbf{D}_{1_{scl}}^i(v) \end{bmatrix} < 0 \tag{2.48}$$

In this single objective output feedback problem, parameter elimination is executed as described in [46] and following synthesis LMIs are obtained as in Eq. (2.49). If these LMI conditions are feasible and there exist $\mathbf{X}$ and $\mathbf{Y}$ then this means system $\mathcal{H}_{scl}^i$ achieves robust performance of γ, which indirectly means that system with specified controller satisfies the

distributed robust performance condition with uncertainties defined by the scaled filter $\mathbf{\Psi}^{i+}$. Then controller $\boldsymbol{K}^i$ can be reconstructed as illustrated in [46, 50].

$$
\begin{aligned}
&\mathbf{X}(v) = \begin{bmatrix} \mathbf{X} & \mathbf{I} \\ \mathbf{I} & \mathbf{Y} \end{bmatrix} > 0 \\
&\boldsymbol{\theta}^T \left[\begin{array}{cc} \mathbf{I} & \mathbf{0} \\ \bar{\boldsymbol{A}}^i_{scl} & \bar{\boldsymbol{B}}^i_{1_{scl}} \\ \hline \mathbf{0} & \mathbf{I} \\ \bar{\boldsymbol{C}}^i_{1_{scl}} & \bar{\boldsymbol{D}}^i_{1_{scl}} \end{array}\right]^T \left[\begin{array}{cc|cc} \mathbf{0} & \mathbf{X} & \mathbf{0} & \mathbf{0} \\ \mathbf{X} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & -\gamma\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & (\gamma\mathbf{I})^{-1} \end{array}\right] \left[\begin{array}{cc} \mathbf{I} & \mathbf{0} \\ \bar{\boldsymbol{A}}^i_{scl} & \bar{\boldsymbol{B}}^i_{1_{scl}} \\ \hline \mathbf{0} & \mathbf{I} \\ \bar{\boldsymbol{C}}^i_{1_{scl}} & \bar{\boldsymbol{D}}^i_{1_{scl}} \end{array}\right] \boldsymbol{\theta} < 0 \\
&\boldsymbol{\phi}^T \left[\begin{array}{cc} -\bar{\boldsymbol{A}}^{i^T}_{scl} & -\bar{\boldsymbol{C}}^{i^T}_{1_{scl}} \\ \mathbf{I} & \mathbf{0} \\ \hline -\bar{\boldsymbol{B}}^{i^T}_{1_{scl}} & -\bar{\boldsymbol{D}}^{i^T}_{1_{scl}} \\ \mathbf{0} & \mathbf{I} \end{array}\right]^T \left[\begin{array}{cc|cc} \mathbf{0} & \mathbf{Y} & \mathbf{0} & \mathbf{0} \\ \mathbf{Y} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & -(\gamma\mathbf{I})^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \gamma\mathbf{I} \end{array}\right] \left[\begin{array}{cc} -\bar{\boldsymbol{A}}^{i^T}_{scl} & -\bar{\boldsymbol{C}}^{i^T}_{1_{scl}} \\ \mathbf{I} & \mathbf{0} \\ \hline -\bar{\boldsymbol{B}}^{i^T}_{1_{scl}} & -\bar{\boldsymbol{D}}^{i^T}_{1_{scl}} \\ \mathbf{0} & \mathbf{I} \end{array}\right] \boldsymbol{\phi} < 0
\end{aligned}
\tag{2.49}
$$

where $\boldsymbol{\theta} = ker([\bar{\boldsymbol{C}}^i_{scl}\ \bar{\boldsymbol{F}}^i_{1_{scl}}])$ and $\boldsymbol{\phi} = ker([\bar{\boldsymbol{B}}^{T^i}_{scl}\ \bar{\boldsymbol{E}}^{T^i}_{1_{scl}}])$.

Thus, the distributed robust performance analysis and extension of this analysis to synthesis problem allows us to accommodate same algorithm for lumped and distributed cases with ease and the algorithm used to synthesize controllers for both cases are given in the next section 2.3.3.

2.3.3 Algorithm For Nested Robust Controller Synthesis

In this section, a single algorithm will be provided for synthesizing robust controllers for both single-agent and cooperative system. The algorithm is provided in Fig. 2.4 for methodology for lumped system, however, there is no difference between procedures of lumped and distributed approaches except for the construction of the extended systems $\hat{\boldsymbol{G}}_{scl}$ and $\bar{\boldsymbol{G}}^i_{scl}$. Specifically, $\hat{\boldsymbol{G}}_{scl}$ is constructed from the adjacency matrix $\mathbb{A}$ and therefore

Table 2.1. Algorithm for lumped robust synthesis methodology given in Section 2.3.1. *First seven steps are common for single-agent and lumped cooperative systems. More details on robust controller synthesis of a single-agent system can be found in [1].*

	Procedure	Locator
1	Inputs: $\boldsymbol{G}_s$, W_p, W_n, W_u, W_{ref}, $\mathbb{A}$, $\boldsymbol{\Pi}_k (k = 1, \cdots, M)$ and N	
2	Initialization: $j = 1$, $\lambda_1 = 1,\ \lambda_{k>1} = 0,\ \boldsymbol{S}_{\frac{1}{\gamma}} = \text{diag}\{\boldsymbol{I}_{nv}, \boldsymbol{I}_{nd}\}$	
3	Calculate $\boldsymbol{\pi}_\lambda$	Eq. (2.22)
4	Obtain $\boldsymbol{\Psi}_\lambda$ using J-spectral factorization $J(\boldsymbol{\Pi}_\lambda)$	Eqs. (2.23) and (2.24)
5	Rearrange $\boldsymbol{\Psi}_\lambda$ to obtain $\boldsymbol{\Psi}^+$	Eq. (2.30)
6	Obtain $\boldsymbol{G}$ (see Fig. 2.5) and construct $\boldsymbol{G}_{scl} : \mathcal{F}_u(\boldsymbol{G}, \boldsymbol{\Psi}^+)$	Fig. 2.3
7	Synthesize $\boldsymbol{K}(\boldsymbol{\rho})$ using $\min_{\boldsymbol{K}(\boldsymbol{\rho})} \| \boldsymbol{H}_{scl}(\boldsymbol{\rho}) := \mathcal{F}_l(\boldsymbol{G}_{scl}, \boldsymbol{K}(\boldsymbol{\rho})) \|$	Fig. 2.3
8	Find $\boldsymbol{\lambda}_k$ and the best upper bound of $\boldsymbol{H}(\boldsymbol{\rho}) := \mathcal{F}_u(\boldsymbol{G}(\boldsymbol{\rho}), \boldsymbol{K}(\boldsymbol{\rho}))$, which is $\| \mathcal{F}_u(\boldsymbol{H}(\boldsymbol{\rho}), \Delta) \| \leq \gamma$. Find $\gamma_{\text{error}} = \gamma_j - \gamma_{j-1}$.	Eq. (2.26)
9	If $\gamma_{\text{error}} < \gamma_{\text{tolerance}}$, $j = j + 1$, $\lambda_1 > 0,\ \lambda_{m>1} \geq 0$, $\boldsymbol{S}_{\frac{1}{\gamma}} = \text{diag}\{\frac{1}{\gamma}\boldsymbol{I}, \boldsymbol{I}\}$ and go to step 3, else move to step 10.	
10	Construct cooperative system $\hat{\boldsymbol{G}}$ (see 2.4.1.2)	Fig. 2.9
11	Follow steps 1 - 9 for cooperative system. $\boldsymbol{G} \rightarrow \hat{\boldsymbol{G}}$, $\boldsymbol{G}_{scl} \rightarrow \hat{\boldsymbol{G}}_{scl}$,	
12	If $\gamma_{\text{error}} < \gamma_{\text{tolerance}}$ go to step 10, else terminate and print the controller for cooperative system, $\boldsymbol{K}_{co}$	

contains topological effects, while $\bar{\boldsymbol{G}}^i_{scl}$ is constructed to capture the effect of κ and scale up the output $\boldsymbol{z}$ of communication channel. After the synthesis of the controller for single vehicle as illustrated in single-agent portion of Fig. 2.4, stable individual agents, $\boldsymbol{H}^i_s$ are used to construct the cooperative system $\hat{\boldsymbol{G}}$, and the procedure for the cooperative system is followed. The detailed procedure of Fig.2.4 for single-agent system and cooperative system under lumped modeling is provided in Table 2.1. Similarly, Table 2.2 provides the procedure for robust synthesis under distributed modeling. Controllers are synthesized using the method represented in [46] for single-agent, lumped and distributed cooperative cases. Formulation is illustrated only for distributed case in section 2.3.2.2 as in Eq. (2.49). The construction of the state-space definitions of single-agent system for synthesis and analysis phases are illustrated in section 2.4.1.2. In section 2.4.1.2, the construction of the state-space definitions for cooperative system is also provided in detail.

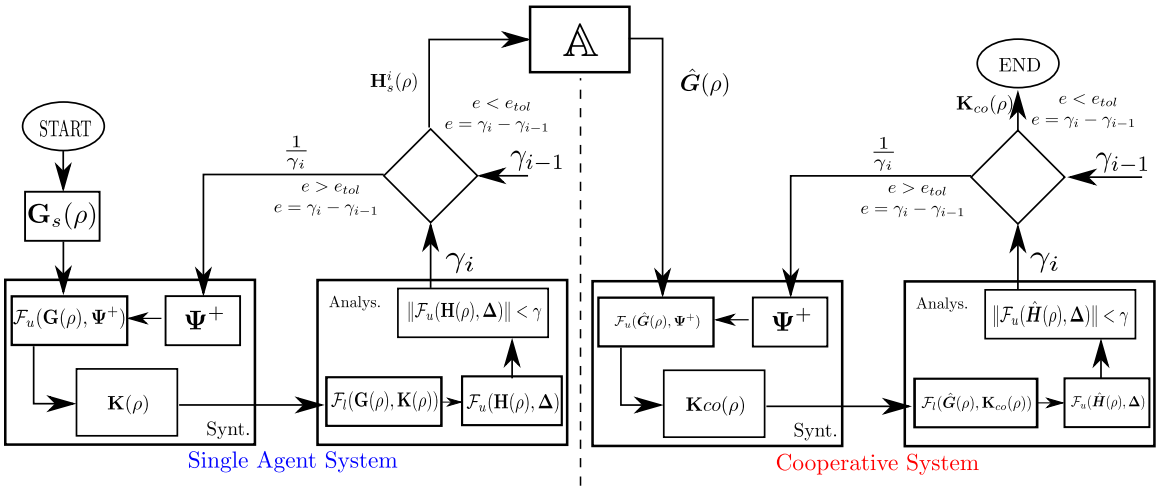


Figure 2.4. Nested robust control synthesis algorithm.

Table 2.2. Algorithm for distributed robust synthesis methodology given in Section 2.3.2.

(One should follow the first 9 steps of Table 2.1 before pursuing this one.)

	Procedure	**Locator**
13	Construct $\bar{\boldsymbol{G}}^i \;\; (i = 1, \cdots, N)$, using scalar κ	Eq. (2.42)
14	Obtain $\boldsymbol{\Psi}_\lambda$ using J-spectral factorization $J\left(\boldsymbol{\Pi}_\lambda\right)$	Eq. (2.24)
15	Rearrange $\boldsymbol{\Pi}_\lambda$ to obtain $\boldsymbol{\Psi}^+$	Eq. (2.44)
16	Obtain $\bar{\boldsymbol{G}}^i_{scl} : \mathcal{F}_u(\bar{\boldsymbol{G}}^i, \boldsymbol{\Psi}^{+i})$	Fig. 2.3
17	Synthesize $\boldsymbol{K}^i(\boldsymbol{\rho})$ using $\min_{\boldsymbol{K}^i(\boldsymbol{\rho})} \left\| \mathcal{H}^i_{scl}(\boldsymbol{\rho}) := \mathcal{F}_l\left(\bar{\boldsymbol{G}}^i_{scl}, \boldsymbol{K}^i(\boldsymbol{\rho})\right) \right\|$	Eq. (2.49)
18	Find the best upper bound of $\mathcal{H}^i(\boldsymbol{\rho}) := \mathcal{F}_u\left(\bar{\boldsymbol{G}}^i(\boldsymbol{\rho}), \boldsymbol{K}^i(\boldsymbol{\rho})\right)$, which is $\left\|\mathcal{F}_u\left(\mathcal{H}^i(\boldsymbol{\rho}), \Delta\right)\right\| \leq \gamma$ Find $\gamma_{\text{error}} = \gamma_j - \gamma_{j-1}$.	Eq. (2.39)
19	If $\gamma_{\text{error}} < \gamma_{\text{tolerance}}$ go to step 2, else terminate and print the controller for cooperative system, $\boldsymbol{K}^i$	

2.4 Simulation and Results

Verification of the nested robust controller algorithm provided in this paper is executed on Intel(R) Xeon(R) E-2124G CPU @ 3.40 GHz Computer with Matlab R18a software. LMI problems to reach solutions are solved using Matlab LMI Toolbox. Individual vehicles' dynamics are modeled by LPV model of short period dynamics of F16 Vista aircraft and the cooperative system is composed of multiple of these agents. The LPV model [120], uses Mach number and altitude as exogenous scheduling parameters.

The standard short period equations of motion of the vehicle can be modeled as,

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} Z_\alpha & 1 \\ M_\alpha & M_q \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} Z_{\delta_e} \\ M_{\delta_e} \end{bmatrix} \delta_e \tag{2.50}$$

where, α is the angle of attack, q is the pitch rate and δ_e is the elevator deflection. Further, Z_α, M_α, and M_q are the dimensional stability derivatives, and Z_{δ_e} and M_{δ_e} are the dimensional control derivatives for the aircraft's longitudinal motion. The elevator actuator is modeled as a first-order lag filter,

$$\delta_e = G_{act}\delta_{ec} \tag{2.51}$$

where, δ_{ec} is the commanded elevator deflection and $G_{act} = \dfrac{-20.2}{s + 20.2}$ is the actuator transfer function.

At trimmed level flight, the dimensional stability derivatives are the functions of Mach number (M) and the altitude (h) of the aircraft. The LPV model of F-16 VISTA aircraft short period dynamics is taken from [120] which expresses these derivatives in terms of Mach number and altitude as,

$$
\begin{aligned}
Z_\alpha(h,M) &= 0.22 - 4.1\times10^{-7}h - 2.6M + 5.15\times10^{-5}Mh \\
M_\alpha(h,M) &= 17.1 - 8.07\times10^{-4}h - 68.4M + 3.31\times10^{-3}Mh + 56.2M^2 - 10^{-3}M^2h \\
M_q(h,M) &= -0.228 + 7.06\times10^{-6}h - 2.12M + 4.86\times10^{-5}Mh \\
Z_{\delta_e}(h,M) &= -1.38\times10^{-3} + 8.75\times10^{-8}h - 0.34M + 7.98\times10^{-6}Mh \\
M_{\delta_e}(h,M) &= -8.16 + 1.73\times10^{-4}h + 40.6M - 8.96\times10^{-4}Mh - 99.3M^2 + 2.42\times10^{-3}M^2h
\end{aligned}
\tag{2.52}
$$

The LPV model of Eq. (2.50) with stability and control derivatives in (3.57) is valid throughout the flight envelope $h \in [5000\text{ ft}, 25000\text{ ft}]$ and $M \in [0.4, 0.8]$ [120]. For the purpose of simulation, they are selected to be $h = 15000$ ft and $M = [0.4,\ 0.6,\ 0.8]$.

The Weighting functions W_p, W_n, W_u, W_{ref} for loop shaping and actuator dynamics, G_{Act} are provided in Eq. (2.53) for the single vehicle case. The angle of attack tracking problem with W_{ref} as the target model can be considered as a model-matching problem. The selection of W_{ref} can be made considering the flying qualities specifications into the control design. In this problem, we consider the target model to be a second-order system with natural frequency of 3 rad/s, and the damping ratio of 0.6. The performance weight given as W_p serves to normalize the error in the model-following between the target model and the LPV model. The weighting function W_p is selected to limit the tracking error less than 10%. W_n is the scaling multiplier for noise input to the feedback, and W_u penalizes the input in the synthesis of the robust controller.The schematic of the open loop model with weighting functions and a disturbance input is illustrated in Fig. 2.5 in the Appendix.

$$
\begin{aligned}
W_p &= \frac{0.5s + 50}{s + 5}, \quad W_n = 0.01 \\
W_{ref} &= \frac{9}{s^2 + 0.36s + 9}, \quad W_u = 0.02
\end{aligned}
\tag{2.53}
$$

IQCs for single and cooperative systems are selected to compensate saturation uncertainty. For cooperative system case, an IQC for slowly time varying signals and an IQC for

constant time delay is defined as provided in work [42] and represented by $\mathbf{\Pi}_2$ and $\mathbf{\Pi}_3$ in the Table 2.3. These IQCs compensate for the time varying variation of the graph topology and time delay in edges. $\mathbf{H}(i\omega)$, $\mathbf{H}_s(i\omega)$ in IQC definitions are $\mathbf{H}(i\omega) = (j\omega + 1)^{-1}$, $\mathbf{H}_s(i\omega) = \mathbf{H}(i\omega)$, $\phi(H, d) = 1$ and $\nu = 1$. It should be noted that IQCs enlarge to comply with the size of the system. If a multiplier is partitioned as in Eq. (3.19), then an enlarged multiplier is as given in Eq. (3.63), where N and n_z are the number of vehicles and dimension of the output vector $\boldsymbol{z}$ of each vehicle.

$$\mathbf{\Pi} := \left[\begin{array}{c|c} \mathbf{\Pi}_{11} & \mathbf{\Pi}_{12} \\ \hline \mathbf{\Pi}_{21} & \mathbf{\Pi}_{22} \end{array}\right] \tag{2.54}$$

$$\mathbf{\Pi} := \left[\begin{array}{c|c} \text{diag}\{\mathbf{\Pi}_{11}\}_1^{Nn_z} & \text{diag}\{\mathbf{\Pi}_{12}\}_1^{Nn_z} \\ \hline \text{diag}\{\mathbf{\Pi}_{21}\}_1^{Nn_z} & \text{diag}\{\mathbf{\Pi}_{22}\}_1^{Nn_z} \end{array}\right] \tag{2.55}$$

Ψ_0 is taken from [42] for $\theta \in [0, 1]$ s. Dead zone for the actuator signal is $\Delta_d \in [-0.01, 0.01]$ and constant time delay is defined as $\Delta_{td} = \text{diag}\{\Delta_i\}$, where

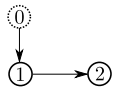
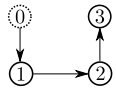
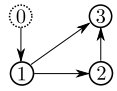
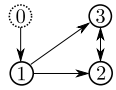
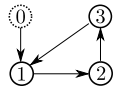
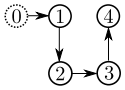
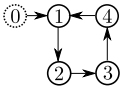
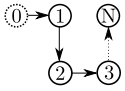
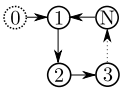
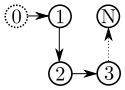
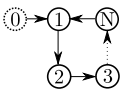
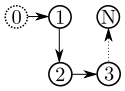
$$\Delta_i = \begin{cases} 0.25\ i_{(mod\ 3)}, & \text{if } i_{(mod\ 3)} \neq 0 \\ 0.75, & \text{if } i_{(mod\ 3)} = 0 \end{cases} \tag{2.56}$$

Table 2.3. IQCs used for the simulation

	Single-agent	**Multi-agent**
$\mathbf{\Pi}_1$	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
$\mathbf{\Pi}_2$	$\begin{bmatrix} 0 & 1 \\ 1 & -2 \end{bmatrix}$	$\begin{bmatrix} (1+\nu)\{\mathbf{H}(-j\omega)\mathbf{H}(j\omega) + \frac{\phi(H,d)^2}{\nu}\mathbf{I} & 0 \\ 0 & -\mathbf{H}(-j\omega)\mathbf{H}(j\omega) \end{bmatrix}$
$\mathbf{\Pi}_3$	$\begin{bmatrix} 0 & 1+\mathbf{H}(j\omega) \\ 1+\mathbf{H}(-j\omega) & -2(1+Re(\mathbf{H}(j\omega))) \end{bmatrix}$	$\begin{bmatrix} \lvert\Psi_{0(\theta,\omega)}\rvert & 0 \\ 0 & -1 \end{bmatrix}$

Tests are executed on various Graph topologies with different cooperative system sizes, N, given in Table 2.4. These tests reveals the properties of each methodology and make comparison between lumped and distributed methodologies. Comparison metrics are defined in terms of elapsed CPU time to find controller for given decision variable size, n_x i.e. $\boldsymbol{X} \in \mathbb{R}^{n_x \times n_x}$ and robust performance of the resulting controllers. Robust performance of these controllers are measured in terms of H_∞ norm of the system with given controllers, which is given by the induced $\mathcal{L}_2$ norm or equivalently γ. To obtain comparable time measurements, elapsed CPU time is defined as the average of elapsed CPU times for each iteration of the algorithm, where iterations continue until the error between γ_j and γ_{j-1} becomes smaller than than tolerance. Before introducing the results, it should be noted that $\boldsymbol{K}_{nom}$ is the controller that is synthesized without considering the uncertainties, while $\boldsymbol{K}_{rob}$ is the controller that is synthesized using the IQCs. After synthesis, both controllers are plugged into the same system with uncertainty, where robust performance is measured. The robust performance comparison between $\boldsymbol{K}_{nom}$ and $\boldsymbol{K}_{rob}$ is made by this way. For single vehicle case, induced $\mathcal{L}_2$ gain, γ of the system with nominal controller under the uncertainty in the actuators are calculated to be 2.29. With the robust controller, γ of the system is calculated as 1.95. The stable single vehicle system with this synthesized controller is used in both lumped and distributed methodologies and the comparison between the two methodologies is made using the directed Graph for 3 vehicles (*representative case*) as illustrated in 4^{th} row of Table 2.4.

Table 2.4. Results summary for the verification on various Graph topologies and cooperative system sizes. (DP: Directed Path, DG: Directed Graph, DC: Directed Cycle)

	Graph Type	System Size, N	Graph Topology	Lumped Methodology				Distributed Methodology			
				time,s	γ_{nom}	γ_{rob}	Size of X	time,s	γ_{nom}	γ_{rob}	Size of X
1	DP	2		542.94	5.23	4.56	$\mathbb{R}^{68\times68}$	203.79	5.61	4.91	$\mathbb{R}^{24\times24}$
2	DP	3						287.16	5.41	5.03	$\mathbb{R}^{24\times24}$
3	DG	3						362.99	5.64	5.23	$\mathbb{R}^{24\times24}$
4	DG	3		3360	5.76	5.17	$\mathbb{R}^{102\times102}$	306.55	5.59	5.22	$\mathbb{R}^{24\times24}$
5	DC	3						331.16	5.60	5.35	$\mathbb{R}^{24\times24}$
6	DP	4						450.30	5.62	4.91	$\mathbb{R}^{24\times24}$
7	DC	4						454.87	5.62	5.14	$\mathbb{R}^{24\times24}$
8	DP	5						558.54	5.62	5.15	$\mathbb{R}^{24\times24}$
9	DC	5						555.07	5.62	5.15	$\mathbb{R}^{24\times24}$
10	DP	6						1412.59	5.62	5.14	$\mathbb{R}^{24\times24}$
11	DC	6						1424.58	5.62	5.14	$\mathbb{R}^{24\times24}$
12	DP	10						922.3607	5.62	5.14	$\mathbb{R}^{24\times24}$

2.4.1 Implementation of Robust Controller Synthesis on Matlab

2.4.1.1 Construction of Single-agent System for Robust Synthesis

This section will walk through the construction of single-agent system for the readers. Here $\boldsymbol{W}_p$, $\boldsymbol{W}_{ref}$, $\boldsymbol{W}_u$ and $\boldsymbol{W}_n$ are the weighting functions. One can consider these weighting functions as systems having input and output channels. For the sake of notational simplicity, the output of these systems will be denoted with the name of the system itself e.g., the output of $\boldsymbol{W}_p$ is called as $\boldsymbol{W}_p$. For synthesizing a controller $\boldsymbol{K}$ for the single agent, an open loop system has to be constructed. Let us denote the extended single-agent system by $\boldsymbol{G}$ and the schematic representation of the system is provided in Fig. 2.5, which gives the transfer function matrix as in Eq. (2.57). $\boldsymbol{G}_s$ in Fig. 2.5 contains the F16 Vista longitudinal model given in (2.50).

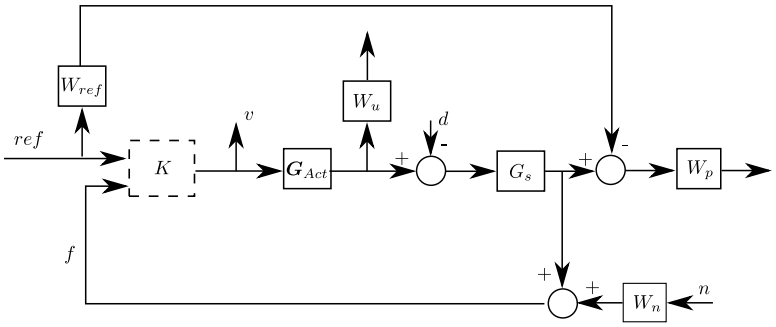


Figure 2.5. Synthesis-ready extended single-agent system, $\boldsymbol{G}$.

$$\boldsymbol{G} := \left[\begin{array}{c|c|c|c} \boldsymbol{G}_{v,d} & \boldsymbol{G}_{v,ref} & \boldsymbol{G}_{v,n} & \boldsymbol{G}_{v,u} \\ \hline \boldsymbol{G}_{wp,d} & \boldsymbol{G}_{wp,ref} & \boldsymbol{G}_{wp,n} & \boldsymbol{G}_{wp,u} \\ \hline \boldsymbol{G}_{wu,d} & \boldsymbol{G}_{wu,ref} & \boldsymbol{G}_{wu,n} & \boldsymbol{G}_{wu,u} \\ \hline \boldsymbol{G}_{ref,d} & \boldsymbol{G}_{ref,ref} & \boldsymbol{G}_{ref,n} & \boldsymbol{G}_{ref,u} \\ \hline \boldsymbol{G}_{f,d} & \boldsymbol{G}_{f,ref} & \boldsymbol{G}_{f,n} & \boldsymbol{G}_{f,u} \end{array}\right] \tag{2.57}$$

Following the notation given in Algorithm 2.4, in every iteration a system called as $\boldsymbol{G}_{scl} := \mathcal{F}_u(\boldsymbol{G}, \boldsymbol{\Psi}^+)$ has to be constructed. This upper LFT connection can be realized in MATLAB using *'lft'* for $\boldsymbol{G}$ from Eq. (2.57) along with the calculated $\boldsymbol{\Psi}^+$. Moreover, as described in the Algorithm 2.4, a system $\boldsymbol{H} := \mathcal{F}_l(\boldsymbol{G}, \boldsymbol{K}_{rob})$ should be constructed to find the robust performance of $\boldsymbol{H}$.

After completing the robust synthesis for single agent, closed-loop system in Fig. 2.6 is used in the construction of the cooperative system.

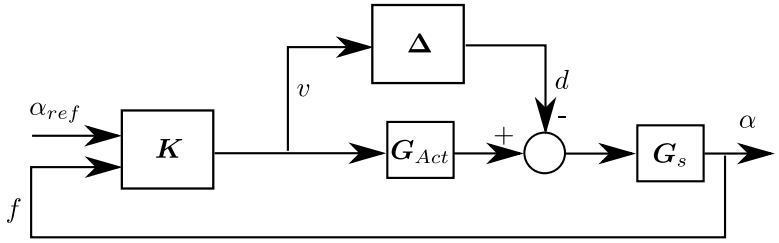


Figure 2.6. Robustly stable closed-loop system as a single agent. This system is constructed from longitudinal dynamics of F16 Vista Aircraft..

2.4.1.2 Construction of Cooperative System for Robust Synthesis

Closed-loop single-agent system, $\boldsymbol{H}_i = \boldsymbol{H}$ for all $i = 1, \cdots, N$ is used to construct cooperative system as follows. First robustly stable single agents are grouped together to obtain the system, $\boldsymbol{G}_{group}$ as illustrated in Fig. 2.7.

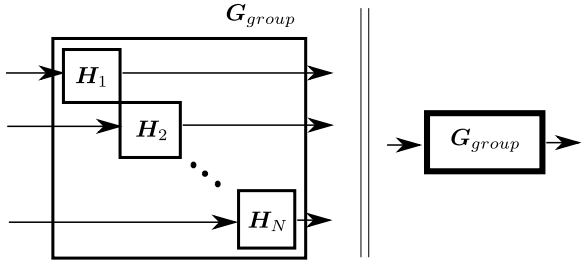


Figure 2.7. Accumulated group of agents without any interconnection..

Subsequently, a synthetic system, $\boldsymbol{G}_{cons}$ is created to duplicate the outputs of the single agent, which are $\left[\boldsymbol{z}^T,\ \bar{\boldsymbol{y}}^T\right]^T$ for given set of inputs $\left[\boldsymbol{d}^T,\ \boldsymbol{u}^T\right]^T$. Graphical representation of the system is provided in Fig. 2.8. This system will be useful in creating the open loop cooperative system that is ready for upper and lower LFT to construct systems for synthesis and performance analysis.

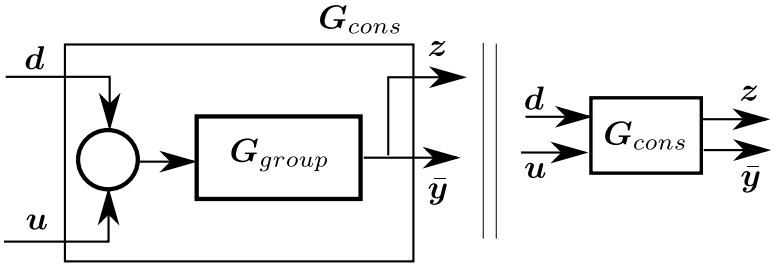


Figure 2.8. Construction of the synthetic system, $\boldsymbol{G}_{cons}$..

Finally, $\hat{\boldsymbol{G}}$ is created as depicted in Fig. 2.9. This system has a simpler representation in Fig. 2.2.

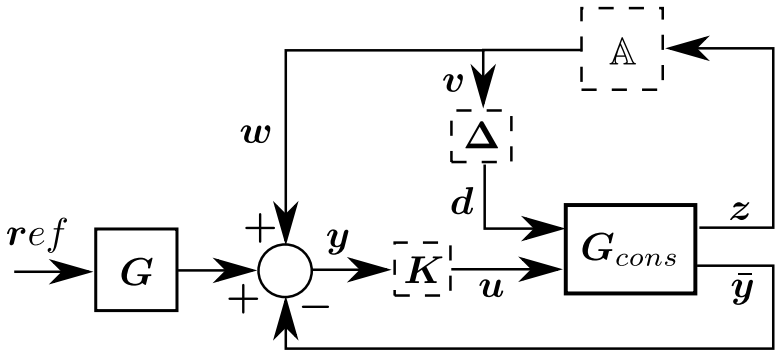


Figure 2.9. Construction of synthesis and analysis ready system, $\hat{\boldsymbol{G}}$..

2.4.2 Results for Robust Controller Synthesis Using Lumped Model

For cooperative systems with Graph topology as given in 1^{st} and 4^{th} row of the Table 2.4, performance of the cooperative system with nominal controller $\boldsymbol{K}_{nom}$ and robust controller $\boldsymbol{K}_{rob}$ are measured along with the elapsed CPU time. The decision variable size is also provided for these two Graph to illustrate the dependence of elapsed CPU time to decision variable size. For Graph topology given in 1^{st} row, $\boldsymbol{K}_{nom}$ gives $\gamma = 5.2301$ and $\boldsymbol{K}_{rob}$ gives $\gamma = 4.5602$. Elapsed CPU time for this Graph with $\boldsymbol{X} \in \mathbb{R}^{68\times 68}$ is $t_{lumped} = 542$s. Doing the same experiment for Graph topology given in 4^{th} row, $\boldsymbol{K}_{nom}$ gives $\gamma = 5.7552$ and $\boldsymbol{K}_{rob}$ gives $\gamma = 5.1702$. Elapsed CPU time for this Graph with $\boldsymbol{X} \in \mathbb{R}^{102\times 102}$ is $t_{lumped} = 3360$s.

For the cooperative system output, which is α, the objective is to track the leader, $\boldsymbol{G}_0$, which is a unit step input. *The output α in this study is used as a surrogate for the speed of each of the vehicles*. For the specified Graph (3 vehicle directed Graph) and conditions, output tracking of the agents over the grid is illustrated separately in Fig. (2.10). We can see that system with $\boldsymbol{K}_{nom}$ performs poorer in terms of rejecting disturbances than the one with $\boldsymbol{K}_{rob}$ except for the **Vehicle 1**, which is commented in Section 2.5. $\boldsymbol{K}_{rob}$ has oscillations due to constant time delay on the edges, however, the amplitude and time of the oscillations are less than the one occurs for $\boldsymbol{K}_{nom}$.

Considering the lumped methodology, results of the algorithm reveals improvement in the performance of the cooperative system as induced $\mathcal{L}_2$ gain of the system with $\boldsymbol{K}_{rob}$ get smaller. Observing the system behavior in simulation results for lumped methodology case, system under control of $\boldsymbol{K}_{rob}$ oscillates with smaller amplitudes and attenuate faster than the one with $\boldsymbol{K}_{nom}$. Referring to the results, it was observed that **Vehicle 1** with $\boldsymbol{K}_{nom}$ performs better than any vehicle with any controller. This is primarily due to the Adjacency matrix, as **Vehicle 1** is not receiving any signals back from the other vehicles which means it is also not receiving any disturbance due to time delay either. The reason **Vehicle 1**

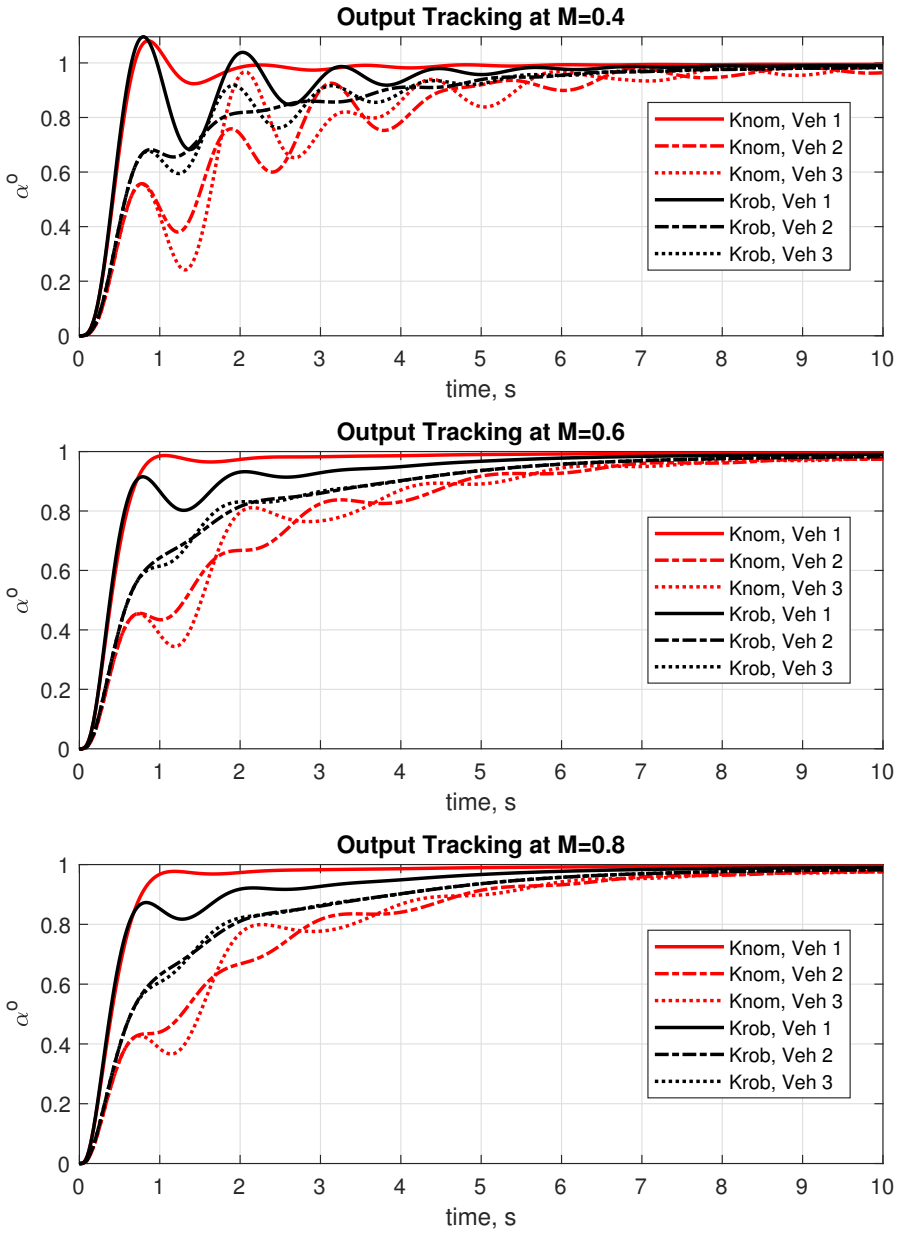


Figure 2.10. Representative case: output tracking results of cooperative system with $\Delta = \text{diag}\{0.25, 0.5, 0.75\}$ s.

with $\boldsymbol{K}_{rob}$ degrades in performance is that, the controller $\boldsymbol{K}_{rob}$ is a centralized controller and designed considering the lumped system $\hat{G}$, which has N many vehicles and N many outputs. Therefore performance of the system is checked from these multiple inputs and outputs and points out that performance of the controllers should be evaluated with respect to overall cooperative system performance.

2.4.3 Results for Robust Controller Synthesis Using Distributed Model

For the specified Graph and conditions output tracking of the agents over the grid is illustrated separately in Fig. 2.11. System with nominal controller $\boldsymbol{K}^i_{nom}$ gives $\gamma = 5.59$ and robust controller $\boldsymbol{K}^i_{rob}$ gives $\gamma = 5.22$ while the elapsed CPU time for the calculation is $t_{dist} = 306.55$s. It is observed by the induced $\mathcal{L}_2$ gains and verified in the figure that robust controller performs slightly better than the nominal controller for this configuration for each vehicle. It is realized that, cooperative systems with distributed controllers $\boldsymbol{K}^i_{nom}$ and $\boldsymbol{K}^i_{rob}$ does not induce oscillations, when it is compared to the Fig. 2.10. The settling time using the distributed methodology is slightly higher than the case with using the lumped method which is understandable. However, we see a significant reduction in the CPU time for the computation in the distributed case while recovering the robust performance.

Considering the directed path for 3 vehicle and 4 vehicle cases given in 2^{nd} and 6^{th} rows of Table 2.4, distributed synthesis method provides following results. For the cooperative system with 3 vehicles, system with nominal controller $\boldsymbol{K}^i_{nom}$ gives $\gamma = 5.412$ and $\boldsymbol{K}^i_{rob}$ gives $\gamma = 5.027$. Elapsed CPU time for this calculation is $t_{dist_d} = 287.16$s. It is observed from Fig. 2.12 that, system driven by $\boldsymbol{K}^i_{rob}$ provides better performance compared to the $\boldsymbol{K}^i_{nom}$ in terms of overshoot. Improvement in convergence to the reference is slightly better for $\boldsymbol{K}^i_{rob}$. In the same figure, control input δ_e is also provided along with the trajectory of the agents. It is illustrated in the figure that, input is bounded to 2.6° and

goes upto 3.2° for small amount of time as a consequence of the unit step reference to the cooperative system.

For the cooperative system with 4 vehicles, system with nominal controller $\boldsymbol{K}^i_{nom}$ gives $\gamma = 5.6230$ and $\boldsymbol{K}^i_{rob}$ gives $\gamma = 4.9106$. Elapsed CPU time for this calculation is $t_{dist_d} = 450.29$s.

The elapsed CPU time for all Graph topologies pertaining to the distributed methodology are plotted in Fig. 2.14. It reveals that synthesis time for distributed methodology increases almost linearly with the number of vehicles (system size N) in the cooperative system, while such a result cannot be concluded for the lumped method with the amount of experiments. Further, for the lumped case, any result for a cooperative system with 4 or higher number of vehicles was not feasible in terms of time for the system, where simulations executed. Black asterisks shows elapsed CPU time to calculate robust controller for each Graph using distributed methodology in Table 2.4. Fig. 2.14 reveals that while the configuration affects the calculation time, the dominant parameter however is the system size, N.

Finally, the γ_{rob} values obtained for 2 and 3 vehicle cases are plotted in Fig. 2.15, which shows the relation among number of vehicles, Graph topology and γ_{rob}. It is observed that the graph topology affects the induced $\mathcal{L}_2$ gain of the cooperative system and we can see that DP has the best robust performance, while DC has the least. Based on this *it would seem that*, increasing number of vehicles in the system, results in an increase in the γ_{rob}, which means a decrease in robust performance. For example in the 3 vehicle case, where γ_{rob} is best for DP but worse than the 2 vehicle case, worst for DC, and DGs being in between. However, for all cases with greater than 3 vehicles the performance is consistent with that observed for the 3 vehicles case.

Observing the system behavior in simulation results for distributed methodology case, system driven by $\boldsymbol{K}^i_{rob}$ shows better performance but improvement in the performance depends on the configuration. It should be noted that performance of $\boldsymbol{K}^i_{rob}$ cannot be worse

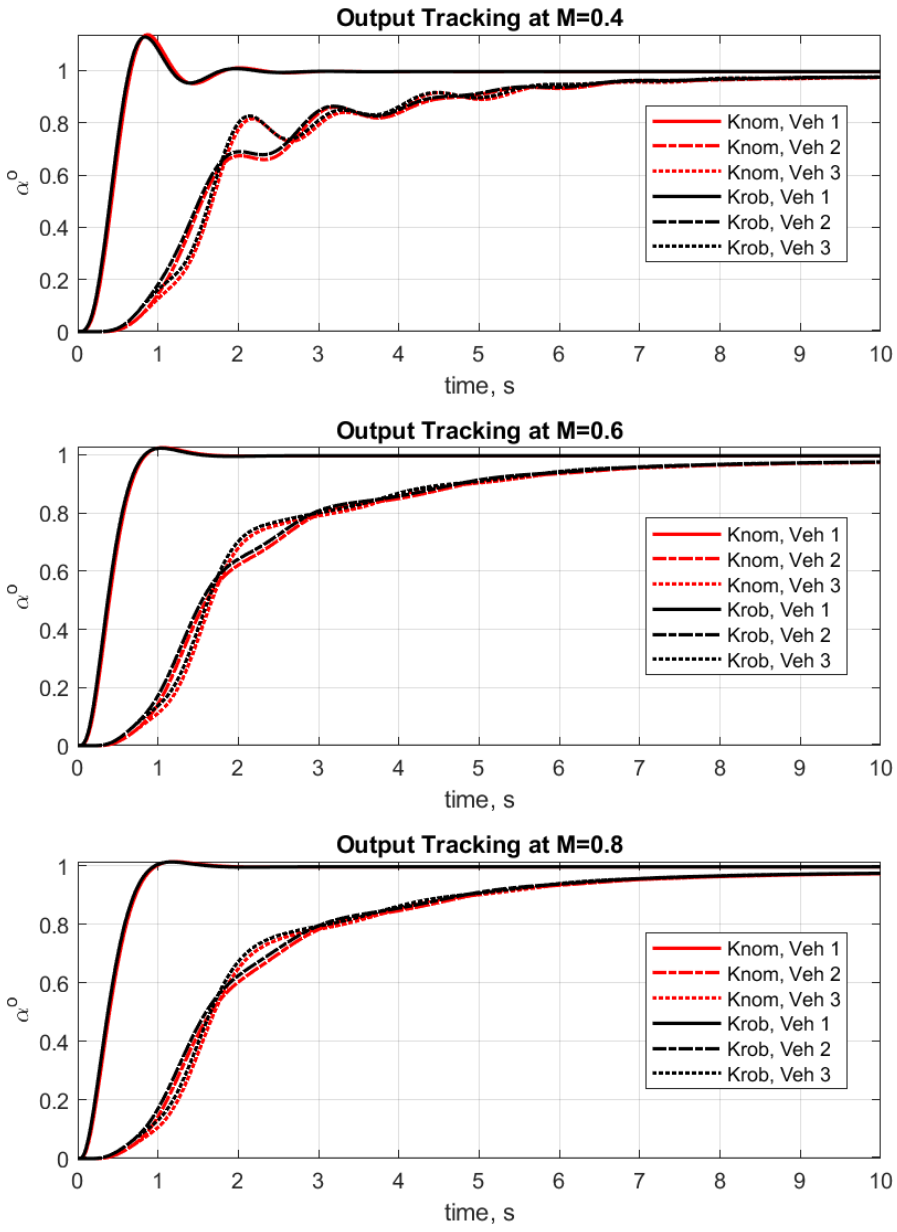


Figure 2.11. Representative case: output tracking results of cooperative system with $\Delta = \text{diag}\{0.25, 0.5, 0.75\}$ s..

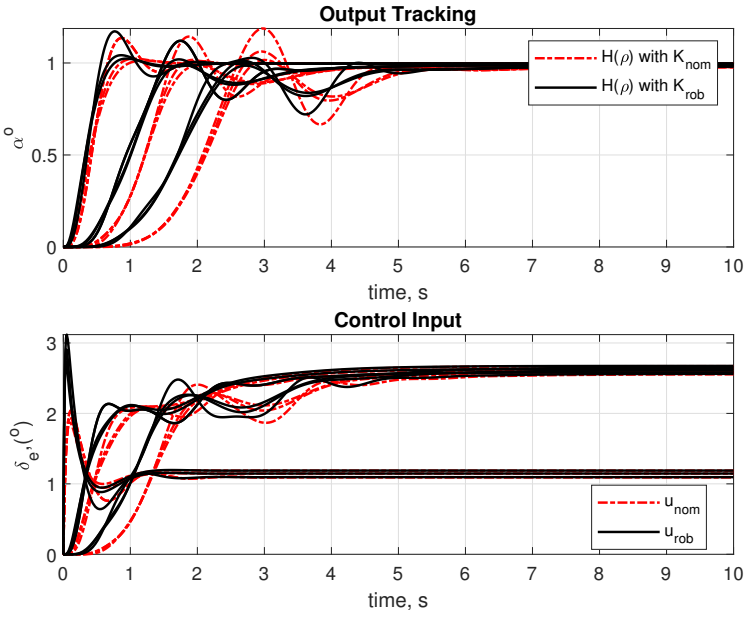


Figure 2.12. Output tracking of cooperative system with 3 vehicles and time delay of $\Delta = \text{diag}\{0.25, 0.5, 0.75\}$s..

than $\boldsymbol{K}^i_{nom}$, since the algorithm converges. The dependence on configuration is immediate if a comparison between figures Fig. 2.11 and Fig. 2.12 is made. Table 2.4 reveals that cooperative systems with higher number of agents, shows resembling performances for $\boldsymbol{K}^i_{nom}$ and $\boldsymbol{K}^i_{rob}$. This is due to the search range provided for all Graph topologies. The search range is provided by scaling the induced $\mathcal{L}_2$ norm of the system $\mathcal{F}_l\left(\tilde{\boldsymbol{G}}^i, \boldsymbol{K}^i_{nom}\right)$ up and down by 1 and 20. Therefore the bisection algorithm searches within this range and for cases enumerated as 7 to 12 it hits the minimum.

As illustrated in Table 2.4 and Figs. 2.10, 2.11, 2.12, 2.13, 2.14 and 2.15 following results can be summarized:

- The robust controller synthesis procedure achieves converging robust performance values regarding to the decrease in γ_{rob} with respect to γ_{nom} for any Graph topology.
- The synthesis times are very small compared to lumped model case.

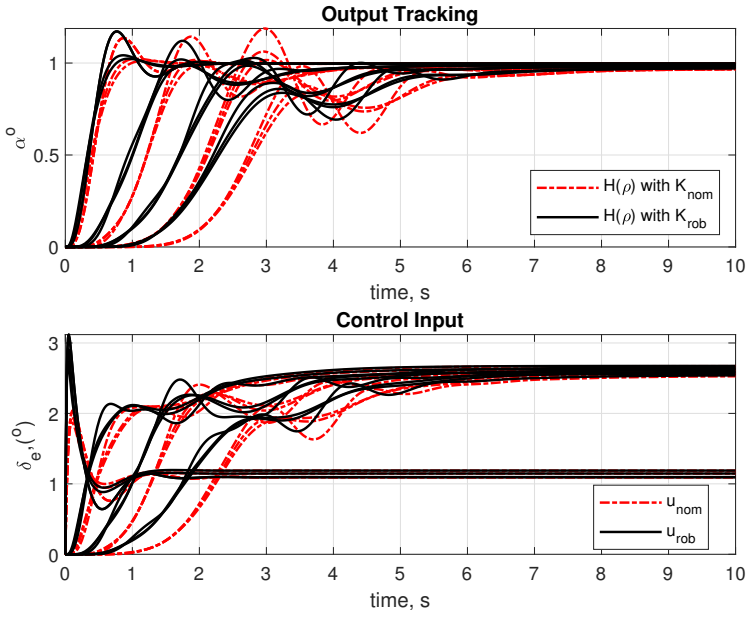


Figure 2.13. Output tracking of cooperative system with 4 vehicles and time delay of $\Delta = \text{diag}\{0.25, 0.5, 0.75, 0.25\}$ s..

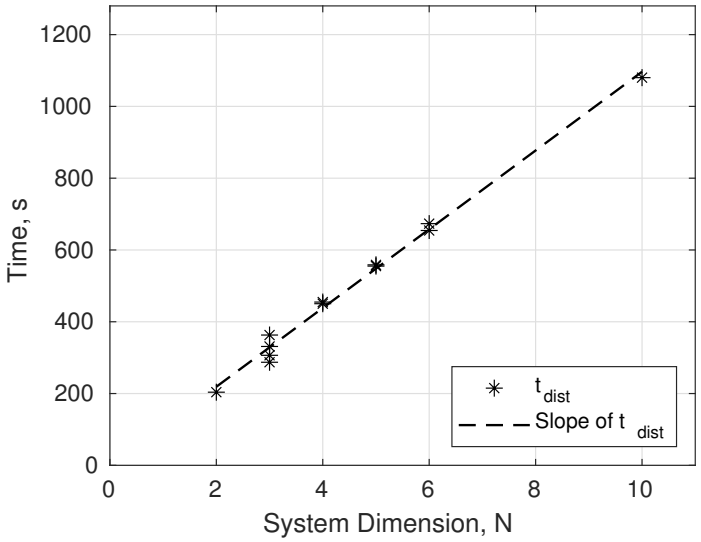


Figure 2.14. Elapsed time to find robust controller.

- The distributed synthesis is tractable in terms of calculation time as it increases linearly with system size as illustrated in Fig. 2.14. As the system size grows, calculation time for lumped model grows drastically.

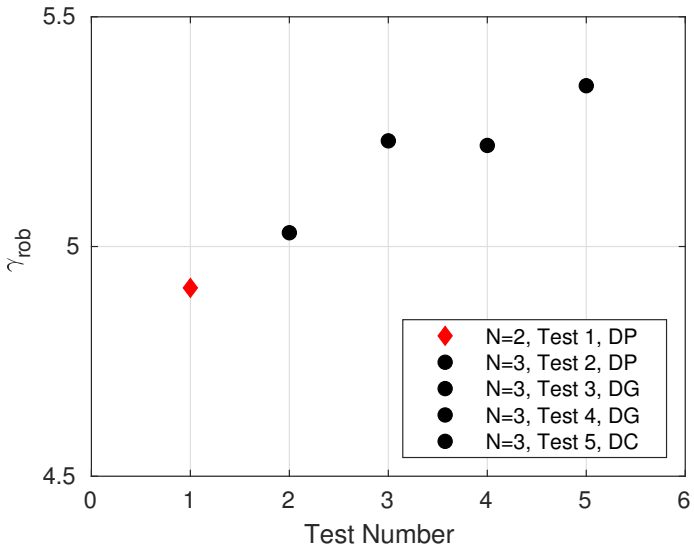


Figure 2.15. Change in γ_{rob} with system size and Graph topology. (DP: Directed Path, DG: Directed Graph, DC: Directed Cycle).

- The performance of the distributed methodology varies considerably with the varying Graph structure.

2.5 Conclusions

This paper adopts methods proposed for design of a robust controller for a single linear parameter varying system, which enables robust synthesis using IQC analysis, and successfully extends and implements them for both single and mutli-agent systems in a nested manner. The novel recasting of the dynamics of the single and mutli-agent systems, in a form such that uncertainties and controllers are connected to the system in the same manner as before, lends the methods to be tractable. This is important as there is a quadratic relation between system size and the size of the decision variable which leads to intractable simulations like the one calculated for test number four. For distributed case, synthesis has to be done for every agent and that is why calculation time increases almost linearly. The implementation of the robust controller synthesis algorithm provides a nested set

of controllers that can potentially be extended to any hierarchical framework, such as networks of networks of agents, address uncertainties at each of the layers and increase the performance of the vehicle, as well as the layer in the hierarchy and the overall system. The computation times are found to scale linearly hence manageable.

Acknowledgments

This work is supported by the Office of Naval Research via award number N00014-18-1-2215. The authors thank the anonymous reviewers for critically reading the manuscript and suggesting substantial improvements.

Chapter 3

Robust Edge Weight Synthesis for Multi Agent Systems With Integral Quadratic Constraints

3.1 Preliminaries

3.1.1 Notation

Let $\mathbb{R}$ and $\mathbb{R}_{++}$ denote real and positive real numbers, respectively. $\mathbb{R}_{+}$ be real number that is equal or greater than 0. $\boldsymbol{S}^{p}$ denotes symmetric matrix of size p. A_{ij} represents element in the i^{th} row j^{th} column of matrix $\boldsymbol{A}$. Matrix inequality conditions are defined with $<$ and $>$, which stands for $\leq$ and $\geq$, respectively. A matrix $\boldsymbol{A} := diag(\cdot)$ is a block matrix, where diagonal entries are the arguments of $diag$. A matrix $\boldsymbol{A} := col(\cdot)$ is a block matrix, where vertical entries are the arguments of col. Finally, $\mathcal{F}_L$ and $\mathcal{F}_U$ stands for lower and upper LFT.

3.1.2 Linear Parameter Varying Systems

The transfer matrix $\boldsymbol{G}(\boldsymbol{\rho})$ of an LPV system with input $\boldsymbol{w} \in \mathbb{R}^{n_w}$ and output $\boldsymbol{z} \in \mathbb{R}^{n_z}$ can be defined over the feasible parameter trajectory $\boldsymbol{\iota}(\boldsymbol{\rho})$ as

$$\boldsymbol{G}(\boldsymbol{\rho}) := \left[\begin{array}{c|c} \boldsymbol{A}(\boldsymbol{\rho}) & \boldsymbol{B}(\boldsymbol{\rho}) \\ \hline \boldsymbol{C}(\boldsymbol{\rho}) & \boldsymbol{D}(\boldsymbol{\rho}) \end{array}\right], \tag{3.1}$$

$$\boldsymbol{\iota}(\boldsymbol{\rho}) := \{\boldsymbol{\rho} \in \mathcal{P} \subset \mathbb{R}^{p},\ |\dot{\boldsymbol{\rho}}(t)| < r \geq 0\ \ \forall t \geq 0\}$$

where $\boldsymbol{A}(\boldsymbol{\rho}) \in \mathbb{R}^{n_x \times n_x}$, $\boldsymbol{B}(\boldsymbol{\rho}) \in \mathbb{R}^{n_x \times n_w}$, $\boldsymbol{C}(\boldsymbol{\rho}) \in \mathbb{R}^{n_z \times n_w}$, and $\boldsymbol{D}(\boldsymbol{\rho}) \in \mathbb{R}^{n_z \times n_w}$ are the state-space matrices, and $\boldsymbol{\rho}(t) = [\rho_1, \rho_2, \ldots, \rho_p]$ is an exogeneous parameter vector with bounded derivatives.

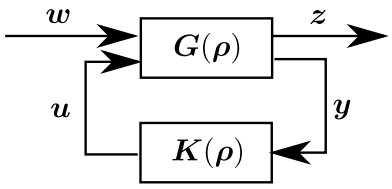


Figure 3.1. LPV system $\boldsymbol{H}(\boldsymbol{\rho})$ maps input $\boldsymbol{w}$ to output $\boldsymbol{z}$..

Let us consider the transfer matrix $\boldsymbol{H}(\boldsymbol{\rho})$ illustrated in Fig. 3.1, which is the lower-fractional transformation (LFT) between $\boldsymbol{G}(\boldsymbol{\rho})$ and the controller $\boldsymbol{K}(\boldsymbol{\rho})$ defined as $\boldsymbol{H}(\boldsymbol{\rho}) := \mathcal{F}_l(\boldsymbol{G}(\boldsymbol{\rho}), \boldsymbol{K}(\boldsymbol{\rho}))$. The performance of the closed-loop LPV system can be measured in terms of induced $\mathcal{L}_2$ gain of the input/output map with zero initial conditions, and is defined as [45],

$$\|\boldsymbol{H}(\rho)\| := \sup_{\rho \in \mathcal{A}} \sup_{\substack{\boldsymbol{w} \in \mathcal{L}_2 \\ \|\boldsymbol{w}\| \neq 0}} \frac{\|\boldsymbol{z}\|_2}{\|\boldsymbol{w}\|_2} \tag{3.2}$$

The Bounded Real Lemma for linear time-invariant (LTI) systems can be extended to obtain the upper bound of the induced $\mathcal{L}_2$ gain of the LPV system [45]. As discussed in [115, 116], an LPV system $\boldsymbol{H}(\boldsymbol{\rho})$ is exponentially stable over the parameter bounded set $\mathcal{P}$ and $\|\boldsymbol{H}(\boldsymbol{\rho})\| < \gamma$ if there exists a differentiable matrix function $\boldsymbol{X}(\boldsymbol{\rho}) = \boldsymbol{X}^T(\boldsymbol{\rho})$ such that,

$$\begin{aligned} \boldsymbol{X}(\boldsymbol{\rho}) &\geq \boldsymbol{0} \\ \begin{bmatrix} \partial \boldsymbol{X}\,(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}}) + \boldsymbol{A}^T(\boldsymbol{\rho})\boldsymbol{X}(\boldsymbol{\rho}) + \boldsymbol{X}(\boldsymbol{\rho})\boldsymbol{A}(\boldsymbol{\rho}) & \boldsymbol{X}(\boldsymbol{\rho})\boldsymbol{B}(\boldsymbol{\rho}) \\ \boldsymbol{B}^T(\boldsymbol{\rho})\boldsymbol{X}(\boldsymbol{\rho}) & \boldsymbol{0} \end{bmatrix} & \\ + \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{C}(\boldsymbol{\rho}) & \boldsymbol{D}(\boldsymbol{\rho}) \end{bmatrix}^T \boldsymbol{P} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{C}(\boldsymbol{\rho}) & \boldsymbol{D}(\boldsymbol{\rho}) \end{bmatrix} &\leq \boldsymbol{0} \end{aligned} \tag{3.3}$$

where $\boldsymbol{P} = \begin{bmatrix} -\gamma^2 \boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix}$ and $\gamma > 0$. In Eq. (3.3), $\partial \boldsymbol{X}\left(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}}\right)$ is defined as [45]

$$\partial \boldsymbol{X}\left(\boldsymbol{\rho}\left(t\right), \dot{\boldsymbol{\rho}}\left(t\right)\right) \;=\; \frac{d}{dt}\,\boldsymbol{X}\left(\boldsymbol{\rho}\left(t\right)\right) \;=\; \sum_{i=1}^{p} \frac{\partial \boldsymbol{X}\left(\boldsymbol{\rho}\left(t\right)\right)}{\partial \boldsymbol{\rho}_i\left(t\right)} \dot{\boldsymbol{\rho}}_i\left(t\right) \tag{3.4}$$

Equation (3.3) introduces parameter dependent Linear Matrix Inequalities. Moreover, if the system defined in Eq. 3.1 is an affine function on the set $\mathcal{P}$, then $\boldsymbol{X}(\boldsymbol{\rho})$ becomes stationary and $\partial \boldsymbol{X}(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}}) = \mathbf{0}$. Eventually, the resulting LMI must be satisfied on the set $\mathcal{P}$, thereby resulting in finite set of LMIs. Dropping the argument $\boldsymbol{\rho}$ from the matrices for the sake of brevity, the LMIs can be written as

$$\begin{aligned} &\boldsymbol{X} \geq \mathbf{0} \\ &\begin{bmatrix} \boldsymbol{A}^T\boldsymbol{X} + \boldsymbol{X}\boldsymbol{A} & \boldsymbol{X}\boldsymbol{B} \\ \boldsymbol{B}^T\boldsymbol{X} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \boldsymbol{I} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix}^T \boldsymbol{P} \begin{bmatrix} \mathbf{0} & \boldsymbol{I} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix} \leq \mathbf{0} \end{aligned} \tag{3.5}$$

3.1.3 Multi-agent System Modelling

Underlying Graph Structure. A MAS is represented by a Graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, which is a pair of a node set $\mathcal{N}$ and a edge set $\mathcal{E} \subset \mathcal{N} \times \mathcal{N}$. Edge is described between nodes $i \in \mathcal{N}$ and $j \in \mathcal{N}$ such that $(j, i) \in \mathcal{E}$ denotes that node j transmits information to node i. The adjacency matrix $\mathbb{A} = [a_{ij}] \in \mathbb{R}^{N \times N}$ of the Graph of N agents is defined as in the following equation. The weighting scalar a_{ij} is called as an edge weight from j^{th} to i^{th} agent. This connection is graphically illustrated as in Figure 3.2.

$$\mathbb{A}_{ij} = \begin{cases} a_{ij} > 0, \quad j \neq i, \;\; (j,i) \in \mathcal{E} \\ a_{ij} = 0, \quad otherwise \end{cases} \tag{3.6}$$

Construction of MAS. A group of unconnected agents, which is denoted by $\boldsymbol{G}^i$, can be treated as a single system, which is denoted with $\boldsymbol{S}$ and represented in state space form as

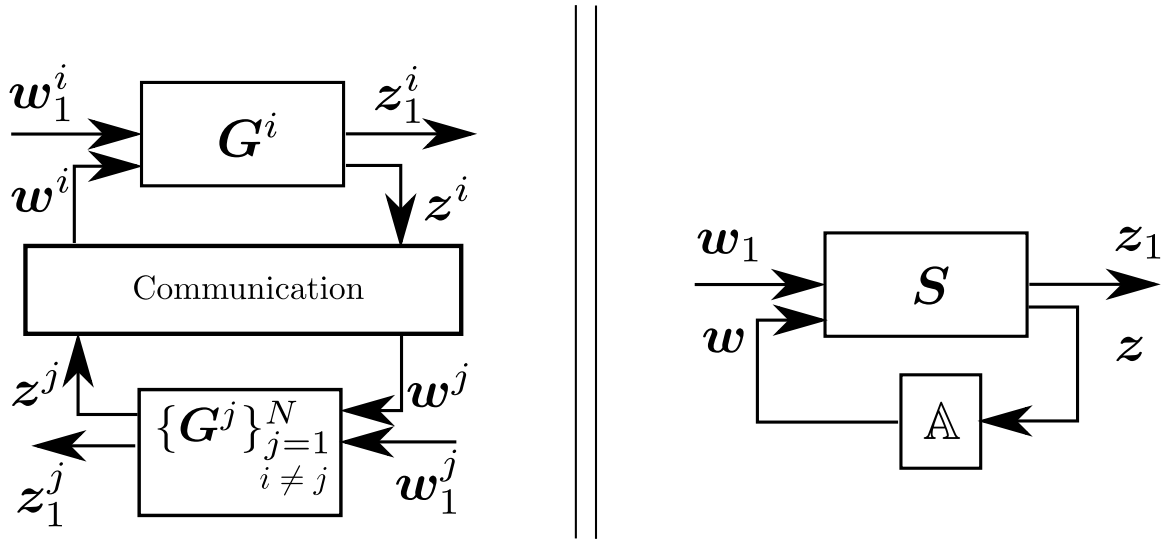


Figure 3.2. Interconnection of agents and lumped representation of MAS $\boldsymbol{H} = \mathcal{F}_L(\boldsymbol{S}, \mathbb{A})$.
.

in (6.22). $\boldsymbol{S}$ is simply a block diagonal concatenation of $\boldsymbol{G}^i$ such that $\boldsymbol{S} = diag\{\boldsymbol{G}^i\}_{i=1}^N$. Inputs and outputs of the system are $\left[\boldsymbol{w}_1^T \ \boldsymbol{w}^T\right]^T$ and $\left[\boldsymbol{z}_1^T \ \boldsymbol{z}^T\right]^T$, where $\boldsymbol{w}_1 \in \mathbb{R}^{n_{w_1}}$, $\boldsymbol{w} \in \mathbb{R}^{n_w}$, $\boldsymbol{z}_1 \in \mathbb{R}^{nz_1}$, and $\boldsymbol{z} \in \mathbb{R}^{n_z}$. Vehicles within the system shares their output information with regarding agents, and these shared signals are called as spatial signals, which flow through the input output ports denoted as $\boldsymbol{w} - \boldsymbol{z}$ such that $\boldsymbol{w} = \mathbb{A}\boldsymbol{z}$. Channel through ports $\boldsymbol{w}_1 - \boldsymbol{z}_1$ carry the information of reference input and output of the system. States of the system are defined as $\boldsymbol{x} \in \mathbb{R}^{n_x}$.

$$\left[\begin{array}{c} \dot{\boldsymbol{x}} \\ \hline \boldsymbol{z}_1 \\ \boldsymbol{z} \end{array}\right] = \left[\begin{array}{c|cc} \boldsymbol{A} & \boldsymbol{B}_1 & \boldsymbol{B} \\ \hline \boldsymbol{C}_1 & \boldsymbol{D}_1 & \boldsymbol{E}_1 \\ \boldsymbol{C} & \boldsymbol{F}_1 & \boldsymbol{0} \end{array}\right] \left[\begin{array}{c} \boldsymbol{x} \\ \hline \boldsymbol{w}_1 \\ \boldsymbol{w} \end{array}\right] \tag{3.7}$$

A cooperative system, $\boldsymbol{H}$, can be constructed using an adjacency matrix ($\mathbb{A}$) to connect vehicles to each other as represented in lower linear fractional transformation (LFT) such that $\boldsymbol{H} = \mathcal{F}_L(\boldsymbol{S}, \mathbb{A})$. The state space representation of $\boldsymbol{H}$ is provided in (6.23). The connection couples vehicles with a strength prescribed by a weighting scalar a_{ij}.

$$\begin{aligned}\left[\begin{array}{c|c} \boldsymbol{\mathcal{A}} & \boldsymbol{\mathcal{B}}_1 \\ \hline \boldsymbol{\mathcal{C}}_1 & \boldsymbol{\mathcal{D}}_1 \end{array}\right] &= \left[\begin{array}{c|c} \boldsymbol{A} & \boldsymbol{B}_1 \\ \hline \boldsymbol{C}_1 & \boldsymbol{D}_1 \end{array}\right] + \left[\begin{array}{c} \boldsymbol{B} \\ \boldsymbol{E}_1 \end{array}\right] \mathbb{A} \left[\begin{array}{cc} \boldsymbol{C} & \boldsymbol{F}_1 \end{array}\right] \\ &= \left[\begin{array}{c|c} \boldsymbol{A} + \boldsymbol{B}\mathbb{A}\boldsymbol{C} & \boldsymbol{B}_1 + \boldsymbol{B}\mathbb{A}\boldsymbol{F}_1 \\ \hline \boldsymbol{C}_1 + \boldsymbol{E}_1\mathbb{A}\boldsymbol{C} & \boldsymbol{D}_1 + \boldsymbol{E}_1\mathbb{A}\boldsymbol{F}_1 \end{array}\right] \end{aligned} \tag{3.8}$$

3.1.4 Feasibility Condition for H_∞ Performance

Assuming system given in (6.23) for a given $\mathbb{A}$ is asymptotically stable, the supremum of the maximum singular value of the system is given by the scalar γ, which is also called as induced $\mathcal{L}_2$ norm of the system. Latter result is provided by the following theorem.

Theorem 3.1.1 *Assuming system in* (6.23) *is asymptotically stable then* $\|\boldsymbol{H}\|_\infty < \gamma$ *and there is a solution for* $\boldsymbol{\mathcal{X}}$ *such that linear matrix inequalities (LMIs) in* (6.24) *holds.*

$$\begin{aligned} \boldsymbol{\mathcal{X}} &\geq \mathbf{0} \\ \left[\begin{array}{c} \boldsymbol{\mathcal{X}} \\ \mathbf{0} \end{array}\right] \left[\begin{array}{cc} \boldsymbol{\mathcal{A}} & \boldsymbol{\mathcal{B}}_1 \end{array}\right] + \left[\begin{array}{c} \boldsymbol{\mathcal{A}}^T \\ \boldsymbol{\mathcal{B}}_1^T \end{array}\right] \left[\begin{array}{cc} \boldsymbol{\mathcal{X}} & \mathbf{0} \end{array}\right] & \\ + \left[\begin{array}{c} \boldsymbol{\mathcal{C}}_1^T \\ \boldsymbol{\mathcal{D}}_1^T \end{array}\right] \left[\begin{array}{cc} \boldsymbol{\mathcal{C}}_1 & \boldsymbol{\mathcal{D}}_1 \end{array}\right] + \left[\begin{array}{cc} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\gamma^2\boldsymbol{I} \end{array}\right] &\leq \mathbf{0} \end{aligned} \tag{3.9}$$

Given new variable names such that $\mathbb{X} = \left[\begin{array}{cc} \boldsymbol{\mathcal{X}} & \mathbf{0} \end{array}\right]$, $\mathbb{Y} = \left[\begin{array}{cc} \boldsymbol{\mathcal{A}} & \boldsymbol{\mathcal{B}}_1 \end{array}\right]$ and $\mathbb{Z}$, which collects the remaining terms of second inequality in (6.24), inequality constraints given in (6.24) has the following form as illustrated in (3.10).

$$\begin{aligned} \boldsymbol{\mathcal{X}} &\geq \mathbf{0} \\ \mathbb{X}^T\mathbb{Y} + \mathbb{Y}^T\mathbb{X} + \mathbb{Z} &\leq \mathbf{0} \end{aligned} \tag{3.10}$$

Proof 3.1.2 (Proof of Theorem 3.1.1) *Proof of the theorem comes after pre and post multiplying the second inequality in* (6.24) *with* $\begin{bmatrix} \boldsymbol{x}^T & \boldsymbol{w}_1^T \end{bmatrix}$ *and* $\begin{bmatrix} \boldsymbol{x}^T & \boldsymbol{w}_1^T \end{bmatrix}^T$*, which yields* $\gamma^2\|\boldsymbol{w}_1\|_2^2 - \|\boldsymbol{z}_1\|_2^2 < \boldsymbol{0}$*. Equivalently this means that* $\|\boldsymbol{H}\|_\infty < \gamma$*.*

If $\mathbb{A}$ is not predefined by the designer and needs to be synthesized, second inequality given in (3.10) depends non-linearly on matrix variables $\boldsymbol{\mathcal{X}}$ and $\mathbb{A}$. This non-linear matrix inequality is called as BMI and can be solved using sequential LMI approach, which will be discussed in the following section.

Definition 3.1.1 *[121] A mapping* $\boldsymbol{G}(\boldsymbol{\mu}) : \mathbb{R}^m \rightarrow \boldsymbol{S}^p$ *is positive semi-definite convex(psd-convex) on a convex subset* $C \subseteq \mathbb{R}^m$ *if the inequality*

$$\eta\boldsymbol{G}(\boldsymbol{\mu}) + (1-\eta)\,\boldsymbol{G}(\boldsymbol{\nu}) \geq \boldsymbol{G}(\eta\boldsymbol{\mu} + (1-\eta)\,\boldsymbol{\nu}) \tag{3.11}$$

holds for any $\boldsymbol{\mu},\ \boldsymbol{\nu} \in C$ *and all* $\eta \in [0,\ 1]$*.*

On a convex subset C, mapping $\boldsymbol{G}(\boldsymbol{\mu})$ is said to be differentiable if its derivative $D\boldsymbol{G}(\boldsymbol{\mu})$ exist for all $\boldsymbol{\mu}$.

Definition 3.1.2 *[121, 122] Derivative of the mapping* $\boldsymbol{G}(\boldsymbol{\mu})$ *at* $\boldsymbol{\mu}$ *is a linear mapping* $D\boldsymbol{G}(\boldsymbol{\mu})$ *from* $\mathbb{R}^m$ *to* $\mathbb{R}^{p\times p}$ *that is defined as*

$$D\boldsymbol{G}(\boldsymbol{\mu})h \ := \ \sum_{i=1}^{m} h_i \frac{\partial \boldsymbol{G}}{\partial \mu_i}(\boldsymbol{\mu}),\ \ \forall h \in \mathbb{R}^m. \tag{3.12}$$

where h *is a real valued function on* μ*.*

Lemma 3.1.3 *[121, 122] A matrix valued mapping* $\boldsymbol{G}(\boldsymbol{\mu})$ *is psd-convex on a convex subset* C *if and only if* $\boldsymbol{r}^T\boldsymbol{G}(\boldsymbol{\mu})\boldsymbol{r}$ *is convex on* C*. Proof is provided in [121].*

Lemma 3.1.4 *[122] A matrix valued mapping* $\boldsymbol{G}(\boldsymbol{\mu})$ *is psd-convex on a convex subset* C *if and only if inequality in* (3.13) *holds for all* $\boldsymbol{\mu}$ *and* $\boldsymbol{\nu}$*.*

$$\boldsymbol{G}(\boldsymbol{\nu}) - \boldsymbol{G}(\boldsymbol{\mu}) \geq D\boldsymbol{G}(\boldsymbol{\mu})(\boldsymbol{\nu} - \boldsymbol{\mu}) \tag{3.13}$$

Sequential LMI approach relies on a decomposition called as convex-concave decomposition. Multiple convex-concave decompositions provided in literature [122,123] and as expressed in [122] different decompositions uniquely shifts the curvature between convex and concave parts.

For given affine matrices $\mathbb{X}$ and $\mathbb{Y}$, a bilinear matrix mapping can be defined as $\boldsymbol{M}(\mathbb{X},\mathbb{Y}) := \mathbb{X}^T\mathbb{Y} + \mathbb{Y}^T\mathbb{X}$ and following lemma provide an alternate way to verify the convexity of the matrix function $\boldsymbol{M}(\mathbb{X},\mathbb{Y})$.

Lemma 3.1.5 *[123] Affine functions $\mathbb{X}(\eta)$ and $\mathbb{Y}(\beta)$ are defined as in* (3.14) *for independent variables η_i and β_j for $i = 1,\ldots,k$ and $j = 1,\ldots,l$ respectively.*

$$\begin{aligned} \mathbb{X} &= \mathbb{X}_0 + \sum_{i=1}^{k} \eta_i \mathbb{X}_i \\ \mathbb{Y} &= \mathbb{Y}_0 + \sum_{j=1}^{l} \beta_j \mathbb{Y}_j \end{aligned} \tag{3.14}$$

A quadratic mapping can be defined based the bilinear mapping $\boldsymbol{M}(\mathbb{X},\mathbb{Y}) := \mathbb{X}^T\mathbb{Y} + \mathbb{Y}^T\mathbb{X}$ by redefining $\boldsymbol{M}(\mathbb{X},\mathbb{Y})$ as in (3.15)*. Then $\boldsymbol{M}(\mathbb{X},\mathbb{Y})$ is psd-convex on a convex subset $P \subseteq \mathbb{R}^{k+l}$ if the matrix $\boldsymbol{Q}$ is positive semi-definite. Proof of the lemma is provided in [123].*

$$\boldsymbol{M}(\mathbb{X},\mathbb{Y}) := \begin{bmatrix} \mathbb{X}^T & \mathbb{Y}^T \end{bmatrix} \boldsymbol{Q} \begin{bmatrix} \mathbb{X} \\ \mathbb{Y} \end{bmatrix} \tag{3.15}$$

3.1.5 Integral Quadratic Constraints

IQCs are widely used tools to capture behaviour uncertainties in dynamic systems by changing uncertain portions of the system with a quadratic constraint on its inputs and outputs [124]. Let us define uncertainties by a bounded casual operator Δ i.e. $\Delta \in \boldsymbol{\Delta}$, where $\boldsymbol{\Delta} = \{\Delta_m \mid \Delta_m : \mathcal{L}_{2e}^{n_z}[0,\infty) \to \mathcal{L}_{2e}^{n_w}[0,\infty), 1 \le m \le M\}$. Signals $\boldsymbol{z} \in \mathcal{L}_2^{n_z}[0,\infty)$

and $\boldsymbol{w} \in \mathcal{L}_2^{n_w}[0,\infty)$ has following relation $\boldsymbol{w}(t) = \Delta(\boldsymbol{z}(t))$ and has Fourier transforms of $\hat{Z}(j\omega)$ and $\hat{W}(j\omega)$.

Suppose $\mathbf{\Pi} \in \mathbb{RL}_\infty^{(n_z+n_w)\times(n_z+n_w)}$ is a bounded rational weighting function that defines an IQC, then signals $\boldsymbol{z}$ and $\boldsymbol{w}$ satisfy the frequency domain IQC, $\mathbf{\Pi}$, if inequality in (3.16) holds true.

$$\int_{-\infty}^{\infty} \begin{bmatrix} \hat{Z}(i\omega) \\ \hat{W}(i\omega) \end{bmatrix}^* \mathbf{\Pi}(i\omega) \begin{bmatrix} \hat{Z}(i\omega) \\ \hat{W}(i\omega) \end{bmatrix} d\omega \geq 0 \tag{3.16}$$

By factorizing $\mathbf{\Pi}(i\omega) = \mathbf{\Psi}(i\omega)^* \, \boldsymbol{P} \, \mathbf{\Psi}(i\omega)$, a dynamic filter ($\mathbf{\Psi}(i\omega)$) and a correlator ($\boldsymbol{P}$) is obtained. Here $\mathbf{\Psi}(i\omega)$ is given in state space form as in (3.17) and inputs and outputs of this filter are $\begin{bmatrix} \boldsymbol{w} & \boldsymbol{z} \end{bmatrix}^T$ and $\boldsymbol{z}_\psi$ respectively.

$$\mathbf{\Psi} = \left[\begin{array}{c|c} \boldsymbol{A}_\psi & \boldsymbol{B}_\psi \\ \hline \boldsymbol{C}_\psi & \boldsymbol{D}_\psi \end{array}\right], \qquad \boldsymbol{z}_\psi = \mathbf{\Psi} \begin{bmatrix} \boldsymbol{d} \\ \boldsymbol{v} \end{bmatrix} \tag{3.17}$$

Time domain form of (3.16) can be written in the quadratic form as illustrated in (3.18). Then we can say that signals $\boldsymbol{z}$ and $\boldsymbol{w}$ satisfy the time domain IQC in (3.18).

$$\int_0^T \boldsymbol{z}_\psi^T \boldsymbol{P} \boldsymbol{z}_\psi dt \geq 0 \tag{3.18}$$

In this work, J-spectral factorization [65] is used to decompose IQC multipliers. For these multipliers to be valid time domain IQCs, following assumptions has to be satisfied.

Assumption 3.1.6 *Let IQC multiplier* $\mathbf{\Pi}$ *be partitioned as in* (3.19)*. Then multiplier should satisfy* $\mathbf{\Pi}_{11}(j\omega) \geq 0$ *and* $\mathbf{\Pi}_{22}(j\omega) \leq 0, \ \forall\omega \cup \{\infty\}$.

$$\mathbf{\Pi} := \left[\begin{array}{c|c} \mathbf{\Pi}_{11} & \mathbf{\Pi}_{12} \\ \hline \mathbf{\Pi}_{21} & \mathbf{\Pi}_{22} \end{array}\right] \tag{3.19}$$

Assumption 3.1.7 *The uncertainty block has to satisfy* $\|\Delta\| \leq 1$. *This is ensured by assigning* $\boldsymbol{\Pi}_1 = diag\{\boldsymbol{I}_{nz_1}, -\boldsymbol{I}_{nw_1}\}$.

3.2 Methodology

This section devices sLMI method to synthesize adjacency matrix $\mathbb{A}$. This is done without uncertainty using the representation given in (6.23) in subsection 3.2.1. Then an uncertainty channel is added to the MAS and an augmented sLMI method is proposed to incorporate IQC analysis along with the performance certification in subsection 3.2.2.

3.2.1 Adjacency Matrix Synthesis for Nominal H_∞ Performance

Originally, the optimization problem for synthesizing $\mathbb{A}$ is defined formally in (3.20), which is a optimization with a BMI constraint.

$$\begin{aligned} &\text{minimize} \quad \gamma \\ &\text{s. t.} \qquad \boldsymbol{\mathcal{X}} \geq \mathbf{0} \\ &\qquad\qquad \mathbb{X}^T\mathbb{Y} + \mathbb{Y}^T\mathbb{X} + \mathbb{Z} \leq \mathbf{0} \end{aligned} \tag{3.20}$$

This optimization problem can be rewritten using convex-concave decomposition [122]. In this work, a decomposition proposed in [123] is selected. Following this decomposition, left hand side of the BMI constraint given in (3.20) is fractioned into $\boldsymbol{M}_1 = \mathbb{X}^T\mathbb{Y} + \mathbb{Y}^T\mathbb{X}$ and $\boldsymbol{M}_2 = \mathbb{Z}$. Then $\boldsymbol{M}_1$ is redefined in a quadratic form as provided in (3.21).

$$\boldsymbol{M}_1 = \mathbb{X}^T\mathbb{Y} + \mathbb{Y}^T\mathbb{X} = \begin{bmatrix} \mathbb{X}^T & \mathbb{Y}^T \end{bmatrix} \begin{bmatrix} \mathbf{0} & \boldsymbol{I} \\ \boldsymbol{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbb{X} \\ \mathbb{Y} \end{bmatrix} \tag{3.21}$$

The indefinite matrix in the quadratic form is then decomposed into two positive semi-definite matrices, which are calculated as given in (3.22) and leads to a convex and a concave matrix definitions. The decomposition uses generalized left eigenvectors defined

as $\boldsymbol{v}$ and eigenvalues that are diagonally blocked in $\boldsymbol{\lambda} := diag(-\boldsymbol{I}, \boldsymbol{I})$ with respective sizes. When the diagonal eigenvalues are separated into two matrices while securing the dimension of the original matrix, equation of (3.22) is obtained.

$$\begin{bmatrix} \mathbf{0} & \boldsymbol{I} \\ \boldsymbol{I} & \mathbf{0} \end{bmatrix} = \boldsymbol{v} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix} \boldsymbol{v}^T - \boldsymbol{v} \begin{bmatrix} \boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \boldsymbol{v}^T \tag{3.22}$$

Then, $\boldsymbol{M}_1$ can be written using the convex-concave matrix definitions given in (3.22), which is explicitly provided in (3.23).

$$\begin{aligned} \boldsymbol{M}_1 &= \boldsymbol{M}_{11} - \boldsymbol{M}_{12} = \\ &\begin{bmatrix} \mathbb{X}^T & \mathbb{Y}^T \end{bmatrix} \boldsymbol{v} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix} \boldsymbol{v}^T \begin{bmatrix} \mathbb{X} \\ \mathbb{Y} \end{bmatrix} \\ &- \begin{bmatrix} \mathbb{X}^T & \mathbb{Y}^T \end{bmatrix} \boldsymbol{v} \begin{bmatrix} \boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \boldsymbol{v}^T \begin{bmatrix} \mathbb{X} \\ \mathbb{Y} \end{bmatrix} \end{aligned} \tag{3.23}$$

Finally BMI constraint given in (3.20) can be collected such that, $\boldsymbol{M}_{11}$ and $\boldsymbol{M}_2$ is collected under a single matrix variable, which is named as $\boldsymbol{M}_{11,2}$, and $\boldsymbol{M}_{12}$ is left by itself. Using these matrix definitions, decomposed constraint an optimization problem can be written as provided in (3.24).

$$\begin{aligned} &\text{minimize} \quad \gamma \\ &\text{s. t.} \qquad \boldsymbol{\mathcal{X}} \geq \mathbf{0} \\ &\qquad\qquad \boldsymbol{M}_{11,2} - \boldsymbol{M}_{12} \leq \mathbf{0} \end{aligned} \tag{3.24}$$

Recalling that the decision variables for this optimization problem are $\boldsymbol{\mathcal{X}}$ and $\mathbb{A}$, sequential LMI approach requires the initial conditions for these matrix variables to be in a relative interior (strictly feasible ball) to start. This requirement is satisfied by the following strategy. $\mathbb{A}$, which is given by the user can be used as an initial point. Using the standard

optimization problem given in (3.25), we are able to find a strictly feasible initial solution denoted as $\boldsymbol{\mu}^0 = \{\boldsymbol{\mathcal{X}}^0,\ \gamma^0\}$ for given $\mathbb{A}^0$ assuming that such a solution exist for given adjacency matrix. This can be verified easily by checking if cooperative system given in (6.23) is Hurwitz. This automatically ensures that there exists a decision variable $\boldsymbol{\mathcal{X}}$ for the problem defined in (3.25) based on the result given in Theorem 3.1.1. In this equation $\boldsymbol{C}(\boldsymbol{\mathcal{X}})$ represents the set of constraints, which is provided in (6.24).

$$\begin{aligned} &\text{minimize} \quad \gamma \\ &\text{s. t.} \qquad\quad \boldsymbol{C}(\boldsymbol{\mathcal{X}}) \end{aligned} \tag{3.25}$$

The solutions $\boldsymbol{\mathcal{X}}^0$, γ^0 and given initial adjacency $\mathbb{A}^0$ is then passed to the sLMI algorithm, which will be expressed in the following text. sLMI approach is an iterative procedure and in the formulations same variable will have its existing solution and future solution. Existing solution of any variable have a superscript t as $\{\cdot\}^t$, while future solution will be expressed as is.

Considering the matrix decision variables $\mathbb{X}$ and $\mathbb{Y}$, we can collect the decision variables constructing them within a vector $\boldsymbol{\mu}$ such that $\boldsymbol{\mu} = [vec(\mathbb{X}),\ vec(\mathbb{Y})]$. Complying with the notation, their existing and future representations are provided as $\boldsymbol{\mu}^t$ and $\boldsymbol{\mu}$ respectively. Another point to be expressed here is the partial derivatives of the matrix decision variables at existing solution with respect to $\boldsymbol{\mu}^t$. To do that a formal partial derivative of $\mathbb{X}$ is taken as provided in (4.38).

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{\mu}^t}\{\mathbb{X}\} &= \sum_{i=1}^{n} \left(\eta_i - \eta_i^t\right) \mathbb{X}_i^t \\ &= \mathbb{X}_0 + \sum_{i=1}^{n} \left(\eta_i\right) \mathbb{X}_i - \mathbb{X}_0 - \sum_{i=1}^{n} \left(\eta_i^t\right) \mathbb{X}_i^t \end{aligned} \tag{3.26}$$

Clearly, right hand side of the partial derivative is nothing but $\mathbb{X}-\mathbb{X}^t$. Same manipulations can be applied to $\mathbb{Y}$ and eventually they can be written as provided in (3.27).

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\mu}^t}\{\mathbb{X}\} &= D^t\mathbb{X} = \mathbb{X} - \mathbb{X}^t \\ \frac{\partial}{\partial \boldsymbol{\mu}^t}\{\mathbb{Y}\} &= D^t\mathbb{Y} = \mathbb{Y} - \mathbb{Y}^t\end{aligned} \tag{3.27}$$

Based on what is provided in latter equation, partial derivative of $\boldsymbol{M}_{12}$ is written in (3.28). This is used to linearise $\boldsymbol{M}_{12}$ and relying on the result provided in Lemma 4.1.2, optimization problem given in (3.24) is reformulated as in (3.30).

$$\begin{aligned} D^t\boldsymbol{M}_{12} \;=\; & \begin{bmatrix} \mathbb{X} \\ \mathbb{Y} \end{bmatrix}^T \boldsymbol{v} \begin{bmatrix} \boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \boldsymbol{v}^T \begin{bmatrix} \mathbb{X}^t \\ \mathbb{Y}^t \end{bmatrix} \\ +\; & \begin{bmatrix} \mathbb{X}^t \\ \mathbb{Y}^t \end{bmatrix}^T \boldsymbol{v} \begin{bmatrix} \boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \boldsymbol{v}^T \begin{bmatrix} \mathbb{X} \\ \mathbb{Y} \end{bmatrix} \\ -\; 2 & \begin{bmatrix} \mathbb{X}^t \\ \mathbb{Y}^t \end{bmatrix}^T \boldsymbol{v} \begin{bmatrix} \boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \boldsymbol{v}^T \begin{bmatrix} \mathbb{X}^t \\ \mathbb{Y}^t \end{bmatrix} \end{aligned} \tag{3.28}$$

Linearised $\boldsymbol{M}_{12}$ is denoted as $L\boldsymbol{M}_{12}$ and provided the following equation.

$$\boldsymbol{M}_{12} \;\geq\; L\boldsymbol{M}_{12} \;=\; \boldsymbol{M}_{12}^t + D^t\boldsymbol{M}_{12} \tag{3.29}$$

$$\begin{aligned} &\text{minimize} \quad \gamma \\ &\text{s. t.} \qquad \boldsymbol{\mathcal{X}} \;\geq\; \mathbf{0} \\ &\qquad\qquad \boldsymbol{M}_{11,2} - L\boldsymbol{M}_{12} \;\leq\; \mathbf{0} \end{aligned} \tag{3.30}$$

Second constraint of the problem defined in (3.30) is a quadratic matrix inequality which can be easily transformed to an LMI constraint using Shur Complement Lemma [125], as given in (3.31).

$$\boldsymbol{\Phi} = \left[\begin{array}{c|c} -\boldsymbol{I} & \begin{bmatrix} \mathbf{0} & \boldsymbol{I} \end{bmatrix} \boldsymbol{v}^T \begin{bmatrix} \mathbb{X} \\ \mathbb{Y} \end{bmatrix} \\ \hline \begin{bmatrix} \mathbb{X} \\ \mathbb{Y} \end{bmatrix}^T \boldsymbol{v} \begin{bmatrix} \mathbf{0} \\ \boldsymbol{I} \end{bmatrix} & \mathbb{Z} - L\boldsymbol{M}_{12} \end{array}\right] \leq \mathbf{0} \tag{3.31}$$

Eventually, optimization problem takes the final form as provided in (3.32).

$$\begin{aligned} &\text{minimize} && \gamma \\ &\text{s. t.} && \boldsymbol{\mathcal{X}} \geq \mathbf{0} \\ & && \boldsymbol{\Phi} \leq \mathbf{0} \end{aligned} \tag{3.32}$$

Thus far two optimization problems have been discussed. First one is aimed to find the initial strictly feasible solutions for a given initial adjacency matrix, which are denoted as $\boldsymbol{\mathcal{X}}^0$, γ^0 and $\mathbb{A}^0$, respectively and given in (3.25). The second one (see (3.32)) is desired to find solutions to decision variables $\boldsymbol{\mathcal{X}}$, $\mathbb{A}$ while minimizing γ starting from the initial values of $\mathcal{X}^0$, γ^0 and $\mathbb{A}^0$, which are passed from the first optimization problem. These two higher level optimization problems are formally provided in Algorithm 1.

Algorithm 1 Algorithm to solve problem given in Section 3.2.1

Optimization 1: Find initial feasible solution for given $\mathbb{A}^0$

minimize γ

for $\gamma_{error} > \epsilon$ **do**

Compute $\gamma_{try} = (\gamma_{up} - \gamma_{lw})/2$

Compute feasibility of problem given in (3.25) for γ_{try}

if (3.25) is feasible **then**

$\gamma_{up} = \gamma_{try}$

else

$\gamma_{lw} = \gamma_{try}$

end if

Compute γ_{error}

end for

End Optimization 1

Optimization 2: Start sLMI for given initial solutions to find optimal $\mathbb{A}^{nom}$

Define $\boldsymbol{\mathcal{X}}^t = \boldsymbol{\mathcal{X}}^0$, $\gamma^t = \gamma^0$, $\mathbb{A}^t = \mathbb{A}^0$

i = 1

for $\boldsymbol{\mu}_{err} > \epsilon_{\boldsymbol{\mu}}$ & $i < i^{max}$ **do**

if i=1 **then**

$\boldsymbol{\mathcal{X}}^i = \boldsymbol{\mathcal{X}}^t$, $\gamma^i = \gamma^t$, $\mathbb{A}^i = \mathbb{A}^t$

end if

Solve (3.32) using $\boldsymbol{\mathcal{X}}^i$, γ^i, $\mathbb{A}^i$ to obtain $\boldsymbol{\mathcal{X}}^{i+1}$, γ^{i+1}, $\mathbb{A}^{i+1}$

$\boldsymbol{\mu}^i = [vec(\mathbb{X}),\ vec(\mathbb{Y})]$, calculate $\boldsymbol{\mu}_{err} = |\boldsymbol{\mu}^i - \boldsymbol{\mu}^{i-1}|$

if $\boldsymbol{\mu}_{err} < \epsilon_{\boldsymbol{\mu}}$ **then**

break

end if

i = i + 1

end for

End Optimization 2

This section is dedicated to implement sLMI method in synthesizing $\mathbb{A}$ in terms of nominal H_∞ performance. Next section extends this method to include uncertainties in synthesis using IQCs.

3.2.2 Robust Adjacency Matrix Synthesis with IQCs

Including IQC analysis requires a modification in system defined in (6.22). Here an uncertainty channel is introduced to the system denoted as $\boldsymbol{S}_u$ that results in the state space representation given in (3.33). Here $\boldsymbol{z}_1 - \boldsymbol{w}_1$ and $\boldsymbol{z}_2 - \boldsymbol{w}_2$ channels are designated for uncertainty and performance channels, respectively. Inputs and outputs of the system are $\left[\boldsymbol{w}_1^T\ \boldsymbol{w}_2^T\ \boldsymbol{w}^T\right]^T$ and $\left[\boldsymbol{z}_1^T\ \boldsymbol{z}_2^T\ \boldsymbol{z}^T\right]^T$, where $\boldsymbol{w}_1 \in \mathbb{R}^{n_{w_1}}$, $\boldsymbol{w}_2 \in \mathbb{R}^{n_{w_2}}$, $\boldsymbol{w} \in \mathbb{R}^{n_w}$, $\boldsymbol{z}_1 \in \mathbb{R}^{nz_1}$, $\boldsymbol{z}_2 \in \mathbb{R}^{nz_2}$ and $\boldsymbol{z} \in \mathbb{R}^{n_z}$. State vector of the system has a dimension of $\boldsymbol{x} \in \mathbb{R}^{nx_s}$. $\boldsymbol{z}_1 - \boldsymbol{w}_1$ channel is included in the system to transfer spatial information of the MAS that is shared between the agents to uncertainty block. Therefore it is just a feed forward channel and matrices within is defined as $\boldsymbol{C}_1 = \boldsymbol{0} \in \mathbb{R}^{nz_1,\, nx_s}$, $\boldsymbol{D}_{11} = \boldsymbol{0} \in \mathbb{R}^{nz_1,\, nw_1}$, $\boldsymbol{D}_{12} = \boldsymbol{0} \in \mathbb{R}^{nz_1,\, nw_2}$ and $\boldsymbol{E}_1 = \boldsymbol{I} \in \mathbb{R}^{nz_1,\, nw}$. Graphical representation of the MAS is given in Figure 3.3.

$$\left[\begin{array}{c} \dot{\boldsymbol{x}} \\ \hline \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \\ \boldsymbol{z} \end{array}\right] = \left[\begin{array}{c|ccc} \boldsymbol{A} & \boldsymbol{B}_1 & \boldsymbol{B}_2 & \boldsymbol{B} \\ \hline \boldsymbol{C}_1 & \boldsymbol{D}_{11} & \boldsymbol{D}_{12} & \boldsymbol{E}_1 \\ \boldsymbol{C}_2 & \boldsymbol{D}_{21} & \boldsymbol{D}_{22} & \boldsymbol{E}_2 \\ \boldsymbol{C} & \boldsymbol{F}_1 & \boldsymbol{F}_2 & \boldsymbol{0} \end{array}\right] \left[\begin{array}{c} \boldsymbol{x} \\ \hline \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \\ \boldsymbol{w} \end{array}\right] \tag{3.33}$$

Given the uncertainty set $\boldsymbol{\Delta}$, satisfying the assumptions given in Section 3.1.5, multipliers within the uncertainty set $\boldsymbol{\Delta} = (\boldsymbol{\Pi}_k, k = 1, \cdots, M)$ can be factorized as $(\boldsymbol{\Psi}_k, \boldsymbol{P}_k)$, which are given in (3.34).

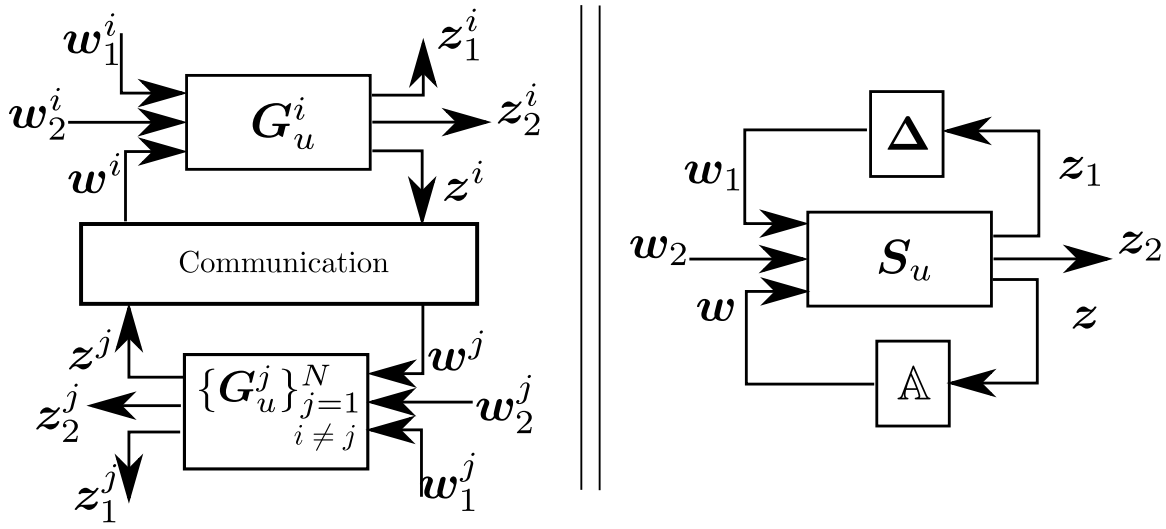


Figure 3.3. Interconnection of agents and lumped representation of MAS $\boldsymbol{H} = \mathcal{F}_U(\mathcal{F}_L(\boldsymbol{S}_u, \mathbb{A}), \boldsymbol{\Delta})$. .

$$
\begin{aligned}
\boldsymbol{\Psi}_k &= \left[\begin{array}{c|cc} \boldsymbol{A}_{\psi_k} & \boldsymbol{B}_{\psi_k,w_1} & \boldsymbol{B}_{\psi_k,z_1} \\ \hline \boldsymbol{C}_{\psi_k} & \boldsymbol{D}_{\psi_k,w_1} & \boldsymbol{D}_{\psi_k,z_1} \end{array}\right] \\
&= \left[\begin{array}{c|cc} \boldsymbol{A}_{\psi} & \boldsymbol{B}_{\psi_k,w_1} & \boldsymbol{B}_{\psi_k,z_1} \\ \hline \boldsymbol{C}_{\psi_k(1)} & \boldsymbol{D}_{\psi_k(1),w_1} & \boldsymbol{D}_{\psi_k(1),z_1} \\ \boldsymbol{C}_{\psi_k(2)} & \boldsymbol{D}_{\psi_k(2),w_1} & \boldsymbol{D}_{\psi_k(2),z_1} \end{array}\right] \\
\boldsymbol{P}_k &= \left[\begin{array}{c|c} \boldsymbol{I}_{(nw_1,nw_1)} & \boldsymbol{0} \\ \hline \boldsymbol{0} & -\boldsymbol{I}_{(nz_1,nz_1)} \end{array}\right]
\end{aligned}
\tag{3.34}
$$

System $\boldsymbol{S}_u$ given in (3.33) is extended using $\boldsymbol{\Psi}_k$ as defined in Figure 3.4.Extended state space definition for the group of vehicles is denoted as $\boldsymbol{S}_E$ and given in (3.35).

$$
\boldsymbol{S}_E \;:=\; \left[\begin{array}{cccc|ccc}
\boldsymbol{A} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{B}_1 & \boldsymbol{B}_2 & \boldsymbol{B} \\
\boldsymbol{0} & \boldsymbol{A}_{\psi_1} & \cdots & \boldsymbol{0} & \boldsymbol{B}_{\psi_1,w_1} & \boldsymbol{0} & \boldsymbol{B}_{\psi_1,z_1}\boldsymbol{E}_1 \\
\vdots & \boldsymbol{0} & \ddots & \boldsymbol{0} & \vdots & \vdots & \vdots \\
\boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{A}_{\psi_M} & \boldsymbol{B}_{\psi_M,w_1} & \boldsymbol{0} & \boldsymbol{B}_{\psi_M,z_1}\boldsymbol{E}_1 \\
\hline
\boldsymbol{0} & \boldsymbol{C}_{\psi_1} & \cdots & \boldsymbol{0} & \boldsymbol{D}_{\psi_1,w_1} & \boldsymbol{0} & \boldsymbol{D}_{\psi_1,z_1}\boldsymbol{E}_1 \\
\vdots & \boldsymbol{0} & \ddots & \boldsymbol{0} & \vdots & \vdots & \vdots \\
\boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{C}_{\psi_M} & \boldsymbol{D}_{\psi_M,w_1} & \boldsymbol{0} & \boldsymbol{D}_{\psi_M,z_1}\boldsymbol{E}_1 \\
\boldsymbol{C}_2 & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{D}_{21} & \boldsymbol{D}_{22} & \boldsymbol{E}_2 \\
\boldsymbol{C} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{F}_1 & \boldsymbol{F}_2 & \boldsymbol{0}
\end{array}\right] \tag{3.35}
$$

As clearly seen in (3.35), number of multipliers increase the size of composite state and output vectors such that composite state vector is $\boldsymbol{\zeta}^T := \left[\boldsymbol{x}^T,\ \boldsymbol{x}_{\psi,1}^T,\ \cdots,\ \boldsymbol{x}_{\psi,M}^T\right]^T \in \mathbb{R}^{nx+nx_{psi,1}+\cdots+nx_{psi,M}}$ and $\boldsymbol{z}_e^T := \left[\boldsymbol{z}_2^T,\ \boldsymbol{z}_{\psi,1}^T,\ \cdots,\ \boldsymbol{z}_{\psi,M}^T\right]^T \in \mathbb{R}^{nz_2+nz_{psi,1}+\cdots+nz_{psi,M}}$. To simplify this representation, $\boldsymbol{S}_E$ is rewritten as provided in (3.36), where $\boldsymbol{A}_{\psi_K} := diag\{\boldsymbol{A}_{\psi_k}\}_{k=1}^M$, $\boldsymbol{C}_{\psi_K} := diag\{\boldsymbol{C}_{\psi_k}\}_{k=1}^M$, $\boldsymbol{B}_{\psi_K,w_1} := col\{\boldsymbol{B}_{\psi_k,w_1}\}_{k=1}^M$, $\boldsymbol{D}_{\psi_K,w_1} := col\{\boldsymbol{D}_{\psi_k,w_1}\}_{k=1}^M$, $\boldsymbol{B}_{\psi_K,z_1}\boldsymbol{E}_1 := col\{\boldsymbol{B}_{\psi_k,z_1}\boldsymbol{E}_1\}_{k=1}^M$ and $\boldsymbol{D}_{\psi_K,z_1}\boldsymbol{E}_1 := col\{\boldsymbol{D}_{\psi_k,z_1}\boldsymbol{E}_1\}_{k=1}^M$. Likewise, $\boldsymbol{B}_{\psi_K,z_1}\boldsymbol{E}_1\mathbb{A}\boldsymbol{C} := col\{\boldsymbol{B}_{\psi_k,z_1}\boldsymbol{E}_1\mathbb{A}\boldsymbol{C}\}_{k=1}^M$ and $\boldsymbol{D}_{\psi_K,z_1}\boldsymbol{E}_1\mathbb{A}\boldsymbol{C} := col\{\boldsymbol{D}_{\psi_k,z_1}\boldsymbol{E}_1\mathbb{A}\boldsymbol{C}\}_{k=1}^M$.

$$\boldsymbol{S}_E := \left[\begin{array}{cc|ccc} \boldsymbol{A} & \boldsymbol{0} & \boldsymbol{B}_1 & \boldsymbol{B}_2 & \boldsymbol{B} \\ \boldsymbol{0} & \boldsymbol{A}_{\psi_K} & \boldsymbol{B}_{\psi_K,w_1} & \boldsymbol{0} & \boldsymbol{B}_{\psi_K,z_1}\boldsymbol{E}_1 \\ \hline \boldsymbol{0} & \boldsymbol{C}_{\psi_K} & \boldsymbol{D}_{\psi_K,w_1} & \boldsymbol{0} & \boldsymbol{D}_{\psi_K,z_1}\boldsymbol{E}_1 \\ \boldsymbol{C}_2 & \boldsymbol{0} & \boldsymbol{D}_{21} & \boldsymbol{D}_{22} & \boldsymbol{E}_2 \\ \boldsymbol{C} & \boldsymbol{0} & \boldsymbol{F}_1 & \boldsymbol{F}_2 & \boldsymbol{0} \end{array}\right] \tag{3.36}$$

MAS can then be constructed by $\boldsymbol{w} = \mathbb{A}\boldsymbol{z}$. This creates the system $\boldsymbol{H}_{E,K}$, which is provided in (3.37).

$$\boldsymbol{H}_{E,K} := \left[\begin{array}{c|cc} \mathcal{A}_K & \mathcal{B}_{K,1} & \mathcal{B}_{K,2} \\ \hline \mathcal{C}_{K_\psi} & \mathcal{D}_{K_{\psi,1}} & \mathcal{D}_{K_{\psi,2}} \\ \mathcal{C}_K & \mathcal{D}_{K,1} & \mathcal{D}_{K,2} \end{array}\right] \tag{3.37}$$

$$\begin{aligned} \mathcal{A}_K &= \begin{bmatrix} \boldsymbol{A} + \boldsymbol{B}\mathbb{A}\boldsymbol{C} & \boldsymbol{0} \\ \boldsymbol{B}_{\psi_K,z_1}\boldsymbol{E}_1\mathbb{A}\boldsymbol{C} & \boldsymbol{A}_{\psi_K} \end{bmatrix} \\ \mathcal{B}_{K,1} &= \begin{bmatrix} \boldsymbol{B}_1 + \boldsymbol{B}\mathbb{A}\boldsymbol{F}_1 \\ \boldsymbol{B}_{\psi_K,w_1} + \boldsymbol{B}_{\psi_K,z_1}\boldsymbol{E}_1\mathbb{A}\boldsymbol{F}_1 \end{bmatrix} \\ \mathcal{B}_{K,2} &= \begin{bmatrix} \boldsymbol{B}_2 + \boldsymbol{B}\mathbb{A}\boldsymbol{F}_2 \\ \boldsymbol{B}_{\psi_K,z_1}\boldsymbol{E}_1\mathbb{A}\boldsymbol{F}_2 \end{bmatrix} \\ \begin{bmatrix} \mathcal{C}_{K_\psi} \\ \mathcal{C}_K \end{bmatrix} &= \begin{bmatrix} \boldsymbol{D}_{\psi_K,z_1}\boldsymbol{E}_1\mathbb{A}\boldsymbol{C} & \boldsymbol{C}_{\psi_K} \\ \boldsymbol{C}_2 + \boldsymbol{E}_2\mathbb{A}\boldsymbol{C} & \boldsymbol{0} \end{bmatrix} \\ \begin{bmatrix} \mathcal{D}_{K_{\psi,1}} \\ \mathcal{D}_{K,1} \end{bmatrix} &= \begin{bmatrix} \boldsymbol{B}_{\psi_K,w_1} + \boldsymbol{B}_{\psi_K,z_1}\boldsymbol{E}_1\mathbb{A}\boldsymbol{F}_1 \\ \boldsymbol{D}_{21} + \boldsymbol{E}_2\mathbb{A}\boldsymbol{F}_1 \end{bmatrix} \\ \begin{bmatrix} \mathcal{D}_{K_{\psi,2}} \\ \mathcal{D}_{K,2} \end{bmatrix} &= \begin{bmatrix} \boldsymbol{B}_{\psi_K,z_1}\boldsymbol{E}_1\mathbb{A}\boldsymbol{F}_2 \\ \boldsymbol{D}_{22} + \boldsymbol{E}_2\mathbb{A}\boldsymbol{F}_2 \end{bmatrix} \end{aligned} \tag{3.38}$$

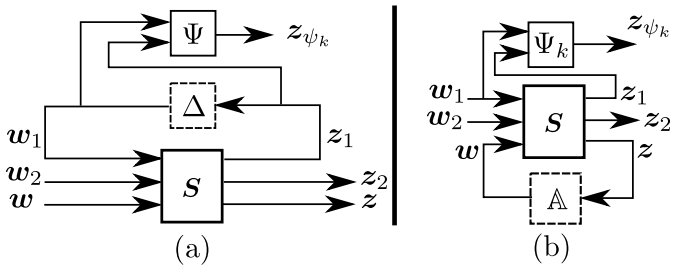


Figure 3.4. a) Extended $\boldsymbol{S}_E$ to include filter $\boldsymbol{\Psi}$. b) Extended cooperative system is created with $\mathbb{A}$. .

Robust stability of $\boldsymbol{H}_{E,K}$ is certificated by robust performance of the system, $\boldsymbol{H}_{E,\lambda}$, that originates from $\boldsymbol{S}$, which is extended with a combined multiplier $\boldsymbol{\Pi}_\lambda$. This multiplier is defined as $\boldsymbol{\Pi}_\lambda := \sum_{k=1}^{M} \lambda_k \boldsymbol{\Pi}_k$ and can be factorized using the factorization given in work [44], which yields $(\boldsymbol{\Psi}_\lambda, \boldsymbol{P}_\lambda)$ as given in (3.39).

$$
\begin{aligned}
\boldsymbol{\Psi}_\lambda &= \left[\begin{array}{c|cc} \boldsymbol{A}_{\psi_\lambda} & \boldsymbol{B}_{\psi_\lambda,w_1} & \boldsymbol{B}_{\psi_\lambda,z_1} \\ \hline \boldsymbol{C}_{\psi_\lambda} & \boldsymbol{D}_{\psi_\lambda,w_1} & \boldsymbol{D}_{\psi_\lambda,z_1} \end{array}\right] \\
&= \left[\begin{array}{c|cc} \boldsymbol{A}_{\psi} & \boldsymbol{B}_{\psi_\lambda,w_1} & \boldsymbol{B}_{\psi_\lambda,z_1} \\ \hline \boldsymbol{C}_{\psi_\lambda(1)} & \boldsymbol{D}_{\psi_\lambda(1),w_1} & \boldsymbol{D}_{\psi_\lambda(1),z_1} \\ \boldsymbol{C}_{\psi_\lambda(2)} & \boldsymbol{D}_{\psi_\lambda(2),w_1} & \boldsymbol{D}_{\psi_\lambda(2),z_1} \end{array}\right] \\
\boldsymbol{P}_\lambda &= \left[\begin{array}{c|c} \boldsymbol{I}_{(nw_1,nw_1)} & \boldsymbol{0} \\ \hline \boldsymbol{0} & -\boldsymbol{I}_{(nz_1,nz_1)} \end{array}\right]
\end{aligned}
\tag{3.39}
$$

Specifically, $\boldsymbol{H}_{E,\lambda}$ is obtained by extending $\boldsymbol{S}$ with combined multiplier $\boldsymbol{\Pi}_\lambda$ and connecting agents with each other through $\mathbb{A}$ as in Figure 3.4. This yields the state space definition for $\boldsymbol{H}_{E,\lambda}$as given in (3.40).

$$
\begin{aligned}
\boldsymbol{H}_{E,\lambda} &:= \left[\begin{array}{c|cc} \mathcal{A}_\lambda & \mathcal{B}_{\lambda,1} & \mathcal{B}_{\lambda,2} \\ \hline \mathcal{C}_{\lambda_\psi} & \mathcal{D}_{\lambda_{\psi,1}} & \mathcal{D}_{\lambda_{\psi,2}} \\ \mathcal{C}_\lambda & \mathcal{D}_{\lambda,1} & \mathcal{D}_{\lambda,2} \end{array}\right] \\
&= \left[\begin{array}{c|cc} \mathcal{A}_\lambda & \mathcal{B}_{\lambda,1} & \mathcal{B}_{\lambda,2} \\ \hline \mathcal{C}_{\lambda_\psi}(1) & \mathcal{D}_{\lambda_{\psi,1}}(1) & \mathcal{D}_{\lambda_{\psi,2}}(1) \\ \mathcal{C}_{\lambda_\psi}(2) & \mathcal{D}_{\lambda_{\psi,1}}(2) & \mathcal{D}_{\lambda_{\psi,2}}(2) \\ \mathcal{C}_\lambda & \mathcal{D}_{\lambda,1} & \mathcal{D}_{\lambda,2} \end{array}\right]
\end{aligned} \tag{3.40}
$$

$$
\begin{aligned}
\mathcal{A}_\lambda &= \left[\begin{array}{cc} \boldsymbol{A} + \boldsymbol{B}\mathbb{A}\boldsymbol{C} & \boldsymbol{0} \\ \boldsymbol{B}_{\psi_\lambda,z_1}\boldsymbol{E}_1\mathbb{A}\boldsymbol{C} & \boldsymbol{A}_{\psi_\lambda} \end{array}\right] \\
\mathcal{B}_{\lambda,1} &= \left[\begin{array}{c} \boldsymbol{B}_1 + \boldsymbol{B}\mathbb{A}\boldsymbol{F}_1 \\ \boldsymbol{B}_{\psi_\lambda,w_1} + \boldsymbol{B}_{\psi_\lambda,z_1}\boldsymbol{E}_1\mathbb{A}\boldsymbol{F}_1 \end{array}\right] \\
\mathcal{B}_{\lambda,2} &= \left[\begin{array}{c} \boldsymbol{B}_2\boldsymbol{B}\mathbb{A}\boldsymbol{F}_2 \\ \boldsymbol{B}_{\psi_\lambda,z_1}\boldsymbol{E}_1\mathbb{A}\boldsymbol{F}_2 \end{array}\right] \\
\left[\begin{array}{c} \mathcal{C}_{\lambda_\psi} \\ \mathcal{C}_K \end{array}\right] &= \left[\begin{array}{cc} \boldsymbol{D}_{\psi_\lambda,z_1}\boldsymbol{E}_1\mathbb{A}\boldsymbol{C} & \boldsymbol{C}_{\psi_\lambda} \\ \boldsymbol{C}_2\boldsymbol{E}_2\mathbb{A}\boldsymbol{C} & \boldsymbol{0} \end{array}\right] \\
\left[\begin{array}{c} \mathcal{D}_{\lambda_{\psi,1}} \\ \mathcal{D}_{\lambda,1} \end{array}\right] &= \left[\begin{array}{c} \boldsymbol{B}_{\psi_\lambda,w_1} + \boldsymbol{B}_{\psi_\lambda,z_1}\boldsymbol{E}_1\mathbb{A}\boldsymbol{F}_1 \\ \boldsymbol{D}_{21} + \boldsymbol{E}_2\mathbb{A}\boldsymbol{F}_1 \end{array}\right] \\
\left[\begin{array}{c} \mathcal{D}_{\lambda_{\psi,2}} \\ \mathcal{D}_{\lambda,2} \end{array}\right] &= \left[\begin{array}{c} \boldsymbol{B}_{\psi_\lambda,z_1}\boldsymbol{E}_1\mathbb{A}\boldsymbol{F}_2 \\ \boldsymbol{D}_{22} + \boldsymbol{E}_2\mathbb{A}\boldsymbol{F}_2 \end{array}\right]
\end{aligned} \tag{3.41}
$$

Theorem 3.2.1 *[65] Assuming upper LFT of $(\boldsymbol{S}, \boldsymbol{\Delta})$ is well posed and $\{\lambda_k\}_{k=1}^M$ satisfies the assumptions given in Section 3.1.5. Let $\gamma > 0$ and $\lambda_1 \in \mathbb{R}_{++}$ and $\lambda_{k>1} \in \mathbb{R}_+$. Let $\boldsymbol{\Pi}_k$ has a factorization $(\boldsymbol{\Psi}_k, \boldsymbol{P}_k)$, where $\boldsymbol{\Psi}_k$ is stable and $\boldsymbol{\Pi}_\lambda := \sum_{k=1}^M \lambda_k \boldsymbol{\Pi}_k$ has a J-Spectral*

factorization as $(\boldsymbol{\Psi}_\lambda, \boldsymbol{P}_\lambda)$ *based on the results given in work [44]. Let* $\boldsymbol{r}_k(\star_{K_\psi})$ *denote regarding rows of* $\star_{K_\psi}$*. Then following are equivalent.*

1. $\boldsymbol{H}_{E,K}$ *achieves robust performance of* γ *if the matrix inequality in (3.42) holds true for* $\mathcal{X}_K^T = \mathcal{X}_K \in \mathbb{R}^{n_S+n_\psi}$*, scalar* $\{\lambda_k\}_{k=1}^M$ *and* γ*.*

$$
\begin{aligned}
\mathcal{X}_K \;&\geq\; \mathbf{0} \\
&\begin{bmatrix} \mathcal{X}_K \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathcal{A}_K & \mathcal{B}_{K,1} & \mathcal{B}_{K,2} \end{bmatrix} + \\
&\begin{bmatrix} \mathcal{A}_K^T \\ \mathcal{B}_{K,1}^T \\ \mathcal{B}_{K,2}^T \end{bmatrix} \begin{bmatrix} \mathcal{X}_K & \mathbf{0} & \mathbf{0} \end{bmatrix} + \\
&\begin{bmatrix} \mathcal{C}_K^T & \mathbf{0} \\ \mathcal{D}_{K,1}^T & \mathbf{0} \\ \mathcal{D}_{K,2}^T & \gamma \boldsymbol{I} \end{bmatrix} \boldsymbol{P}_p \begin{bmatrix} \mathcal{C}_K & \mathcal{D}_{K,1} & \mathcal{D}_{K,2} \\ \mathbf{0} & \mathbf{0} & \gamma \boldsymbol{I} \end{bmatrix} + \\
\sum_{k=1}^{M} \lambda_k &\begin{bmatrix} \boldsymbol{r}_k(\mathcal{C}_{K_\psi})^T \\ \boldsymbol{r}_k(\mathcal{D}_{K_{\psi,1}})^T \\ \boldsymbol{r}_k(\mathcal{D}_{K_{\psi,2}})^T \end{bmatrix} \boldsymbol{P}_k \begin{bmatrix} \boldsymbol{r}_k(\mathcal{C}_{K_\psi})^T \\ \boldsymbol{r}_k(\mathcal{D}_{K_{\psi,1}})^T \\ \boldsymbol{r}_k(\mathcal{D}_{K_{\psi,2}})^T \end{bmatrix}^T \;\leq\; \mathbf{0}
\end{aligned}
\tag{3.42}
$$

2. $\boldsymbol{H}_{E,\lambda}$ *achieves robust performance of* γ *if the matrix inequality in (3.43) holds true for* $\mathcal{X}_\lambda^T = \mathcal{X}_\lambda \in \mathbb{R}^{nx_S+nx_{\psi,\lambda}}$ *and* γ*, where* $\boldsymbol{P}_p = diag(\boldsymbol{I}, -\boldsymbol{I}) \in \mathbb{R}^{(nz_2+nw_2, nz_2+nw_2)}$*.*

$$
\begin{aligned}
\mathcal{X}_\lambda \;&\geq\; \mathbf{0} \\
\begin{bmatrix} \mathcal{X}_\lambda \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} &\begin{bmatrix} \mathcal{A}_\lambda & \mathcal{B}_{\lambda,1} & \mathcal{B}_{\lambda,2} \end{bmatrix} + \\
\begin{bmatrix} \mathcal{A}_\lambda^T \\ \mathcal{B}_{\lambda,1}^T \\ \mathcal{B}_{\lambda,2}^T \end{bmatrix} &\begin{bmatrix} \mathcal{X}_\lambda & \mathbf{0} & \mathbf{0} \end{bmatrix} + \\
\begin{bmatrix} \mathcal{C}_\lambda^T & \mathbf{0} \\ \mathcal{D}_{\lambda,1}^T & \mathbf{0} \\ \mathcal{D}_{\lambda,2}^T & \gamma \boldsymbol{I} \end{bmatrix} \boldsymbol{P}_p &\begin{bmatrix} \mathcal{C}_\lambda & \mathcal{D}_{\lambda,1} & \mathcal{D}_{\lambda,2} \\ \mathbf{0} & \mathbf{0} & \gamma \boldsymbol{I} \end{bmatrix} + \\
\begin{bmatrix} \mathcal{C}_{\lambda_\psi}^T \\ \mathcal{D}_{\lambda_{\psi,1}}^T \\ \mathcal{D}_{\lambda_{\psi,2}}^T \end{bmatrix} \boldsymbol{P}_\lambda &\begin{bmatrix} \mathcal{C}_{\lambda_\psi}^T \\ \mathcal{D}_{\lambda_{\psi,1}}^T \\ \mathcal{D}_{\lambda_{\psi,2}}^T \end{bmatrix}^T \;\leq\; \mathbf{0}
\end{aligned}
\tag{3.43}
$$

Proof 3.2.2 (Proof of the Theorem 3.2.1) *Proof is provided in work [65].*

Theorem 3.2.1 ensures that $\boldsymbol{H}_{E,K}$ is a dissipative system and has a valid storage function $V(\boldsymbol{x}(t)) = \boldsymbol{x}^T(t)\boldsymbol{\mathcal{X}}_K\boldsymbol{x}(t)$. Ensuring this allows the proceeding analysis to obtain a BMI inequality. By investigating (3.37), one can see that $\boldsymbol{H}_{E,k}$ is affine function of $\mathbb{A}$ and inequality given in (3.42) contains bilinear terms. To investigate this inequality, matrix variables are renamed as in (3.44). Here scalar $\{\lambda_k\}_{k=1}^M \geq 0$, therefore a new variable $\mu_k = \sqrt{\lambda_k}$ can be defined without loss of generality.

$$
\begin{aligned}
\mathbb{X}_K(\boldsymbol{\mathcal{X}}_K) &= \begin{bmatrix} \boldsymbol{\mathcal{X}}_K & \mathbf{0} & \mathbf{0} \end{bmatrix} \\
\mathbb{Y}_K(\mathbb{A}) &= \begin{bmatrix} \mathcal{A}_K & \mathcal{B}_{K,1} & \mathcal{B}_{K,2} \end{bmatrix} \\
\mathbb{R}_K(\mathbb{A}, \lambda_k) &= \mu_k \begin{bmatrix} \boldsymbol{C}_{K_\psi}(1) & \boldsymbol{D}_{K_\psi, z_1}(1) & \boldsymbol{D}_{K_\psi, w_1}(1) \end{bmatrix} \\
\mathbb{Q}_K(\gamma) &= \begin{bmatrix} \mathbf{0} & \mathbf{0} & \gamma \boldsymbol{I} \end{bmatrix} \\
\mathbb{P}_K(\mathbb{A}) &= \begin{bmatrix} \mathcal{C}_K & \mathcal{D}_{K,1} & \mathcal{D}_{K,2} \end{bmatrix} \\
\mathbb{L}_K(\mathbb{A}, \lambda_k) &= \mu_k \begin{bmatrix} \boldsymbol{C}_{K_\psi}(2) & \boldsymbol{D}_{K_\psi, z_1}(2) & \boldsymbol{D}_{K_\psi, w_1}(2) \end{bmatrix}
\end{aligned}
\tag{3.44}
$$

Similar to Section 3.2.1, second inequality in (3.42) can be rewritten as in (3.46). This is done by representing correlation matrices $\boldsymbol{P}_p$ and $\boldsymbol{P}_K$ with an inverted eigenvalue decomposition of an indefinite matrix as described in (3.45). Here $\boldsymbol{w}$ is the left eigenvector with the respective size. Specifically, eigenvectors related to $\boldsymbol{P}_p$ and $\boldsymbol{P}_K$ can be denoted as $\boldsymbol{w}_p$ and $\boldsymbol{w}_K$.

$$
\begin{aligned}
\begin{bmatrix} \mathbf{0} & \boldsymbol{I} \\ \boldsymbol{I} & \mathbf{0} \end{bmatrix} &= \boldsymbol{w} \begin{bmatrix} \boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & -\boldsymbol{I} \end{bmatrix} \boldsymbol{w}^T \\
\begin{bmatrix} \boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & -\boldsymbol{I} \end{bmatrix} &= \boldsymbol{w}^T \begin{bmatrix} \mathbf{0} & \boldsymbol{I} \\ \boldsymbol{I} & \mathbf{0} \end{bmatrix} \boldsymbol{w}
\end{aligned}
\tag{3.45}
$$

Redefining correlation matrices results in an inequality (3.46).

$$\begin{bmatrix} \mathbb{X}_K \\ \mathbb{Y}_K \end{bmatrix}^T \left[\begin{array}{c|c} \mathbf{0} & \boldsymbol{I} \\ \hline \boldsymbol{I} & \mathbf{0} \end{array}\right] \begin{bmatrix} \mathbb{X}_K \\ \mathbb{Y}_K \end{bmatrix} + \begin{bmatrix} \mathbb{P}_K \\ \mathbb{Q}_K \end{bmatrix}^T \boldsymbol{w}_p^T \left[\begin{array}{c|c} \mathbf{0} & \boldsymbol{I} \\ \hline \boldsymbol{I} & \mathbf{0} \end{array}\right] \boldsymbol{w}_p \begin{bmatrix} \mathbb{P}_K \\ \mathbb{Q}_K \end{bmatrix} + \begin{bmatrix} \mathbb{R}_K \\ \mathbb{L}_K \end{bmatrix}^T \boldsymbol{w}_K^T \left[\begin{array}{c|c} \mathbf{0} & \boldsymbol{I} \\ \hline \boldsymbol{I} & \mathbf{0} \end{array}\right] \boldsymbol{w}_K \begin{bmatrix} \mathbb{R}_K \\ \mathbb{L}_K \end{bmatrix} \leq \mathbf{0} \tag{3.46}$$

This inequality can further be simplified to the inequality in (3.48) using the matrix variables defined as in (3.47). Finally, matrix variables are collected in a single matrix variable called as $\mathbb{H}$ and leads to a concise representation of the inequality as provided in (3.51). Partitioning of the variable $\mathbb{H}$ is provided in (3.50).

$$\begin{bmatrix} \bar{\mathbb{P}}_\lambda \\ \bar{\mathbb{Q}}_K \end{bmatrix} = \boldsymbol{w}_p \begin{bmatrix} \mathbb{P}_K \\ \mathbb{Q}_K \end{bmatrix} \quad \begin{bmatrix} \bar{\mathbb{R}}_K \\ \bar{\mathbb{L}}_K \end{bmatrix} = \boldsymbol{w}_K \begin{bmatrix} \mathbb{R}_K \\ \mathbb{L}_K \end{bmatrix} \tag{3.47}$$

$$\begin{bmatrix} \mathbb{X}_K \\ \mathbb{Y}_K \end{bmatrix}^T \left[\begin{array}{c|c} \mathbf{0} & \boldsymbol{I} \\ \hline \boldsymbol{I} & \mathbf{0} \end{array}\right] \begin{bmatrix} \mathbb{X}_K \\ \mathbb{Y}_K \end{bmatrix} + \begin{bmatrix} \bar{\mathbb{P}}_K \\ \bar{\mathbb{Q}}_K \end{bmatrix}^T \left[\begin{array}{c|c} \mathbf{0} & \boldsymbol{I} \\ \hline \boldsymbol{I} & \mathbf{0} \end{array}\right] \begin{bmatrix} \bar{\mathbb{P}}_K \\ \bar{\mathbb{Q}}_K \end{bmatrix} + \begin{bmatrix} \bar{\mathbb{R}}_K \\ \bar{\mathbb{L}}_K \end{bmatrix}^T \left[\begin{array}{c|c} \mathbf{0} & \boldsymbol{I} \\ \hline \boldsymbol{I} & \mathbf{0} \end{array}\right] \begin{bmatrix} \bar{\mathbb{R}}_K \\ \bar{\mathbb{L}}_K \end{bmatrix} \leq \mathbf{0} \tag{3.48}$$

$$\left[\begin{array}{c} \bar{\mathbb{R}}_K \\ \bar{\mathbb{P}}_K \\ \mathbb{X}_K \\ \hline \mathbb{Y}_K \\ \bar{\mathbb{Q}}_K \\ \bar{\mathbb{L}}_K \end{array}\right]^T \left[\begin{array}{ccc|ccc} \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} \\ \hline \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right] \left[\begin{array}{c} \mathbb{X}_K \\ \bar{\mathbb{P}}_K \\ \bar{\mathbb{R}}_K \\ \hline \mathbb{Y}_K \\ \bar{\mathbb{Q}}_K \\ \bar{\mathbb{L}}_K \end{array}\right] \leq \mathbf{0} \tag{3.49}$$

$$\begin{aligned} \mathbb{H}^T &:= \left[\begin{array}{c|c} \mathbb{H}_1^T & \mathbb{H}_2^T \end{array}\right]^T \\ &= \left[\begin{array}{ccc|ccc} \mathbb{X}_K^T & \bar{\mathbb{P}}_K^T & \bar{\mathbb{R}}_K^T & \mathbb{Y}_K^T & \bar{\mathbb{Q}}_K^T & \bar{\mathbb{L}}_K^T \end{array}\right]^T \end{aligned} \tag{3.50}$$

$$\left[\begin{array}{c} \mathbb{H}_1 \\ \hline \mathbb{H}_2 \end{array}\right]^T \mathbb{Q}_{\mathbb{H}} \left[\begin{array}{c} \mathbb{H}_1 \\ \hline \mathbb{H}_2 \end{array}\right] \leq \mathbf{0} \tag{3.51}$$

At this stage, the matrix inequality given in (3.51) still contains nonlinear terms that cannot be cast into bilinear problem. This is due to the fact that we have $\mathbb{R}_K(\mathbb{A}, \lambda_k)$ and $\mathbb{L}_K(\mathbb{A}, \lambda_k)$ in the formulation. The solution to this nonlinearity is provided in the following context and for now λ_k is assumed to be a constant. This assumption allows us to continue on convex-concave decomposition of the inequality (3.51) by following the method applied on the quadratic definition in (3.21) given in Section 3.2.1. Simply starting with eigenvalue decomposition of $\mathbb{Q}_{\mathbb{H}}$ as given in (3.52) a convex and a concave partition for the inequality is obtained. Then following the formulation given between (3.23) and (3.29) for (3.51), linear inequality given in (3.53) can be calculated. Linearised concave partition of the inequality is denoted as $L\mathbb{M}_{12}$.

$$\mathbb{Q}_{\mathbb{H}} = \boldsymbol{v}_{\mathbb{H}} \left[\begin{array}{cc} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} \end{array}\right] \boldsymbol{v}_{\mathbb{H}}^T - \boldsymbol{v}_{\mathbb{H}} \left[\begin{array}{cc} \boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{array}\right] \boldsymbol{v}_{\mathbb{H}}^T \tag{3.52}$$

$$\boldsymbol{\Gamma} = \left[\begin{array}{c|c} -\boldsymbol{I} & \begin{bmatrix} \mathbf{0} & \boldsymbol{I} \end{bmatrix} \boldsymbol{v}_{\mathbb{H}}^{T}\mathbb{H} \\ \hline \mathbb{H}^{T}\boldsymbol{v}_{\mathbb{H}} \begin{bmatrix} \mathbf{0} \\ \boldsymbol{I} \end{bmatrix} & -L\mathbb{M}_{12} - s\boldsymbol{I} \end{array}\right] \leq \mathbf{0} \tag{3.53}$$

Eventually, optimization problem takes the final form as provided in (3.54).

$$\begin{aligned} &\text{minimize} \quad \gamma \\ &\text{s. t.} \qquad \boldsymbol{\mathcal{X}}_K \quad \geq \mathbf{0} \\ &\qquad\qquad \boldsymbol{\Gamma} \quad \leq \mathbf{0} \end{aligned} \tag{3.54}$$

Once optimization problem given in (3.54) is set, a sequence of optimization has to be solved to reach solution for robust adjacency matrix $\mathbb{A}_{robust}$. This sequence is provided in Algorithm 2 and convergence of the solution using these algorithms can be guaranteed based on the Theorem 3.2.6. Let $\mathbb{D} := \{\mu \in \Omega | \boldsymbol{G}_i(\mu) - \boldsymbol{H}_\mu \leq \mathbf{0},\ i = 1, \cdots, l\}$ be the feasible set of problems defined in (3.32) and (3.54). Let $ri(\mathbb{D}) := \{\mu \in ri(\Omega) | \boldsymbol{G}_i(\mu) - \boldsymbol{H}_\mu \leq \mathbf{0},\ i = 1, \cdots, m\}$ be the classical interiors.

Assumption 3.2.3 *$ri(\mathbb{D})$ is non-empty.*

Assumption 3.2.4 *Let mapping $\boldsymbol{G}_i(\mu)$ be Schur PSD convex and objective function $f(\mu)$ be convex quadratic on $\mathbb{R}$.*

Assumption 3.2.5 *Suppose convex subproblems (3.32) and (3.54) be solvable and satisfy second order sufficient condition.*

Theorem 3.2.6 ([122]) *Objective function $f(\mu)$ is bounded from below on $\mathbb{D}$ if Assumptions 3.2.3, 3.2.4 and 3.2.5 holds.*

Proof 3.2.7 (Proof of the Theorem 3.2.6) *Suppose the LMI problem in (3.25) and (3.42) is equivalent to*

$$\begin{aligned} &\textit{minimize} \quad \gamma \\ &\textit{s. t.} \qquad \boldsymbol{C}(\boldsymbol{\mu}) - s\boldsymbol{I} \ \leq \ \epsilon\boldsymbol{I} \end{aligned} \tag{3.55}$$

, where $\epsilon > 0$ and s is the slack variable. In addition, they are the original problem of (3.32) and (3.54). For nonlinear semidefinite problems of (3.32) and (3.54), initial conditions are obtained from (3.25) and (3.42), respectively, which makes (3.32) and (3.54) strictly feasible. Then Assumption 3.2.3 is satisfied. Second inequalities of (3.32) and (3.54) problems are Schur PSD convex due to Lemma 3.1.5. Lemma 4.1.2 denotes the lower bound of a PSD convex mapping and this is used on second inequalities of (3.32) and (3.54). In addition $f(\mu) := \gamma$ is convex, therefore

Assumption 3.2.4 is satisfied. Finally, problems (3.32) and (3.54) are finite series of LMI problems and due to Assumption 3.2.3 all of them are strictly feasible. This covers the complete proof of the theorem for this problem.

Algorithm 2 Algorithm to solve problem given in Section 3.2.2

Optimization 1: Find initial feasible solution for given $\mathbb{A}^0$ **and for** $\boldsymbol{\lambda}_K = [1,\ 0,\ \cdots,\ 0]$

minimize γ

for $\gamma_{error} > \epsilon$ **do**

    Compute $\gamma_{try} = (\gamma_{up} - \gamma_{lw})/2$

    Compute feasibility of problem given in (3.42) for γ_{try}

    if (3.42) is feasible **then**

        $\gamma_{up} = \gamma_{try}$

    else

        $\gamma_{lw} = \gamma_{try}$

    end if

    Compute γ_{error}

end for

End Optimization 1

Optimization 2: Start sLMI for given initial solutions to find optimal $\mathbb{A}_r$

Define $\boldsymbol{\mathcal{X}}^t_K = \boldsymbol{\mathcal{X}}^0_K$, $\gamma^t = \gamma^0$, $\mathbb{A}^t = \mathbb{A}^0$

i = 1

for $\boldsymbol{\mu}_{err} > \epsilon_{\boldsymbol{\mu}}$ & $i < i^{max}$ **do**

    if i=1 **then**

        $\boldsymbol{\mathcal{X}}^i = \boldsymbol{\mathcal{X}}^t_K$, $\gamma^i = \gamma^t$, $\mathbb{A}^i = \mathbb{A}^t$

    end if

    Solve (3.54) using $\boldsymbol{\mathcal{X}}^i$, γ^i, $\mathbb{A}^i$ to obtain $\boldsymbol{\mathcal{X}}^{i+1}$, γ^{i+1}, $\mathbb{A}^{i+1}$

    $\boldsymbol{\mu}^i = [vec(\mathbb{H})]$, calculate $\boldsymbol{\mu}_{err} = |\boldsymbol{\mu}^i - \boldsymbol{\mu}^{i-1}|$

    if $\boldsymbol{\mu}_{err} < \epsilon_{\boldsymbol{\mu}}$ **then**

        break

    end if

    i = i + 1

end for

End Optimization 2

Algorithm 2 continued

Optimization 1: Find initial feasible solution for given $\mathbb{A}^0$ **and for** $\boldsymbol{\lambda}_K = [1,\ 0,\ \cdots,\ 0]$

End Optimization 1

Optimization 2: Start sLMI for given initial solutions to find optimal $\mathbb{A}_r$

End Optimization 2

Optimization 3: Start LMI for given $\mathbb{A}_r$. Optimize $\boldsymbol{\lambda}_K$ to minimize γ.

minimize γ

for $\gamma_{error} > \epsilon$ **do**

    Compute $\gamma_{try} = (\gamma_{up} - \gamma_{lw})/2$

    Compute feasibility of problem given in (3.42) for γ_{try}

    if (3.42) is feasible **then**

        $\gamma_{up} = \gamma_{try}$

    else

        $\gamma_{lw} = \gamma_{try}$

    end if

    Compute γ_{error}

end for

End Optimization 3

Before moving to the numerical verifications, It should be noted that nominal and robust synthesis procedures are similar. **Step 3** of the Algorithm 2 is required for robust synthesis due to the fact that λ cannot be optimized along with the adjacency matrix $\mathbb{A}$ and decision variable $\mathcal{X}$. Finally, difference between two algorithms is clearly illustrated in Figure 3.5.

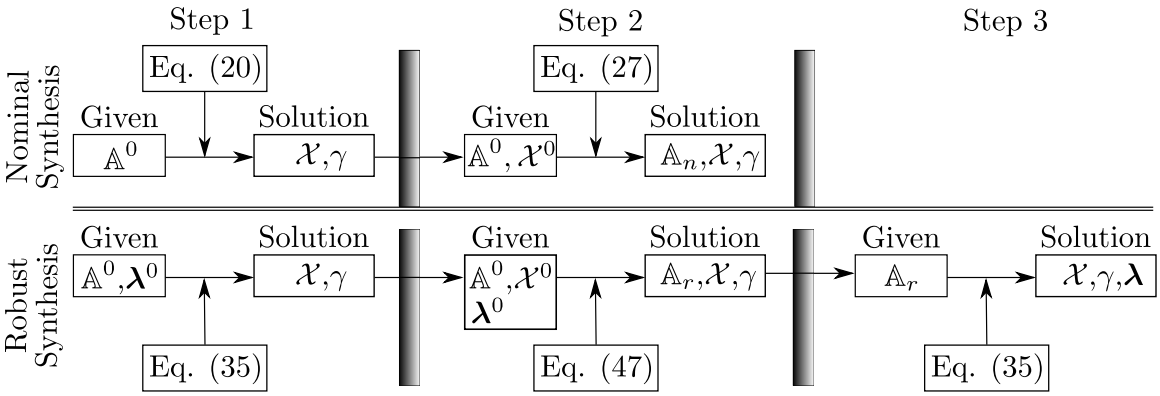


Figure 3.5. Graphical illustration of Algorithm 1. .

3.3 Numerical Verification

Verification of the edge weight synthesis for nominal H_∞ performance using convex concave decomposition based sequential method is executed on *Intel(R) Core i7-4720HQ CPU @2.60 GHz 16GB RAM PC* with Matlab 2019b software. Solutions are obtained using MATLAB LMI toolbox. Dynamics of the individual agents are defined as LPV model of short period dynamics of F16 Vista aircraft(see (3.56)) and cooperative system is composed of three of these agents. LPV model for the agents [126] uses Mach number (M) and altitude (h) as scheduling parameters.

Short period equations of motion is provided in (3.56) with first order approximation of the actuator dynamics. $\boldsymbol{x}_{lng} \in \mathbb{R}^2$ denotes the states of the longitudinal dynamics, which are angle of attack and pitch rate, α and q, respectively, while $x_{act} \in \mathbb{R}$ denotes states of the actuator dynamics. Elevator deflection is denoted as δ_e and dimensional stability and control derivatives, which depends on altitude and Mach number (h, M) at trimmed level flight, are denoted as Z_α, M_α, M_q, Z_e and M_e. The derivatives are expressed as in (3.57).

$$\begin{bmatrix} \dot{\boldsymbol{x}}_{lng} \\ \dot{x}_{act} \end{bmatrix} = \left[\begin{array}{cc|c} Z_\alpha & 1 & 20.2Z_e \\ M_\alpha & M_q & 20.2M_e \\ \hline 0 & 0 & -20.2 \end{array}\right] \begin{bmatrix} \boldsymbol{x}_{lng} \\ \hline x_{act} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \hline 1 \end{bmatrix} \delta_e$$
$$y = \left[\begin{array}{cc|c} 1 & 0 & 0 \end{array}\right] \begin{bmatrix} \boldsymbol{x}_{lng} \\ \hline x_{act} \end{bmatrix} \tag{3.56}$$

$$\begin{aligned} Z_\alpha(h,M) &= 0.22 - 4.1\times10^{-7}h - 2.6M \\ &\quad + 5.15\times10^{-5}Mh \\ M_\alpha(h,M) &= 17.1 - 8.07\times10^{-4}h - 68.4M \\ &\quad + 3.31\times10^{-3}Mh + 56.2M^2 \\ &\quad - 10^{-3}M^2h \\ M_q(h,M) &= -0.228 + 7.06\times10^{-6}h - 2.12M \\ &\quad + 4.86\times10^{-5}Mh \\ Z_{\delta_e}(h,M) &= -1.38\times10^{-3} + 8.75\times10^{-8}h \\ &\quad - 0.34M + 7.98\times10^{-6}Mh \\ M_{\delta_e}(h,M) &= -8.16 + 1.73\times10^{-4}h + 40.6M \\ &\quad - 8.96\times10^{-4}Mh - 99.3M^2 \\ &\quad + 2.42\times10^{-3}M^2h \end{aligned} \tag{3.57}$$

It should be noted that **Agents** 1, 2 and 3 are individually unstable and controllers

$$\begin{aligned} \boldsymbol{K}_1 &= [-0.7445,\ -1.3156,\ 6.8232] \\ \boldsymbol{K}_2 &= [-0.7818 - 1.3827,\ 7.1276] \\ \boldsymbol{K}_3 &= [-0.8174 - 1.4469,\ 7.4156] \end{aligned} \tag{3.58}$$

are used to make these agents stable, respectively.

3.3.1 Verification of Adjacency Synthesis for Nominal H_∞ Performance

For nominal synthesis varying parameters are selected as $h = 1000$ ft and $M = 0.35$. The agents are collected as a single group of agents in a block diagonal fashion such that $\boldsymbol{S} = diag\{\boldsymbol{G}^i\}_{i=1}^N$. Then these agents are connected to each other with an adjacency matrix, $\mathbb{A}$, which is defined as a function of a free scalar variable a and this is provided in (3.59). a is contained in an interval defined as $a = [0,\ 1]$. This interval is selected arbitrarily without violating the fact that edge weight a_{ij} is $0 \leq a_{ij} \leq 1$. Using the system $\boldsymbol{S}$ and the adjacency matrix $\mathbb{A}(a)$, a MAS $\boldsymbol{H}$ is created as described as in (6.23).

$$\mathbb{A}(a) = \begin{bmatrix} 0 & a & (1-a) \\ a & 0 & (1-a) \\ (1-a) & a & 0 \end{bmatrix} \tag{3.59}$$

Verification of the method of synthesizing adjacency weights is performed by comparing the results of sequential LMI approach against the results of conventional calculation of γ over the entire range of a. For this reason, MAS ($\boldsymbol{H}$) is calculated for a range of a value over its prescribed range with increments of $\Delta a = 0.1$ and then γ is calculated for every single MAS using the optimization defined in (3.25). Collection of these results projects the behaviour of the γ of the MAS with the change in a and they are provided in Figure 3.6. In this figure, red line shows value of H_∞ metric γ of the system.

After γ with respect to a is calculated as a benchmark, sequential LMI approach is applied to a MAS created with $\boldsymbol{S}$ and $\mathbb{A}^0$ for $a^0 = 0.3$. This results in a $\mathbb{A}^0$ as provided in (3.60).

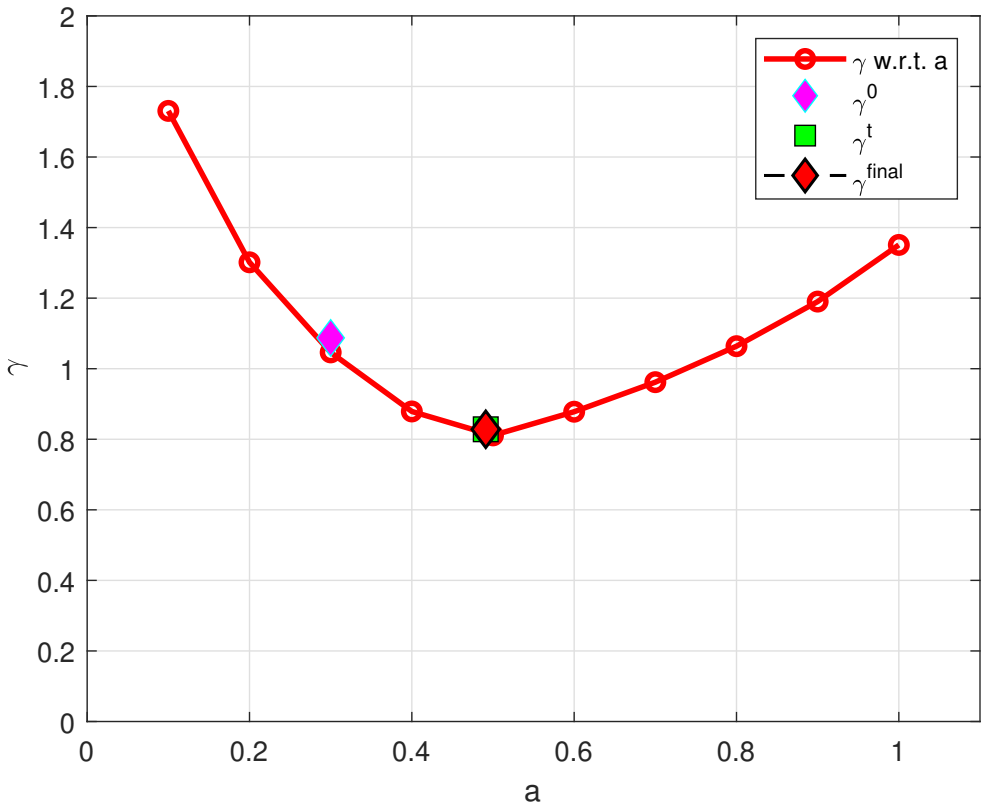


Figure 3.6. Nominal $\mathbb{A}^{nom}(a)$ synthesis results are $h = 1000$ ft and $M = 0.35$..

$$\mathbb{A}^0 = \begin{bmatrix} 0 & 0.3 & 0.7 \\ 0.3 & 0 & 0.7 \\ 0.7 & 0.3 & 0 \end{bmatrix} \tag{3.60}$$

For initial $\mathbb{A}^0$, initial solutions for $\mathcal{X}^0$ and γ^0 are calculated using the optimization described in (3.25) and γ^0 is calculated to be $\gamma^0 = 1.0874$, which is illustrated with a magenta diamond in Figure 3.6.

Set of initial solutions are then supplied to the sequential LMI approach as existing solutions, which are plugged into the variables that has the superscript $\{\cdot\}^t$. For the first

iteration existing solution is the initial solution and for consequent steps existing solution is the result of the optimization described in (3.32) obtained in the previous step. Starting from the initial solution, a converging set of solutions are calculated iteratively until the termination criteria is satisfied, which is $\|\{\gamma\}^t - \{\gamma\}\| < 0.001$. γ values obtained in this phase are illustrated with green squares and final value is represented with a red diamond in Figure 3.6.

After the iterations γ converges to its minimum, which is $\gamma = 0.832$ and a converges to $a = 0.491$. Resulting adjacency matrix is denoted as $\mathbb{A}_n$ and provided in (3.61).

$$\mathbb{A}_n = \begin{bmatrix} 0 & 0.491 & 0.509 \\ 0.491 & 0 & 0.509 \\ 0.509 & 0.491 & 0 \end{bmatrix} \tag{3.61}$$

3.3.2 Verification of Robust Adjacency Matrix Synthesis with IQCs

Identical MAS with three vehicles are introduced an uncertainty channel $\boldsymbol{w}_1 - \boldsymbol{z}_1$ as given in Section 3.2.2. To inject effects of uncertainty, extended system is constructed. This extended system is graphically given in Figure 3.3. Its state space representation is denoted as $\boldsymbol{H}_{E,K}$ and given in (3.37). Interval for a is contained in the same interval defined previously. IQC multipliers $\mathbf{\Pi}_1$ and $\mathbf{\Pi}_2$ are given in (3.62) to ensure $\|\mathbf{\Delta}\| \leq 1$ and monotonically odd non-linearities are modelled, where $\mathbf{H}(j\omega) = (j\omega + 1)^{-1}$.

$$\begin{aligned} \mathbf{\Pi}_1 &= \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \\ \mathbf{\Pi}_2 &= \begin{bmatrix} 0 & 1 + \mathbf{H}(j\omega) \\ 1 + \mathbf{H}(-j\omega) & -2(1 + Re(\mathbf{H}(j\omega))) \end{bmatrix} \end{aligned} \tag{3.62}$$

If the multipliers are partitioned as in (3.19), then an enlarged multiplier is as given in (3.63), where N and n_z are the number of agents and dimension of the output vector $\boldsymbol{z}$ of each vehicle, which is 1 for this work.

$$\boldsymbol{\Pi} := \left[\begin{array}{c|c} \text{diag}\{\boldsymbol{\Pi}_{11}\}_1^{Nn_z} & \text{diag}\{\boldsymbol{\Pi}_{12}\}_1^{Nn_z} \\ \hline \text{diag}\{\boldsymbol{\Pi}_{21}\}_1^{Nn_z} & \text{diag}\{\boldsymbol{\Pi}_{22}\}_1^{Nn_z} \end{array}\right] \tag{3.63}$$

Once the uncertain MAS is set, the same numerical verification procedure is conducted for the robust synthesis case. First, a γ mapping for the entire range of α is calculated. The range of a is given as $a = [0,\ 1]$ with increments of $\Delta a = 0.1$. Then the mapping for γ is calculated using optimization of γ with LMI constraints given in (3.42). The results for these calculations are illustrated in Figure 3.7 with red crosses. It should be noted that in these calculations $\boldsymbol{\lambda} = [1,\ 0]$. Then steps of Algorithm 2 is followed. Magenta diamond in Figure 3.7 shows the initial γ^0 that is calculated along with the $\mathcal{X}^0$ for $\mathbb{A}^0$ given in (3.60). These results satisfy the Assumption 3.2.3. Then **Step 2** of the algorithm, which is the sLMI method for robust synthesis given in (3.54), is conducted. The sequential optimization reveals results given with green squares in Figure 3.7. Finally, **Step 3** of the algorithm is run for optimization of the $\boldsymbol{\lambda}$ to reveal the upper bound for γ. In this step, γ^{final} (red diamond in Figure 3.7) is obtained for optimized $\boldsymbol{\lambda}$ denoted as $\boldsymbol{\lambda}^{opt}$. To illustrate the effect of the final step, a final mapping for γ for optimized $\boldsymbol{\lambda}$ is plotted in Figure 3.7 with blue diamonds. Numerical values for these results are $\boldsymbol{\lambda}^{opt} = [17.029\ \ 42.073]$, $\gamma^{final} = 0.812$ and $a^{opt} = 0.4816$.

Another numerical verification test is designed to show how robust adjacency matrix affects the time response of the MAS. To do that, cooperative systems are created as provided in Figure 3.8 using nominal and robust adjacency matrices $\mathbb{A}^0$ and $\mathbb{A}_r$, respectively. Here topology of the adjacency matrices are kept same. For time response comparison, the

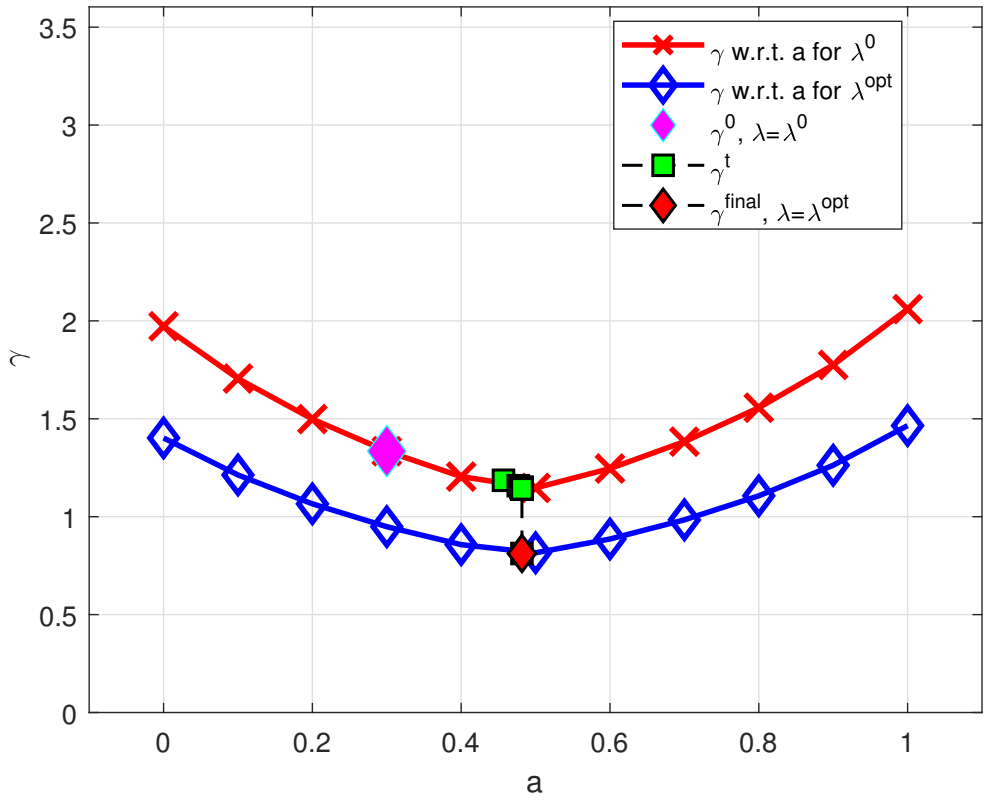


Figure 3.7. Robust $\mathbb{A}(\alpha)$ synthesis results at $h = 1000$ ft and $M = 0.35$. .

uncertainty Δ^i is created such that $\boldsymbol{w}_1^T = \left[\boldsymbol{w}_{1,1}^T,\ \boldsymbol{w}_{1,2}^T,\ \boldsymbol{w}_{1,3}^T\right]^T$ and $\boldsymbol{z}_1^T = \left[\boldsymbol{z}_{1,1}^T,\ \boldsymbol{z}_{1,2}^T,\ \boldsymbol{z}_{1,3}^T\right]^T$ are mapped as given in (3.64), where $\zeta > 0$ and $t_{set,i}|_{i=1}^{N=3} = \{1,\ 1,\ 1\}$ seconds.

$$
\begin{aligned}
\boldsymbol{w}_{1,i} &= \Delta^i(\boldsymbol{z}_{1,i}) \\
\boldsymbol{w}_{1,i} &= \delta(t - t_{set,i}) \cdot \boldsymbol{z}_{1,i} \\
\delta(t - t_{set,i}) &= \begin{cases} 1 & \text{if } \ t \geq t_{set,i}, \\ & \quad t - t_{set,i} \ < \ t \ < \ t_{set,i} + \zeta \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
\tag{3.64}
$$

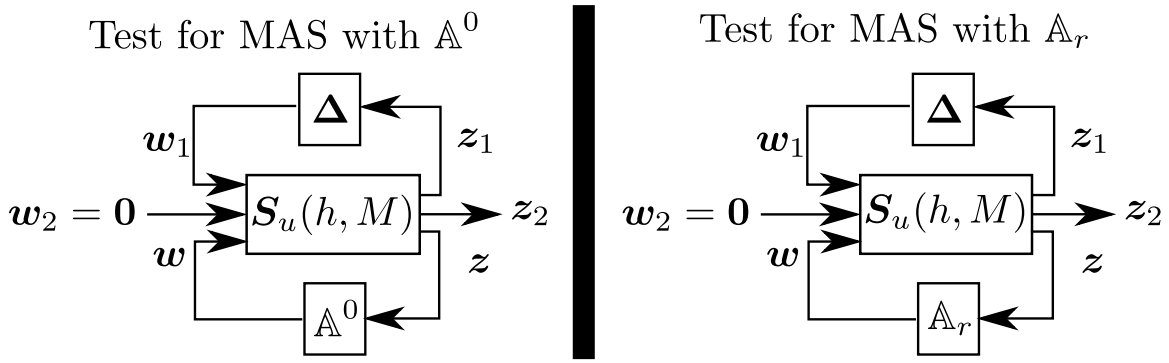


Figure 3.8. Graphical representation of the MAS for time response comparison of MAS with $\mathbb{A}_0$ and $\mathbb{A}_r$ ..

Time response of the MAS with $\mathbb{A}^0$ and $\mathbb{A}_r$ are provided in Figure 3.9. "Initial MAS" and "Robust MAS" denotes MAS with $\mathbb{A}^0$ and $\mathbb{A}_r$, respectively. $\boldsymbol{G}^i$ are the agents. As it is clearly seen, "Robust MAS" (black color coded in the figure) shows a better performance in terms of peak values and rejection of the change in inflow conditions in terms of α

A final analysis that is done for the LPV MAS with developed method is to find robust H_∞ metric over the flight envelope of $\mathrm{h} = [1000,\ \cdots,\ 10000]$ with 1000 ft increments and $\mathrm{M} = [0.5,\ 0.6,\ 0.7]$. It should be noted that, (3.56) are trimmed for a larger flight envelope and this analysis can be extended throughout the entire envelope. During this analysis, agent dynamics, controllers and adjacency matrix topology is kept same as in 3.3.1 and 3.3.2. The method used for this analysis is the robust synthesis method that as is defined in 3.2.2. The strategy followed here is to provide an initial $\mathbb{A}(a = 0.3)$ for each grid point in the flight envelope and find optimal a and γ. The results for this method is illustrated in Figure 3.10, where results are partitioned into two subplots that are titled as H_∞ *Mapping* and a *Mapping*. In H_∞ *Mapping*, areas color coded with blue represents higher robustness while yellow represents lower robustness for the MAS. In a *Mapping*, areas color coded with blue represents lower a while yellow represents higher a for the $\mathbb{A}(a)$. H_∞ *Mapping* reveals that MAS has better H_∞ performance as it operates at lower

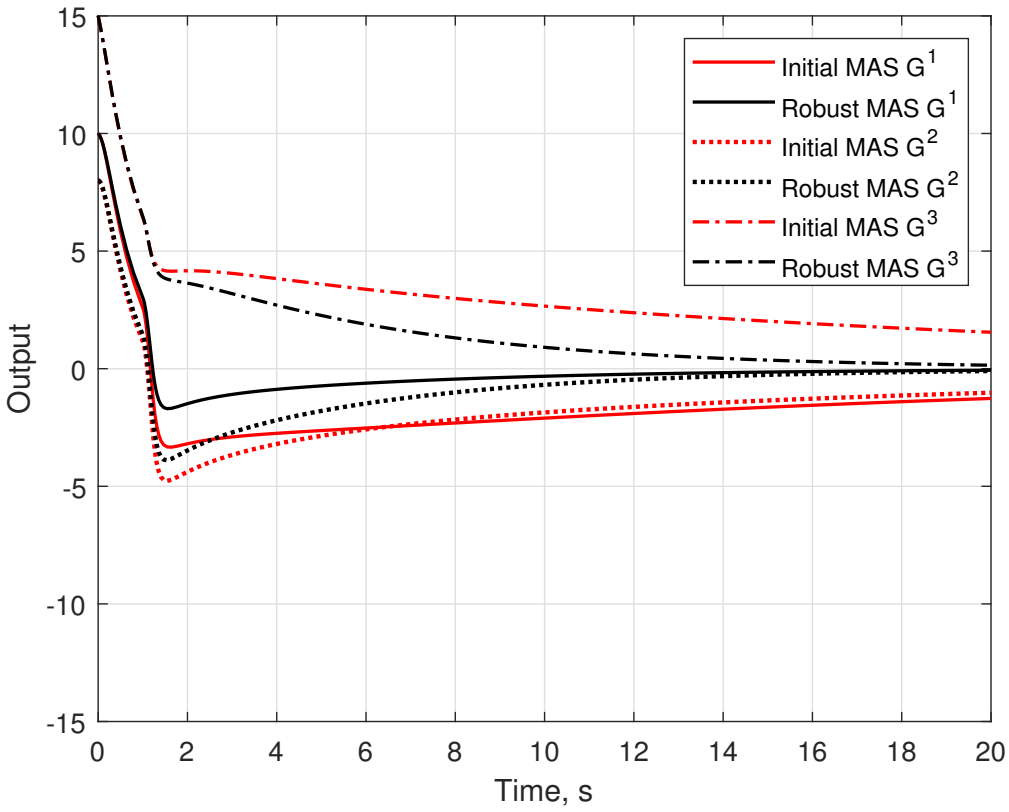


Figure 3.9. Time response comparison of $\mathbb{A}^0$ and $\mathbb{A}_r(\alpha)$ for described uncertainty in (3.64)
..

altitude with higher air speed. The performance drops as the MAS operates at higher altitude and lower air speed. It should be noted that this performance characteristic is not only related to the altitude and air speed but also related to the a *Mapping* as this is a product of the robust synthesis methodology along with the a *Mapping*. When a *Mapping* is investigated for previously described H_∞ *Mapping*, it can be seen that a ranges from 0.36 to 0.385.

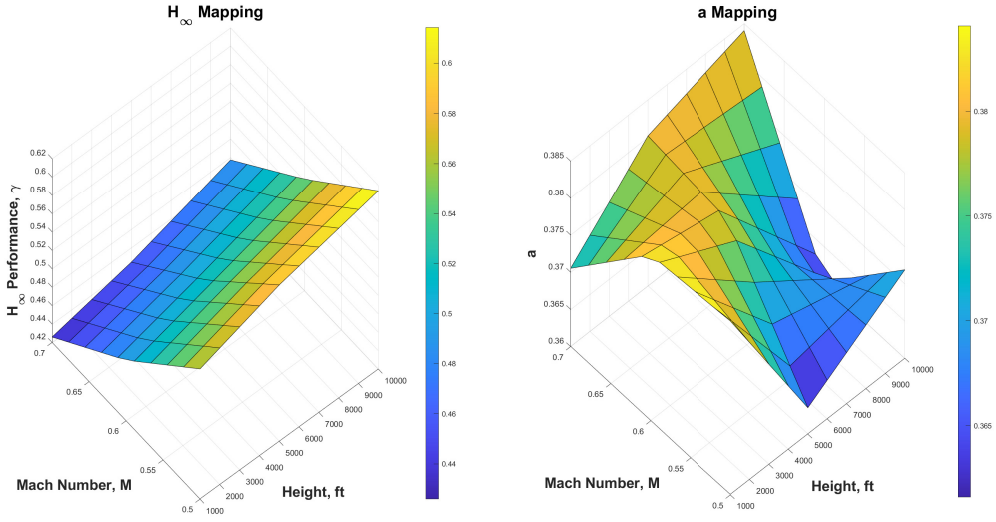


Figure 3.10. $\gamma(\mathrm{h}, \mathrm{M})$ and $a(\mathrm{h}, \mathrm{M})$ mapping for the LPV MAS using the method provided in 3.2.2 ..

3.4 Conclusions

This paper focuses on designing a LPV MAS system by finding the optimal values for edge weights in terms of enhancing H_∞ performance criteria under uncertainties. Without considering the uncertainties, this is an optimization problem with a BMI constraint, and in this paper it is converted to LMI optimization problem and solved in an iterative fashion using sequential LMI approach with convex-concave decomposition. Then the methodology is extended to include valid time domain IQCs as described in 3.3.2. By this way, sLMI method is extended to be used for robust edge weight synthesis for LPV MAS.

This is illustrated on a cooperative system with three F16 Vista aircrafts. Given the topology, edge weights of these systems are synthesized and final results are represented. These results reveal following key conclusions.

1. As shown in in Figure 3.6 and Figure 3.7, synthesized adjacency matrices provide optimal γ values in terms of H_∞ performance for nominal and uncertain MAS. Agents of these systems have slightly different dynamics due to difference in controllers

$\boldsymbol{K}_1$, $\boldsymbol{K}_2$ and $\boldsymbol{K}_3$. For the MAS, optimal γ and a values calculated to be similar, however, this result is not necessarily true as explained in the next item.

2. The MAS with the $\mathbb{A}_r$ is compared to the one with $\mathbb{A}^0$ in time response comparison plot illustrated in Figure 3.9. The disturbance is given in terms of change in inflow conditions as formalized in (3.64). This plot proves that, edge weight synthesis allows improved robust performance without providing cooperative controllers.
3. Finally, an $a(\mathrm{h}, \mathrm{M})$ mapping for LPV MAS is provided for the entire flight envelope of the F16 Vista longitudinal dynamics.

Acknowledgment

Distributed version of this work without uncertainties is published in [127]. This work is supported by the Office of Naval Research via award number N00014-18-1-2215.

Chapter 4

Distributed H_∞ Edge Weight Synthesis for Cooperative Systems Using Layered Successive Linearization and Sequential Optimization

4.1 Preliminaries

Let $\mathbb{R}$ and $\mathbb{R}_{++}$ denote real and positive real numbers, respectively. $\mathbb{R}_+$ be real number that is equal or greater than 0. Similarly, $\mathbb{R}_{--}$ and $\mathbb{R}_-$ represent negative and non-positive real numbers, respectively. $\boldsymbol{S}^p$ denotes symmetric matrix of size p. K_{ij} represents element in the i^{th} row j^{th} column of a matrix $\boldsymbol{K}$.Matrix inequality conditions are defined with $\geq$ and $\leq$, respectively. A matrix $\boldsymbol{K} = diag\{\cdot\}$, $\boldsymbol{K} = col\{\cdot\}$ represent block diagonal and column concatenation of the arguments within $diag\{\cdot\}$ and $col\{\cdot\}$. $\mathcal{F}_L$ and $\mathcal{F}_U$ stands for lower and upper linear fractional transformation (LFT).

4.1.1 Lemmas and Definitions

Lemma 4.1.1 *[121] A matrix valued mapping $\boldsymbol{K}(\boldsymbol{\mu})$ is psd-convex on a convex subset C if and only if $\boldsymbol{r}^T\boldsymbol{G}(\boldsymbol{\mu})\boldsymbol{r}$ is convex on C. Proof is provided in [121].*

Lemma 4.1.2 *[122] A matrix valued mapping $\boldsymbol{K}(\boldsymbol{\mu})$ is psd-convex on a convex subset C if and only if inequality in* (4.1) *holds for all $\boldsymbol{\mu}$ and $\boldsymbol{\beta}$. Proof is provided in [122].*

$$\boldsymbol{K}(\boldsymbol{\beta}) - \boldsymbol{K}(\boldsymbol{\mu}) \geq D\boldsymbol{K}(\boldsymbol{\mu})(\boldsymbol{\beta} - \boldsymbol{\mu}) \tag{4.1}$$

Definition 4.1.1 *[121, 122] Derivative of the mapping $\boldsymbol{K}(\boldsymbol{\mu})$ at $\boldsymbol{\mu}$ is a linear mapping $D\boldsymbol{K}(\boldsymbol{\mu})$ from $\mathbb{R}^l$ to $\mathbb{R}^{p\times p}$ that is defined as*

$$D\boldsymbol{K}(\boldsymbol{\mu})h \;:=\; \sum_{i=1}^{m} h_i \frac{\partial \boldsymbol{K}}{\partial \mu_i}(\boldsymbol{\mu}),\;\; \forall h \in \mathbb{R}^l. \tag{4.2}$$

where h is a real valued function on μ.

Definition 4.1.2 *[122] A matrix-valued mapping $\boldsymbol{F} \;:\; \mathbb{R}^l \to \mathbb{S}^p$ is denoted to be psd-convex concave mapping if $\boldsymbol{F} \;=\; \boldsymbol{K}_1 - \boldsymbol{K}_2$, such that $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$ are psd-convex.*

4.1.2 Underlying Graph Structure

The interaction/communication among agents in the cooperative systems are described by Graph $\mathcal{G} \;=\; (\mathcal{N}, \mathcal{E})$, which consists of node set $\mathcal{N}$ and edge set $\mathcal{E}$ [128]. Edge set $\mathcal{E} \subset \mathcal{N} \times \mathcal{N}$ is given between nodes $i \in \mathcal{N}$ and $j \in \mathcal{N}$ such that $(j, i) \in \mathcal{E}$ denotes node i receives information from j. Adjacency matrix[1] $\boldsymbol{\Upsilon} = [\upsilon_{ij}] \otimes \boldsymbol{I}_{n_{wj}} \in \mathbb{R}^{N\cdot n_{wi}\times N\cdot n_{wi}}$ of $\mathcal{G}$ is composed of weighting scalars υ, where υ quantifies the strength of the connection from node j to node i. N is the number agents in the cooperative system (CS). Formally, Υ_{ij} is described as in the following equation.

$$\boldsymbol{\Upsilon}_{ij} = \begin{cases} \upsilon > 0, \quad j \neq i, \;\; (j,i) \in \mathcal{E} \\ \upsilon = 0, \quad otherwise \end{cases} \tag{4.3}$$

4.1.3 Construction of the Lumped CS

Construction of CS is described starting from agents, which are denoted as ${}^i\boldsymbol{G}$ for $i \;=\; [1, \cdots, N]$. The state space definition of ${}^i\boldsymbol{G}$ is given as in (4.4), where ${}^i\boldsymbol{x} \in \mathbb{R}^{n_{xi}}$, ${}^i\boldsymbol{w}_1 \in \mathbb{R}^{n_{w1i}}$, ${}^i\boldsymbol{w} \in \mathbb{R}^{n_{wi}}$, ${}^i\boldsymbol{z}_1 \in \mathbb{R}^{n_{z1i}}$ and ${}^i\boldsymbol{z} \in \mathbb{R}^{n_{zi}}$. Agents share output information over

[1] Shared signal sizes denoted as n_{wi} and n_{zi} for all i are assumed to be equal.

the input - output ports denoted as ${}^{i}\boldsymbol{w} - {}^{i}\boldsymbol{z}$ such that $\boldsymbol{w} = \boldsymbol{\Upsilon}\boldsymbol{z}$. Signals over this port is denoted as spatial signals. Channel through ports ${}^{i}\boldsymbol{w}_1 - {}^{i}\boldsymbol{z}_1$ is the performance channel.

$$\left[\begin{array}{c} {}^{i}\dot{\boldsymbol{x}} \\ \hline {}^{i}\boldsymbol{z}_1 \\ {}^{i}\boldsymbol{z} \end{array}\right] = \left[\begin{array}{c|cc} {}^{i}\boldsymbol{A} & {}^{i}\boldsymbol{B}_1 & {}^{i}\boldsymbol{B} \\ \hline {}^{i}\boldsymbol{C}_1 & {}^{i}\boldsymbol{D}_1 & {}^{i}\boldsymbol{E}_1 \\ {}^{i}\boldsymbol{C} & {}^{i}\boldsymbol{F}_1 & \boldsymbol{0} \end{array}\right] \left[\begin{array}{c} {}^{i}\boldsymbol{x} \\ \hline {}^{i}\boldsymbol{w}_1 \\ {}^{i}\boldsymbol{w} \end{array}\right] \tag{4.4}$$

A system $\boldsymbol{G}$ is created that represents a group of agents, which is simply described as $\boldsymbol{G} = \text{diag}\{{}^{i}\boldsymbol{G}\}_{i=1}^{N}$ and given in (4.5), where $\boldsymbol{x} \in \mathbb{R}^{n_x}$, $\boldsymbol{w}_1 \in \mathbb{R}^{n_{w1}}$, $\boldsymbol{w} \in \mathbb{R}^{n_w}$, $\boldsymbol{z}_1 \in \mathbb{R}^{n_{z1}}$ and $\boldsymbol{z} \in \mathbb{R}^{n_z}$. Here $n_x = N \cdot n_{xi}$ and rest of the signal sizes are calculated likewise.

$$\left[\begin{array}{c} \dot{\boldsymbol{x}} \\ \hline \boldsymbol{z}_1 \\ \boldsymbol{z} \end{array}\right] = \left[\begin{array}{c|cc} \boldsymbol{A} & \boldsymbol{B}_1 & \boldsymbol{B} \\ \hline \boldsymbol{C}_1 & \boldsymbol{D}_1 & \boldsymbol{E}_1 \\ \boldsymbol{C} & \boldsymbol{F}_1 & \boldsymbol{0} \end{array}\right] \left[\begin{array}{c} \boldsymbol{x} \\ \hline \boldsymbol{w}_1 \\ \boldsymbol{w} \end{array}\right] \tag{4.5}$$

A cooperative system, $\boldsymbol{H}$, can be constructed using $\boldsymbol{\Upsilon}$ to connect vehicles to each other as represented by lower LFT such that $\boldsymbol{H} = \mathcal{F}_L(\boldsymbol{G}, \boldsymbol{\Upsilon})$. The state space representation of $\boldsymbol{H}$ is provided in (4.6).

$$\left[\begin{array}{c|c} \boldsymbol{\mathcal{A}} & \boldsymbol{\mathcal{B}}_1 \\ \hline \boldsymbol{\mathcal{C}}_1 & \boldsymbol{\mathcal{D}}_1 \end{array}\right] = \left[\begin{array}{c|c} \boldsymbol{A} + \boldsymbol{B}\boldsymbol{\Upsilon}\boldsymbol{C} & \boldsymbol{B}_1 + \boldsymbol{B}\boldsymbol{\Upsilon}\boldsymbol{F}_1 \\ \hline \boldsymbol{C}_1 + \boldsymbol{E}_1\boldsymbol{\Upsilon}\boldsymbol{C} & \boldsymbol{D}_1 + \boldsymbol{E}_1\boldsymbol{\Upsilon}\boldsymbol{F}_1 \end{array}\right] \tag{4.6}$$

Definition 4.1.3 (Dissipativity [129]) *$\boldsymbol{W}$ being input space, $\boldsymbol{Z}$ being output space and $\boldsymbol{X}$ being state space, let $P : \boldsymbol{W} \times \boldsymbol{Z} \to \mathbb{R}$ be a supply function. A system $\boldsymbol{G}$ with supply function P is dissipative if there exists a non-negative storage function $V : \boldsymbol{X} \to \mathbb{R}$ for admissible trajectories of w, z and x, such that*

$$V(x(t_0)) + \int_{t_0}^{t_1} P(w(t), z(t))dt \geq V(x(t_1)), \;\; \forall\, t_0 \leq t_1 \tag{4.7}$$

Considering the H_∞ performance, a quadratic supply function is known to be given by bounded real lemma [55, 130] that is given as in (4.8) for the system ${}^{i}\boldsymbol{G}$.

$$
{}^{i}\Psi \;=\; \left[\begin{array}{c} {}^{i}\boldsymbol{z}_1 \\ \hline {}^{i}\boldsymbol{w}_1 \end{array}\right]^{T} \left[\begin{array}{c|c} -\frac{1}{\gamma^2}\boldsymbol{I} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{I} \end{array}\right] \left[\begin{array}{c} {}^{i}\boldsymbol{z}_1 \\ \hline {}^{i}\boldsymbol{w}_1 \end{array}\right] \tag{4.8}
$$

A general quadratic supply function denoted as ${}^{i}_{j}\Phi$ can be given as in (4.9) to stand for the supply rate carried on edge (j, i). Then interconnections for a cooperative system can be imposed using ${}^{i}_{j}\Phi$ as provided in the literature [55, 75]. Spatial signals are mapped within CS by $\boldsymbol{\Upsilon}$, and in (4.9), $\boldsymbol{\Upsilon}$ is embedded in the correlation matrix, ${}^{i}_{j}\boldsymbol{Y} \in \mathbb{R}^{n_{zi}+n_{wi}\times n_{zi}+n_{wi}}$. This definition basically represents supply rate to agent ${}^{i}\boldsymbol{G}$ due to interconnection with ${}^{j}\boldsymbol{G}$. To describe problem efficiently, let input ${}^{i}\boldsymbol{w}$ and output ${}^{i}\boldsymbol{z}$ be the collection of inputs and outputs between ${}^{i}\boldsymbol{G}$ and ${}^{j}\boldsymbol{G}$ such that ${}^{i}\boldsymbol{w} := col\{{}^{i}_{j}\boldsymbol{w}\}_j$ and ${}^{i}\boldsymbol{z} := col\{{}^{i}_{j}\boldsymbol{z}\}_j$, respectively. Then a general supply function can be written for agent ${}^{i}\boldsymbol{G}$ as given in (4.10).

$$
\begin{aligned}
{}^{i}_{j}\Phi \;&=\; \left[\begin{array}{c} {}^{i}_{j}\boldsymbol{w} \\ \hline {}^{i}_{j}\boldsymbol{z} \end{array}\right]^{T} {}^{i}_{j}\boldsymbol{Y} \left[\begin{array}{c} {}^{i}_{j}\boldsymbol{w} \\ \hline {}^{i}_{j}\boldsymbol{z} \end{array}\right] \\
{}^{i}_{j}\boldsymbol{Y} \;&=\; \left[\begin{array}{c|c} {}^{i}_{j}\boldsymbol{Y}_{11} & {}^{i}_{j}\boldsymbol{Y}_{12} \\ \hline {}^{i}_{j}\boldsymbol{Y}_{21} & {}^{i}_{j}\boldsymbol{Y}_{22} \end{array}\right]
\end{aligned} \tag{4.9}
$$

$$
\begin{aligned}
{}^{i}\Phi \;&=\; \sum_{j=1}^{N} \left[\begin{array}{c} {}^{i}_{j}\boldsymbol{w} \\ \hline {}^{i}_{j}\boldsymbol{z} \end{array}\right]^{T} {}^{i}_{j}\boldsymbol{Y} \left[\begin{array}{c} {}^{i}_{j}\boldsymbol{w} \\ \hline {}^{i}_{j}\boldsymbol{z} \end{array}\right] \\
&=\; \left[\begin{array}{c} {}^{i}\boldsymbol{w} \\ \hline {}^{i}\boldsymbol{z} \end{array}\right]^{T} {}^{i}\boldsymbol{Z} \left[\begin{array}{c} {}^{i}\boldsymbol{w} \\ \hline {}^{i}\boldsymbol{z} \end{array}\right] \\
{}^{i}\boldsymbol{Z} \;&=\; \left[\begin{array}{c|c} {}^{i}\boldsymbol{Z}_{11} & {}^{i}\boldsymbol{Z}_{12} \\ \hline {}^{i}\boldsymbol{Z}_{12}^{T} & {}^{i}\boldsymbol{Z}_{22} \end{array}\right]
\end{aligned} \tag{4.10}
$$

, where ${}^{i}\boldsymbol{Z}_{11} \in \mathbb{R}^{n_{wi}\cdot N\times n_{wi}\cdot N}$, ${}^{i}\boldsymbol{Z}_{12} \in \mathbb{R}^{n_{wi}\cdot N\times n_{zi}\cdot N}$ and ${}^{i}\boldsymbol{Z}_{22} \in \mathbb{R}^{n_{wi}\cdot N\times n_{zi}\cdot N}$.

Similar to (4.9), ${}^{j}_{i}\Phi$ represents the supply function of agent ${}^{j}\boldsymbol{G}$ and given in (4.11) as follows,

$$ {}^{j}_{i}\Phi \;=\; \begin{bmatrix} {}^{j}_{i}\boldsymbol{w} \\ {}^{j}_{i}\boldsymbol{z} \end{bmatrix}^{T} {}^{j}_{i}\boldsymbol{Y} \begin{bmatrix} {}^{j}_{i}\boldsymbol{w} \\ {}^{j}_{i}\boldsymbol{z} \end{bmatrix} \tag{4.11} $$

Definition 4.1.4 (Neutrality [75]) *Interconnections are considered to be neutral for neighboring dissipative agents with respect to supply functions as defined in the form of* (4.9) *amd* (4.11)*, if* (4.12) *is satisfied along with the condition in* (4.13)*, where and* ${}^{i}_{j}\Phi$ *is the supply function of agent* ${}^{i}\boldsymbol{G}$ *for* $i, j = [1, \cdots, N]\,,\; i \neq j$.

$$ \begin{bmatrix} {}^{i}_{j}\boldsymbol{z} \\ {}^{i}_{j}\boldsymbol{w} \end{bmatrix} \;=\; \begin{bmatrix} {}^{j}_{i}\boldsymbol{w} \\ {}^{j}_{i}\boldsymbol{z} \end{bmatrix} \quad \forall\, i, j,\; \forall\, t \geq 0 \tag{4.12} $$

$$ {}^{i}_{j}\boldsymbol{\Phi} \;+\; {}^{j}_{i}\Phi \;=\; 0, \quad \forall\, i = [1, \cdots, N] \tag{4.13} $$

4.1.4 Feasibility Condition for H_∞ Performance of Lumped CS

Assuming system given in (4.6) for a given $\boldsymbol{\Upsilon}$ is well-posed and asymptotically stable, then induced $\mathcal{L}_2$ norm of the system is given by the scalar γ and $\|\boldsymbol{H}\|_2 < \gamma$ for all inputs $\boldsymbol{w}_1 \in \ell_2^{n_{w1}}$ [55]. This equivalently means that matrix inequalities (MIs) in (4.14)

holds true. Pre- and post-multiplying the second inequality in (4.14) with $\begin{bmatrix} \boldsymbol{x}^T & \boldsymbol{w}_1^T \end{bmatrix}$ and $\begin{bmatrix} \boldsymbol{x}^T & \boldsymbol{w}_1^T \end{bmatrix}^T$, yields $\gamma^2\|\boldsymbol{w}_1\|_2^2 - \|\boldsymbol{z}_1\|_2^2 \leq 0$.

$$
\begin{aligned}
\mathcal{X} &\geq \mathbf{0} \\
\begin{bmatrix} \boldsymbol{\mathcal{X}} \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{A}} & \boldsymbol{\mathcal{B}}_1 \end{bmatrix} + \begin{bmatrix} \boldsymbol{\mathcal{A}}^T \\ \boldsymbol{\mathcal{B}}_1^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{X}} & \mathbf{0} \end{bmatrix} & \\
+ \frac{1}{\gamma^2} \begin{bmatrix} \boldsymbol{\mathcal{C}}_1^T \\ \boldsymbol{\mathcal{D}}_1^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{C}}_1 & \boldsymbol{\mathcal{D}}_1 \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\boldsymbol{I} \end{bmatrix} &\leq \mathbf{0}
\end{aligned}
\tag{4.14}
$$

Second inequality given in (4.14) depends non-linearly on matrix variables $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\Upsilon}$. This non-linear matrix inequality falls into the class of bilinear matrix inequalities (BMIs). Linearization and solution to this BMI can be obtained by multiple methods, where one of them is given as in [123]. Drawback of lumped methods are computational burden to calculate feasible solutions to $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\Upsilon}$. Number of unknowns to be solved for this set of MIs is given by $(N \cdot n_{xi}(1 + N \cdot n_{xi})/2) + n_\alpha$, where $n_\alpha \in \mathbb{N}$ denotes the number of independent variables used to define adjacency matrix to be synthesized. As clearly seen, state size of each agent and number of agents in the system quadratically increases the number of unknowns to be solved.

4.2 Problem Formulation

Interconnections in distributed CS model are not defined algebraically but rather they are showing themselves as a supply function in the feasibility analysis. Based on the method proposed in [75], CS given in (4.6) can also be described in a distributed fashion and this is realized by imposing constraints on the spatial signal channel of the agents.

Unlike in lumped CS, constraints cannot reveal themselves in the state space representation, however, they are introduced as supply functions to the storage function of the CS. This is done by introducing these constraints in the MIs.

Distributed CS model relies on an interconnection constraint on spatial signal ports of each agent. This constraint is characterized by a supply function given as in (4.7). The relevance of the supply functions to synthesize edge weight is described in the Definitions 4.1.3 and 4.1.4. However, edge weights in (4.10) are not explicit. Investigating CS, which is given in Figure 4.1, shows that ${}^{i}\boldsymbol{z}$ is scaled by ν_{ji} and this scaling can be revealed as in (4.15) based on (4.10) by braking down the segments of ${}^{i}\boldsymbol{z}$ into the out degrees of agent ${}^{i}\boldsymbol{G}$, which are ${}^{i}_{j}\boldsymbol{z}$ and shared with agents ${}^{j}\boldsymbol{G}^{N}_{j=1,\ j\neq i}$.

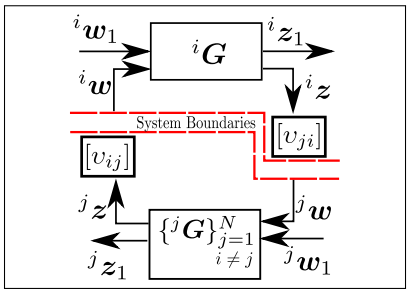


Figure 4.1. Interconnection of agents..

$$
{}^{i}\Phi \;=\; \left[\begin{array}{c} {}^{i}\boldsymbol{w} \\ \hline {}^{i}\boldsymbol{z} \end{array}\right]^{T} \left[\begin{array}{c|c} \boldsymbol{I} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{I}\nu_{1i} \\ \vdots & \vdots \\ \boldsymbol{0} & \boldsymbol{I}\nu_{Ni} \end{array}\right]^{T} \left[\begin{array}{c|c} {}^{i}\boldsymbol{Z}_{11} & {}^{i}\boldsymbol{Z}_{12} \\ \hline {}^{i}\boldsymbol{Z}_{12}^{T} & {}^{i}\boldsymbol{Z}_{22} \end{array}\right] \left[\begin{array}{c|c} \boldsymbol{I} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{I}\nu_{1i} \\ \vdots & \vdots \\ \boldsymbol{0} & \boldsymbol{I}\nu_{Ni} \end{array}\right] \left[\begin{array}{c} {}^{i}\boldsymbol{w} \\ \hline {}^{i}\boldsymbol{z} \end{array}\right] \tag{4.15}
$$

Using ${}^{i}\boldsymbol{\Psi}$ and ${}^{i}\boldsymbol{\Phi}$, H_∞ performance of γ for ${}^{i}\boldsymbol{G}$ in a CS with N agents and given $\boldsymbol{\Upsilon}$ is equivalent to the statements given in (4.16) (Theorem 1 in [55]).

$$
\begin{aligned}
{}^{i}\boldsymbol{X} \;&\geq\; \mathbf{0} \\
{}^{i}\boldsymbol{M}_1^{Ti}\boldsymbol{M}_2{}^{i}\boldsymbol{M}_1 \;&\leq\; \mathbf{0}
\end{aligned}
\tag{4.16}
$$

Given the decision variables $\gamma > 0 \in \mathbb{R}$, ${}^{i}\boldsymbol{X} \;\geq\; \mathbf{0} \;\in \mathbb{S}^{n_{xi}}$, ${}^{i}\boldsymbol{Z}_{11}$, ${}^{i}\boldsymbol{Z}_{12}$, ${}^{i}\boldsymbol{Z}_{21} \;=\; {}^{i}\boldsymbol{Z}_{12}^{T}$, ${}^{i}\boldsymbol{Z}_{22}$ and $\nu_{ji} \;\in \mathbb{R}$ for $i, j = [1, \cdots, N]\,,\; j \neq i$, matrix valued functions ${}^{i}\boldsymbol{M}_1(\nu_{ji})$ and $\boldsymbol{M}_2(\gamma, {}^{i}\boldsymbol{X}, {}^{i}\boldsymbol{Z})$ are given in (4.17). For brevity, decision variables of matrix valued function representations will be omitted in the notation throughout the paper.

$$
\begin{aligned}
{}^{i}\boldsymbol{M}_1 \;&=\; \left[\begin{array}{c} {}^{i}\boldsymbol{M}_{1,1} \\ \hline {}^{i}\boldsymbol{M}_{1,2} \end{array}\right] \;=\; \left[\begin{array}{ccc} \boldsymbol{I}_{n_{nxi}} & \mathbf{0} & \mathbf{0} \\ {}^{i}\boldsymbol{A} & {}^{i}\boldsymbol{B}_1 & row\{{}^{i}\boldsymbol{B}\}_{i=1}^{N} \\ \mathbf{0} & \boldsymbol{I}_{n_{w1i}} & \mathbf{0} \\ {}^{i}\boldsymbol{C}_1 & {}^{i}\boldsymbol{D}_1 & row\{{}^{i}\boldsymbol{E}_1\}_{i=1}^{N} \\ \hline \mathbf{0} & \mathbf{0} & \boldsymbol{I}_{n_w} \\ \nu_{1i}{}^{i}\boldsymbol{C} & \nu_{1i}{}^{i}\boldsymbol{F}_1 & \mathbf{0} \\ \vdots & \vdots & \vdots \\ \nu_{Ni}{}^{i}\boldsymbol{C} & \nu_{Ni}{}^{i}\boldsymbol{F}_1 & \mathbf{0} \end{array}\right] \\
{}^{i}\boldsymbol{M}_2 \;&=\; \left[\begin{array}{c|c} {}^{i}\boldsymbol{M}_{2,1} & \mathbf{0} \\ \hline \mathbf{0} & {}^{i}\boldsymbol{M}_{2,2} \end{array}\right] \\
&=\; \left[\begin{array}{cccc|cc} \mathbf{0} & {}^{i}\boldsymbol{X} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ {}^{i}\boldsymbol{X} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \frac{1}{\gamma^2}\boldsymbol{I} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & {}^{i}\boldsymbol{Z}_{11} & {}^{i}\boldsymbol{Z}_{12} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & {}^{i}\boldsymbol{Z}_{21} & {}^{i}\boldsymbol{Z}_{22} \end{array}\right]
\end{aligned}
\tag{4.17}
$$

Set of inequalities given in (4.16) can be partitioned into linear and non-linear parts of the form given in (4.18), where partitions ${}^{i}\boldsymbol{T}_1$ and ${}^{i}\boldsymbol{T}_2$ are provided explicitly in (4.19).

$$ {}^{i}\boldsymbol{T}_1 + {}^{i}\boldsymbol{T}_2 \leq \boldsymbol{0} \tag{4.18} $$

$$ \begin{aligned} {}^{i}\boldsymbol{T}_1 &:= {}^{i}\boldsymbol{M}_{1,1}^{T}\, {}^{i}\boldsymbol{M}_{2,1}\, {}^{i}\boldsymbol{M}_{1,1} \\ {}^{i}\boldsymbol{T}_2 &:= {}^{i}\boldsymbol{M}_{1,2}^{T}\, {}^{i}\boldsymbol{M}_{2,2}\, {}^{i}\boldsymbol{M}_{1,2} \end{aligned} \tag{4.19} $$

Thus, the distributed nominal H_∞ edge weight synthesis problem can be written as in (4.20), where ${}^{i}\boldsymbol{\Phi} = {}^{i}\boldsymbol{T}_2$.

$$ \begin{aligned} \text{minimize} \quad & {}^{i}\gamma \\ \text{s. t.} \quad & {}^{i}\boldsymbol{X} \quad \geq \boldsymbol{0} \\ & {}^{i}\boldsymbol{T}_1 + {}^{i}\boldsymbol{T}_2 \quad \leq \boldsymbol{0} \end{aligned} \tag{4.20} $$

By looking at (4.19) and (4.20), it is seen that one of the resulting MIs are non-linear and synthesis of the edge weights is a non-linear problem. Section 4.3 provides a solution to the problem given in (4.20).

4.3 Methodology

Solution to the problem given in (4.20) relies on recomposition of the CS as illustrated in Figure 4.2, where all the mapping due to edge weights are substituted into a new system denoted as ${}^{c}\boldsymbol{G}$. As a result, a new CS is constructed, where agents ${}^{i}\boldsymbol{G}$ are talking only with ${}^{c}\boldsymbol{G}$ with new edge weights denoted as ν_{ic} such that $\nu_{ic} = \nu_{ci} = 1$, while all the mapping that is related to the original mapping is kept within ${}^{c}\boldsymbol{G}$.

Signal definitions and formulation given in previous sections are revisited based on this new mapping and this reveals itself as $j = c$. As a result of this, partitions of ${}^{i}\boldsymbol{M}_1$ and ${}^{i}\boldsymbol{M}_2$ in (4.17) for $i = 1, \cdots, N$ simplifies as given in (4.21), which makes a quadratic term once multiplied.

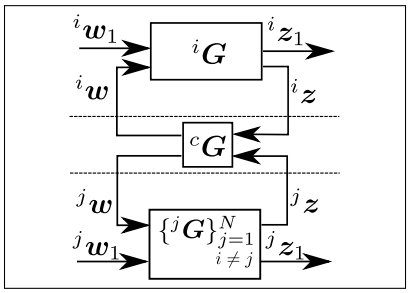


Figure 4.2. New interpretation of the cooperative system..

$$
\begin{aligned}
{}^{i}\boldsymbol{M}_{1,2} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} & \boldsymbol{I}_{n_{wi}} \\ {}^{i}\boldsymbol{C} & {}^{i}\boldsymbol{F}_1 & \mathbf{0} \end{bmatrix} \\
{}^{i}\boldsymbol{M}_{2,22} &= \begin{bmatrix} {}^{i}_{c}\boldsymbol{Y}_{11} & {}^{i}_{c}\boldsymbol{Y}_{12} \\ {}^{i}_{c}\boldsymbol{Y}_{21} & {}^{i}_{c}\boldsymbol{Y}_{22} \end{bmatrix}
\end{aligned}
\tag{4.21}
$$

There is an additional agent in the CS in the new composition and that is ${}^{c}\boldsymbol{G}$, where partitions of ${}^{i}\boldsymbol{M}_1$ and ${}^{i}\boldsymbol{M}_2$ in (4.17) becomes as given in (4.22).

$$
\begin{aligned}
{}^{c}\boldsymbol{M}_{1,1} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I}_{n_w} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Upsilon} & \mathbf{0} \end{bmatrix} \\
{}^{c}\boldsymbol{M}_{1,2} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} & \boldsymbol{I}_{n_w} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Upsilon} \end{bmatrix} \\
{}^{c}\boldsymbol{M}_{2,1} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \frac{1}{\gamma^2}\boldsymbol{I} \end{bmatrix} \\
{}^{c}\boldsymbol{M}_{2,2} &= \begin{bmatrix} {}^{c}\boldsymbol{Z}_{11} & {}^{c}\boldsymbol{Z}_{12} \\ {}^{c}\boldsymbol{Z}_{21} & {}^{c}\boldsymbol{Z}_{22} \end{bmatrix}
\end{aligned}
\tag{4.22}
$$

, where

$$
\begin{aligned}
{}^{c}\boldsymbol{Z}_{11} &= diag\{{}^{c}_{i}\boldsymbol{Y}_{11}\}_{i=1}^{N}, \quad {}^{c}_{i}\boldsymbol{Y}_{11} = -{}^{i}_{c}\boldsymbol{Y}_{22} \\
{}^{c}\boldsymbol{Z}_{12} &= diag\{{}^{c}_{i}\boldsymbol{Y}_{12}\}_{i=1}^{N}, \quad {}^{c}_{i}\boldsymbol{Y}_{12} = -{}^{i}_{c}\boldsymbol{Y}_{12}^{T} \\
{}^{c}\boldsymbol{Z}_{22} &= diag\{{}^{c}_{i}\boldsymbol{Y}_{22}\}_{i=1}^{N}, \quad {}^{c}_{i}\boldsymbol{Y}_{22} = -{}^{i}_{c}\boldsymbol{Y}_{11}
\end{aligned}
\tag{4.23}
$$

Based on the redefined terms given in (4.21, 4.22 and 4.23), inequality given in (4.18) are written for i and c. Then following lemma can be proposed.

Lemma 4.3.1 *Assuming ${}^{i}\boldsymbol{G}$ and ${}^{c}\boldsymbol{G}$ maintain neutrality as defined in Definition 4.1.4, let ${}^{i}\boldsymbol{G}$ for $i = 1, \cdots, N$ be dissipative agents with state space realization of* (4.4)*. CS consisting of ${}^{i}\boldsymbol{G}$ have a robust performance of γ for input ${}^{i}_{c}w \in \ell_2^{n_{wi}}$ for $i = [1, \cdots, N]$ if there exists ${}^{i}\boldsymbol{X} \in \mathbb{S}^{n_{xi}}$, ${}^{i}_{c}\boldsymbol{Y}_{11} \in \mathbb{S}^{n_{wi}}$, ${}^{i}_{c}\boldsymbol{Y}_{22} \in \mathbb{S}^{n_{wi}}$, ${}^{i}_{c}\boldsymbol{Y}_{12} \in \mathbb{S}_{s}^{n_{wi}}$, ${}^{c}\boldsymbol{Z}_{11} \in \mathbb{S}^{N \cdot n_{wi}}$, ${}^{c}\boldsymbol{Z}_{12} \in \mathbb{S}_{s}^{N \cdot n_{wi}}$ and ${}^{c}\boldsymbol{Z}_{22} \in \mathbb{S}^{N \cdot n_{wi}}$ such that*

$$
\begin{aligned}
{}^{i}\hat{\boldsymbol{X}} \;&\geq\; \mathbf{0} \\
{}^{i}\boldsymbol{T}_1 + {}^{i}\boldsymbol{T}_2 \;&\leq\; \mathbf{0} \\
{}^{c}\boldsymbol{T}_1 + {}^{c}\boldsymbol{T}_2 \;&\leq\; \mathbf{0}
\end{aligned}
\tag{4.24}
$$

It should be noted that, second inequality in (4.24) is quadratic, which can be linearized, however, third inequality is non-linear. Assuming there is a valid linear approximation to and ${}^{c}\boldsymbol{T}_2$ that satisfies inequalities in (4.24), Theorem 4.3.2 can be written. Proof to Lemma 4.3.1 will be provided along with the proof of Theorem 4.3.2 in proof.

Theorem 4.3.2 *Let ${}^{i}\boldsymbol{G}$ for $i = 1, \cdots, N$ be dissipative agents with state space realization of* (4.4)*. The following two statements are valid:*

1. *Assuming agents maintain neutrality as defined in Definition 4.1.4, CS consisting of ${}^{i}\boldsymbol{G}$ achieves induced $\mathcal{L}_2$ gain performance of γ^* for input ${}^{i}w_1 \in \ell_2^{N \cdot n_{w1i}}$ for $i = [1, \cdots, N]$ if there exist matrices ${}^{i}\boldsymbol{X} = {}^{i}\boldsymbol{X}^T$, ${}^{i}_{j}\boldsymbol{Y}_{11}$, ${}^{i}_{j}\boldsymbol{Y}_{12}$, ${}^{i}_{j}\boldsymbol{Y}_{22}$ and ν_{ji} such that LMIs in* (4.20) *is satisfied for $i, j = [1, \cdots, N]$.*
2. *CS consisting of ${}^{i}\boldsymbol{G}$ and ${}^{c}\boldsymbol{G}$ achieves induced $\mathcal{L}_2$ gain performance of γ such that $\gamma^2 := \gamma^{*2} + \alpha^2$ if Lemma 4.3.1 is satisfied.*

Proof. Second inequality in (4.20) is pre- and post-multiplied with state-input vector of $\left[{}^{i}\boldsymbol{x}^T \; {}^{i}\boldsymbol{w}_1^T \; {}^{i}\boldsymbol{w}^T\right]^T$. Summing these inequalities over i yields the results as in (4.25). It should be noted that ${}^{i}\boldsymbol{w}$ is as given in (4.10).

$$
\sum_{i=1}^{N} {}^{i}\dot{V} \;+\; \sum_{i=1}^{N} \left(\frac{1}{\gamma^2} {}^{i}\boldsymbol{z}_1^{Ti}\boldsymbol{z}_1 \;-\; {}^{i}\boldsymbol{w}_1^{Ti}\boldsymbol{w}_1\right) \;+\; \sum_{i=1}^{N} {}^{i}\boldsymbol{\Phi} \;\leq\; 0. \tag{4.25}
$$

, where

$$
\begin{aligned}
{}^{i}\dot{V} \;=\; & {}^{i}\boldsymbol{x}^{T}({}^{i}\boldsymbol{X}\,{}^{i}\boldsymbol{A}+{}^{i}\boldsymbol{A}^{T}\,{}^{i}\boldsymbol{X}){}^{i}\boldsymbol{x} \;+ \\
& {}^{i}\boldsymbol{x}^{T}\,{}^{i}\boldsymbol{X}\,{}^{i}\boldsymbol{B}_{1}\,{}^{i}\boldsymbol{w}_{1} \;+\; {}^{i}\boldsymbol{x}^{T}\,{}^{i}\boldsymbol{X}\,{}^{i}\boldsymbol{B}\,{}^{i}\boldsymbol{w} \;+ \\
& {}^{i}\boldsymbol{w}^{T}\,{}^{i}\boldsymbol{B}^{T}\,{}^{i}\boldsymbol{X}\,{}^{i}\boldsymbol{x} \;+\; {}^{i}\boldsymbol{w}_{1}^{T}\,{}^{i}\boldsymbol{B}_{1}^{T}\,{}^{i}\boldsymbol{X}\,{}^{i}\boldsymbol{x} \\
{}^{i}\boldsymbol{\Phi} \;=\; & {}^{i}\boldsymbol{z}^{T}\,{}^{i}\boldsymbol{Z}_{22}\,{}^{i}\boldsymbol{z} \;+\; {}^{i}\boldsymbol{z}^{T}\,{}^{i}\boldsymbol{Z}_{12}^{T}\,{}^{i}\boldsymbol{w} \;+ \\
& {}^{i}\boldsymbol{w}^{T}\,{}^{i}\boldsymbol{Z}_{12}\,{}^{i}\boldsymbol{z} \;+\; {}^{i}\boldsymbol{w}^{T}\,{}^{i}\boldsymbol{Z}_{11}\,{}^{i}\boldsymbol{w}
\end{aligned}
\tag{4.26}
$$

Integrating (4.25) along the admissible trajectory for $\left[{}^{i}\boldsymbol{x}\;{}^{i}\boldsymbol{w}_{1}^{T}\;{}^{i}\boldsymbol{w}^{T}\right]^{T}$ from time $t=0$ to $t=T$ and taking $\sum_{i=1}^{N}{}^{i}\boldsymbol{\Phi} \;=\; 0$ yields

$$
V(\{{}^{i}\boldsymbol{x}(T)\}_{i=1}^{N})-V(\{{}^{i}\boldsymbol{x}(0)\}_{i=1}^{N})+\frac{1}{\gamma^{2}}\int_{0}^{T}\sum_{i=1}^{N}{}^{i}\boldsymbol{z}_{1}^{Ti}\boldsymbol{z}_{1}dt-\int_{0}^{T}\sum_{i=1}^{N}{}^{i}\boldsymbol{w}_{1}^{Ti}\boldsymbol{w}_{1}dt \;\leq\; 0 \tag{4.27}
$$

Same procedure is applied to the second and third inequalities of (4.24) for $i = [1,\cdots,N]$ and following results are obtained. This time column matrix that is being pre- and post-multiplied with the second and third inequalities are $\left[{}^{i}\boldsymbol{x}^{T}\;{}^{i}\boldsymbol{w}_{1}^{T}\;{}^{i}_{c}\boldsymbol{w}^{T}\right]^{T}$ for $i = [1,\cdots,N]$ and $\left[\boldsymbol{0}\;{}^{c}\boldsymbol{w}_{1}^{T}\;{}^{c}\boldsymbol{w}^{T}\right]^{T}$, respectively. Summation of the resulting inequalities give (4.28).

$$
\begin{aligned}
\sum_{i=1}^{N}{}^{i}_{c}\dot{\hat{V}} \;+\; \sum_{i=1}^{N}(\frac{1}{\gamma^{2}}{}^{i}\boldsymbol{z}_{1}^{Ti}\boldsymbol{z}_{1} \;-\; {}^{i}\boldsymbol{w}_{1}^{Ti}\boldsymbol{w}_{1}) \;+\; \sum_{i=1}^{N}{}^{i}_{c}\boldsymbol{\Phi} \\
+\; {}^{c}\boldsymbol{\Phi} \;+\; (\frac{1}{\gamma^{2}}{}^{c}\boldsymbol{z}_{1}^{Tc}\boldsymbol{z}_{1} \;-\; {}^{c}\boldsymbol{w}_{1}^{Tc}\boldsymbol{w}_{1}) \;\leq\; 0
\end{aligned}
\tag{4.28}
$$

, where

$$
\begin{aligned}
{}^{i}_{c}\dot{\hat{V}} \;=\; & {}^{i}\boldsymbol{x}^{T}({}^{i}\hat{\boldsymbol{X}}\,{}^{i}\boldsymbol{A} + {}^{i}\boldsymbol{A}^{T}\,{}^{i}\hat{\boldsymbol{X}})^{i}\boldsymbol{x} \;+ \\
& {}^{i}\boldsymbol{x}^{T}\,{}^{i}\hat{\boldsymbol{X}}\,{}^{i}\boldsymbol{B}_{1}\,{}^{i}\boldsymbol{w}_{1} \;+\; {}^{i}\boldsymbol{x}^{T}\,{}^{i}\hat{\boldsymbol{X}}\,{}^{i}\boldsymbol{B}\,{}^{i}\boldsymbol{w} \;+ \\
& {}^{i}\boldsymbol{w}^{T}\,{}^{i}\boldsymbol{B}^{T}\,{}^{i}\hat{\boldsymbol{X}}\,{}^{i}\boldsymbol{x} \;+\; {}^{i}\boldsymbol{w}_{1}^{T}\,{}^{i}\boldsymbol{B}_{1}^{T}\,{}^{i}\hat{\boldsymbol{X}}\,{}^{i}\boldsymbol{x} \\
{}^{i}_{c}\boldsymbol{\Phi} \;=\; & {}^{i}_{c}\boldsymbol{z}^{T}\,{}^{i}_{c}\boldsymbol{Y}_{22}\,{}^{i}_{c}\boldsymbol{z} \;+\; {}^{i}_{c}\boldsymbol{z}^{T}\,{}^{i}_{c}\boldsymbol{Y}_{12}^{T}\,{}^{i}_{c}\boldsymbol{w} \;+ \\
& {}^{i}_{c}\boldsymbol{w}^{T}\,{}^{i}_{c}\boldsymbol{Y}_{12}\,{}^{i}_{c}\boldsymbol{z} \;+\; {}^{i}_{c}\boldsymbol{w}^{T}\,{}^{i}_{c}\boldsymbol{Y}_{11}\,{}^{i}_{c}\boldsymbol{w} \\
{}^{c}\boldsymbol{\Phi} \;=\; & {}^{c}\boldsymbol{z}^{T}\,{}^{c}\boldsymbol{Z}_{22}\,{}^{c}\boldsymbol{z} \;+\; {}^{c}\boldsymbol{z}^{T}\,{}^{c}\boldsymbol{Z}_{12}^{T}\,{}^{c}\boldsymbol{w} \;+ \\
& {}^{c}\boldsymbol{w}^{T}\,{}^{c}\boldsymbol{Z}_{12}\,{}^{c}\boldsymbol{z} \;+\; {}^{c}\boldsymbol{w}^{T}\,{}^{c}\boldsymbol{Z}_{11}\,{}^{c}\boldsymbol{w}
\end{aligned}
\tag{4.29}
$$

$\sum_{i=1}^{N} {}^{i}_{c}\boldsymbol{\Phi} \;+\; {}^{c}\boldsymbol{\Phi} \;=\; 0$ due to the fact that interconnections are neutral, which is dictated to the inequality by relationship given in (4.23).

$$
\begin{aligned}
\sum_{i=1}^{N} {}^{i}\dot{\hat{V}} \;+\; \sum_{i=1}^{N}\Big(\frac{1}{\gamma^{2}}\,{}^{i}\boldsymbol{z}_{1}^{T\,i}\boldsymbol{z}_{1} \;-\; {}^{i}\boldsymbol{w}_{1}^{T\,i}\boldsymbol{w}_{1}\Big) \;+ \\
\Big(\frac{1}{\gamma^{2}}\,{}^{c}\boldsymbol{z}_{1}^{T\,c}\boldsymbol{z}_{1} \;-\; {}^{c}\boldsymbol{w}_{1}^{T\,c}\boldsymbol{w}_{1}\Big) \;\leq\; 0
\end{aligned}
\tag{4.30}
$$

Integrating (4.30) along the admissible trajectory for $\left[{}^{i}\boldsymbol{x}^{T}\;{}^{i}\boldsymbol{w}_{1}^{T}\;{}^{i}_{c}\boldsymbol{w}^{T}\right]^{T}$ for $i = [1, \cdots, N]$ and $\left[\boldsymbol{0}\;{}^{c}\boldsymbol{w}_{1}^{T}\;{}^{c}\boldsymbol{w}^{T}\right]^{T}$ from time $t = 0$ to $t = T$ yields $\hat{V}(\{{}^{i}\boldsymbol{x}(T)\}_{i=1}^{N}) - \hat{V}(\{{}^{i}\boldsymbol{x}(0)\}_{i=1}^{N}) + \frac{1}{\gamma^{2}}\int_{0}^{T}\sum_{i=1}^{N}{}^{i}\boldsymbol{z}_{1}^{T\,i}\boldsymbol{z}_{1}dt - \int_{0}^{T}\sum_{i=1}^{N}{}^{i}\boldsymbol{w}_{1}^{T\,i}\boldsymbol{w}_{1}dt + \frac{1}{\gamma^{2}}\int_{0}^{T}{}^{c}\boldsymbol{z}_{1}^{T\,c}\boldsymbol{z}_{1}dt - \int_{0}^{T}{}^{c}\boldsymbol{w}_{1}^{T\,c}\boldsymbol{w}_{1}dt \;\leq\; 0.$

To prove H_{∞} performance, assume ${}^{i}\boldsymbol{x}(0) \;=\; \boldsymbol{0}$ and ${}^{i}\boldsymbol{x}(T) \;\neq\; \boldsymbol{0}$. Let ${}^{i}\boldsymbol{w}_{1} \;\neq\; \boldsymbol{0}$ and ${}^{c}\boldsymbol{w}_{1} \;\neq\; \boldsymbol{0}$ concatenated as $\boldsymbol{w}_{1}$ and let ${}^{i}\boldsymbol{z}_{1}$ and ${}^{c}\boldsymbol{z}_{1}$ concatenated as $\boldsymbol{z}_{1}$, where (4.31) provides to formal definitions to $\boldsymbol{w}_{1}$ and $\boldsymbol{z}_{1}$, respectively.

$$
\begin{aligned}
\boldsymbol{w}_{1} \;&=\; \begin{bmatrix} col\{{}^{i}\boldsymbol{w}_{1}\}_{i=1}^{N} \\ {}^{c}\boldsymbol{w}_{1} \end{bmatrix} \;=\; \begin{bmatrix} col\{{}^{i}\boldsymbol{w}_{1}\}_{i=1}^{N} \\ col\{{}^{i}_{c}\boldsymbol{z}\}_{i=1}^{N} \end{bmatrix} \\
\boldsymbol{z}_{1} \;&=\; \begin{bmatrix} col\{{}^{i}\boldsymbol{z}_{1}\}_{i=1}^{N} \\ {}^{c}\boldsymbol{z}_{1} \end{bmatrix} \;=\; \begin{bmatrix} col\{{}^{i}\boldsymbol{z}_{1}\}_{i=1}^{N} \\ \boldsymbol{\Upsilon}\,col\{{}^{i}_{c}\boldsymbol{z}\}_{i=1}^{N} \end{bmatrix}
\end{aligned}
\tag{4.31}
$$

Then from (4.30 and 4.31) we get

$$\frac{\|\boldsymbol{z}_1\|_2^2}{\|\boldsymbol{w}_1\|_2^2} \leq \gamma^2 \tag{4.32}$$

for the CS an this completes the proof for Lemma 4.3.1.

Due to triangle inequality [131], $\boldsymbol{w}_1$ and $\boldsymbol{z}_1$ are upper bounded such that $\|col\{{}^i\boldsymbol{w}_1\}_{i=1}^N\|_2^2 + \|{}^c\boldsymbol{w}_1\|_2^2 \geq \|\boldsymbol{w}_1\|_2^2$ and $\|col\{{}^i\boldsymbol{z}_1\}_{i=1}^N\|_2^2 + \|{}^c\boldsymbol{z}_1\|_2^2 \geq \|\boldsymbol{z}_1\|_2^2$. Then rewrite (4.32) as in (4.33), where $\boldsymbol{w}_1$ and $\boldsymbol{z}_1$ are replaced by their upper bounds.

$$\frac{\|\boldsymbol{z}_1\|_2^2}{\|\boldsymbol{w}_1\|_2^2} := \frac{\|col\{{}^i\boldsymbol{z}_1\}_{i=1}^N\|_2^2 + \|\boldsymbol{\Upsilon} col\{{}_c^i\boldsymbol{z}\}_{i=1}^N\|_2^2}{\|col\{{}^i\boldsymbol{w}_1\}_{i=1}^N\|_2^2 + \|col\{{}_c^i\boldsymbol{z}\}_{i=1}^N\|_2^2} \tag{4.33}$$

Recall $\|\boldsymbol{\Upsilon} col\{{}_c^i\boldsymbol{z}\}_{i=1}^N\|_2^2 \leq \|\boldsymbol{\Upsilon}\|_2^2 \|col\{{}_c^i\boldsymbol{z}\}_{i=1}^N\|_2^2$ due to Cauchy-Schwartz inequality [131] and let $\boldsymbol{\Upsilon}$ be a weighted and fully populated adjacency matrix, where $\|\boldsymbol{\Upsilon}\|_2^2 \geq 1$. Then,

$$\begin{aligned} \frac{\|col\{{}^i\boldsymbol{z}_1\}_{i=1}^N\|_2^2 + \|\boldsymbol{\Upsilon} col\{{}_c^i\boldsymbol{z}\}_{i=1}^N\|}{\|col\{{}^i\boldsymbol{w}_1\}_{i=1}^N\|_2^2 + \|col\{{}_c^i\boldsymbol{z}\}_{i=1}^N\|} &\leq \\ \frac{\|col\{{}^i\boldsymbol{z}_1\}_{i=1}^N\|_2^2 + \|\boldsymbol{\Upsilon}\|_2^2 \|col\{{}_c^i\boldsymbol{z}\}_{i=1}^N\|}{\|col\{{}^i\boldsymbol{w}_1\}_{i=1}^N\|_2^2 + \|col\{{}_c^i\boldsymbol{z}\}_{i=1}^N\|} &\leq \\ \frac{\|col\{{}^i\boldsymbol{z}_1\}_{i=1}^N\|_2^2}{\|col\{{}^i\boldsymbol{w}_1\}_{i=1}^N\|_2^2} + \alpha^2 & \end{aligned} \tag{4.34}$$

,where

$$\begin{aligned} \alpha^2 &\leq \|\boldsymbol{\Upsilon}\|_2^2 \\ \frac{\|col\{{}^i\boldsymbol{z}_1\}_{i=1}^N\|_2^2}{\|col\{{}^i\boldsymbol{w}_1\}_{i=1}^N\|_2^2} &\leq \gamma^{*2} \end{aligned} \tag{4.35}$$

Second inequality in (4.35) is the 1^{st} result of Theorem 4.3.2, therefore following $\gamma^{*2} + \alpha^2$ can be related to the performance of the modified system as given in the (4.36) whose inputs and outputs are denoted as $\boldsymbol{w}_1$ and $\boldsymbol{z}_1$, respectively. This completes the proof of Theorem 4.3.2.

$$\frac{\|\boldsymbol{z}_1\|_2^2}{\|\boldsymbol{w}_1\|_2^2} \leq \gamma^{*2} + \alpha^2 \tag{4.36}$$

${}^{c}\boldsymbol{T}_2$ is composed of two classes of non-linearities, which are explicitly revealed in (4.37) and denoted as ${}^{c}\boldsymbol{T}_{2,1}$ and ${}^{c}\boldsymbol{T}_{2,2}$, which are due to the partitions of diagonal and off-diagonal blocks of ${}^{c}\boldsymbol{Z}$.

$$\begin{aligned}
{}^{c}\boldsymbol{T}_2 :=& {}^{c}\boldsymbol{T}_{2,1} + {}^{c}\boldsymbol{T}_{2,2} \\
{}^{c}\boldsymbol{T}_{2,1} =& {}^{c}\boldsymbol{M}_{1,2}^{T} \left[\begin{array}{c|c} \mathbf{0} & {}^{c}\boldsymbol{Z}_{12} \\ \hline {}^{c}\boldsymbol{Z}_{12}^{T} & \mathbf{0} \end{array}\right] {}^{c}\boldsymbol{M}_{1,2} \\
{}^{c}\boldsymbol{T}_{2,2} =& {}^{c}\boldsymbol{M}_{1,2}^{T} \left[\begin{array}{c|c} {}^{c}\boldsymbol{Z}_{11} & \mathbf{0} \\ \hline \mathbf{0} & {}^{c}\boldsymbol{Z}_{22} \end{array}\right] {}^{c}\boldsymbol{M}_{1,2}
\end{aligned} \tag{4.37}$$

Non-linearity due to ${}^{c}\boldsymbol{T}_{2,1}$ is a bilinear matrix function of $\boldsymbol{\Upsilon}$ and ${}^{c}\boldsymbol{Z}_{12}$.

Linear Approximation for ${}^{c}\boldsymbol{T}_{2,1}$: To address non-linearity in ${}^{c}\boldsymbol{T}_{2,1}$, it is convenient to partition ${}^{c}\boldsymbol{M}_{1,2} := \left[\begin{array}{c|c} {}^{c}\boldsymbol{M}_{1,2w}^{T} & {}^{c}\boldsymbol{M}_{1,2z}^{T} \end{array}\right]^{T}$ such that

$$\begin{aligned}
{}^{c}\boldsymbol{M}_{1,2w} &= \left[\begin{array}{ccc} \mathbf{0} & \mathbf{0} & \boldsymbol{I}_{n_w} \end{array}\right] \\
{}^{c}\boldsymbol{M}_{1,2z} &= \left[\begin{array}{ccc} \mathbf{0} & \mathbf{0} & \boldsymbol{\Upsilon} \end{array}\right]
\end{aligned} \tag{4.38}$$

.

Then ${}^{c}\boldsymbol{T}_{2,1}$ can be rewritten as in (4.39), which displays a convex concave pattern as defined in Definition 4.1.2 ,where $\boldsymbol{Q},\ \boldsymbol{Q}_1,\ \boldsymbol{Q}_2 \geq \mathbf{0}$.

$$
\begin{aligned}
{}^{i}\boldsymbol{T}_{2,1} := {}^{i}\boldsymbol{T}_{2,1}(\boldsymbol{Q}_1) - {}^{i}\boldsymbol{T}_{2,1}(\boldsymbol{Q}_2) \;=\; \\
\left[\begin{array}{c} {}^{i}\boldsymbol{M}_{1,2w} \\ \hline {}^{i}\boldsymbol{M}_{1,2z} \end{array}\right]^{T} \left[\begin{array}{c|c} \boldsymbol{0} & {}^{i}\boldsymbol{Z}_{12} \\ \hline {}^{i}\boldsymbol{Z}_{12}^{T} & \boldsymbol{0} \end{array}\right] \left[\begin{array}{c} {}^{i}\boldsymbol{M}_{1,2w} \\ \hline {}^{i}\boldsymbol{M}_{1,2z} \end{array}\right] \;= \\
\left[\begin{array}{c} {}^{i}\boldsymbol{Z}_{12}^{T}\,{}^{i}\boldsymbol{M}_{1,2w} \\ \hline {}^{i}\boldsymbol{M}_{1,2z} \end{array}\right]^{T} \left[\begin{array}{c|c} \boldsymbol{0} & \boldsymbol{I} \\ \hline \boldsymbol{I} & \boldsymbol{0} \end{array}\right] \left[\begin{array}{c} {}^{i}\boldsymbol{Z}_{12}^{T}\,{}^{i}\boldsymbol{M}_{1,2w} \\ \hline {}^{i}\boldsymbol{M}_{1,2z} \end{array}\right] \;= \\
\left[\begin{array}{c} {}^{i}\boldsymbol{Z}_{12}^{T}\,{}^{i}\boldsymbol{M}_{1,2w} \\ \hline {}^{i}\boldsymbol{M}_{1,2z} \end{array}\right]^{T} \boldsymbol{V}(\boldsymbol{Q}_1 - \boldsymbol{Q}_2)\boldsymbol{V}^{T} \left[\begin{array}{c} {}^{i}\boldsymbol{Z}_{12}^{T}\,{}^{i}\boldsymbol{M}_{1,2w} \\ \hline {}^{i}\boldsymbol{M}_{1,2z} \end{array}\right]
\end{aligned}
\tag{4.39}
$$

The decomposition for $\boldsymbol{Q}$, which is given in (4.40) uses generalized left eigenvectors defined as $\boldsymbol{V}$ and eigenvalues that are diagonally blocked in $\boldsymbol{\lambda} := diag(-\boldsymbol{I}, \boldsymbol{I})$ with respective sizes [123]. When the diagonal eigenvalues are separated into two matrices while securing the dimension of the original matrix, equation of (4.40) is obtained.It should be noted that, convex-concave decomposition is not unique and depending on the selection curvature can shift between convex and concave partitions, as denoted in [122].

$$
\boldsymbol{Q} \;=\; \boldsymbol{V}(\boldsymbol{Q}_1 - \boldsymbol{Q}_2)\boldsymbol{V}^{T} \;=\; \boldsymbol{V}\Big(\begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} - \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}\Big)\boldsymbol{V}^{T} \tag{4.40}
$$

If ${}^{c}\boldsymbol{T}_{2,1}(\boldsymbol{Q}_2)$ partition of the decomposition provided in (4.39) is linearized as given in (4.41) and denoted as $L\{{}^{c}\boldsymbol{T}_{2,1}(\boldsymbol{Q}_2)\}$, then ${}^{c}\boldsymbol{T}_{2,1}(\boldsymbol{Q}_2) \geq L\{{}^{c}\boldsymbol{T}_{2,1}(\boldsymbol{Q}_2)\}_k$ therefore ${}^{c}\boldsymbol{T}_{2,1} \geq {}^{c}\boldsymbol{T}_{2,1}(\boldsymbol{Q}_1) - L\{L\{{}^{c}\boldsymbol{T}_{2,1}(\boldsymbol{Q}_2)\}_k\}$ is true due to due to Lemma 4.1.1 and Lemma 4.1.2, where k in $L\{\cdot\}_k$ represents the linearization around existing solution. $\{\cdot\}^k$ in (4.41) represents existing solutions of the decision variables.

$$
\begin{aligned}
L\{{}^{c}\boldsymbol{T}_{2,1}(\boldsymbol{Q}_2)\}_k \;=\; & \left[\frac{{}^{c}\boldsymbol{Z}_{12}^{k\,T}{}^{c}\boldsymbol{M}_{1,2w}^{k}}{{}^{c}\boldsymbol{M}_{1,2z}^{k}}\right]^{T}\boldsymbol{Q}_2\left[\frac{{}^{c}\boldsymbol{Z}_{12}^{T}{}^{c}\boldsymbol{M}_{1,2w}}{{}^{c}\boldsymbol{M}_{1,2z}}\right]+ \\
& \left[\frac{{}^{c}\boldsymbol{Z}_{12}^{T}{}^{c}\boldsymbol{M}_{1,2w}}{{}^{c}\boldsymbol{M}_{1,2z}}\right]^{T}\boldsymbol{Q}_2\left[\frac{{}^{c}\boldsymbol{Z}_{12}^{k\,T}{}^{c}\boldsymbol{M}_{1,2w}^{k}}{{}^{c}\boldsymbol{M}_{1,2z}^{k}}\right]- \\
& \left[\frac{{}^{c}\boldsymbol{Z}_{12}^{k\,T}{}^{c}\boldsymbol{M}_{1,2w}^{k}}{{}^{c}\boldsymbol{M}_{1,2z}^{k}}\right]^{T}\boldsymbol{Q}_2\left[\frac{{}^{c}\boldsymbol{Z}_{12}^{k\,T}{}^{c}\boldsymbol{M}_{1,2w}^{k}}{{}^{c}\boldsymbol{M}_{1,2z}^{k}}\right]
\end{aligned}
\tag{4.41}
$$

Linear Approximation for ${}^{c}\boldsymbol{T}_{2,2}$: Let $L\{{}^{c}\boldsymbol{T}_{2,2}\}$ denote linear approximation of ${}^{c}\boldsymbol{T}_{2,2}$. ${}^{c}\boldsymbol{Z}_{11}$ and ${}^{c}\boldsymbol{Z}_{22}$ dictate in and out supply rate to ${}^{c}\boldsymbol{G}$. Therefore an intuitive selection for ${}^{c}\boldsymbol{Z}_{11}$ and ${}^{c}\boldsymbol{Z}_{22}$ with a priori definiteness assumption can be ${}^{c}\boldsymbol{Z}_{11} = {}^{c}\boldsymbol{Z}_{11}^{T} \leq \boldsymbol{0}$ and ${}^{c}\boldsymbol{Z}_{22} = {}^{i}\boldsymbol{Z}_{22}^{T} \geq \boldsymbol{0}$, respectively.

Let $\eta_\gamma := \frac{1}{\gamma^2}\boldsymbol{I}_{n_z}$ and $\epsilon := \{\eta_\gamma\}_{\gamma=\gamma^0}^{\gamma_{optimal}}$, where $\gamma^0 \geq \gamma \geq \gamma_{optimal} \geq 0$. Then ϵ is non-decreasing sequence and ${}^{c}\boldsymbol{Z}_{22} \geq \limsup_{\gamma\to\gamma_{optimal}} \eta_\gamma \geq 0$. Then ${}^{c}\boldsymbol{T}_{2,2} \geq L\{{}^{c}\boldsymbol{T}_{2,2}\}$, therefore $L\{{}^{c}\boldsymbol{T}_{2,2}\}$ as given in (4.42) along with $L\{{}^{c}\boldsymbol{T}_{2,2}\}$ as given in (4.41), results in

$$
L\{{}^{c}\boldsymbol{T}_{2,2}\} := {}^{c}\boldsymbol{M}_{1,2}^{T}\left[\begin{array}{c|c} {}^{c}\tilde{\boldsymbol{Z}}_{11} & \boldsymbol{0} \\ \hline \boldsymbol{0} & {}^{c}\tilde{\boldsymbol{Z}}_{22} \end{array}\right]{}^{c}\boldsymbol{M}_{1,2} \tag{4.42}
$$

, where ${}^{c}\tilde{\boldsymbol{Z}}_{11} = {}^{c}\boldsymbol{Z}_{11}$ and ${}^{c}\tilde{\boldsymbol{Z}}_{22} = \frac{1}{{}^{c}\gamma^2}\boldsymbol{I}_{n_z}$. This represents successive linearization as provided in [132].

Finally, linear approximation $L\{{}^{c}\boldsymbol{T}_2\}$ is provided in (4.43) will satisfy inequality given in (4.44) and with that problem given in (4.20) can be written as in (4.45).

$$
L\{{}^{c}\boldsymbol{T}_2\} = L\{{}^{c}\boldsymbol{T}_{2,1}\} + L\{{}^{c}\boldsymbol{T}_{2,2}\} \tag{4.43}
$$

$$
{}^{c}\boldsymbol{T}_2 \geq L\{{}^{c}\boldsymbol{T}_{2,1}\} + L\{{}^{c}\boldsymbol{T}_{2,2}\} \tag{4.44}
$$

Given the linear approximations $L\{{}^{i}\boldsymbol{T}_{2,1}\}$, the distributed nominal H_∞ edge weight synthesis problem in (4.20) can be rewritten as given in (4.45). It should be noted that ${}^{c}\boldsymbol{T}_2$ is linear when $\boldsymbol{\Upsilon}$ is given, which is the 1^{st} phase in the Algorithm 4.

$$\begin{aligned} \text{minimize} \quad & \gamma \\ \text{s. t.} \quad & {}^{i}\boldsymbol{X} && \geq \mathbf{0} \\ & {}^{i}\boldsymbol{T}_1 + {}^{i}\boldsymbol{T}_2 && \leq \mathbf{0} \\ & {}^{c}\boldsymbol{T}_1 + L\{{}^{c}\boldsymbol{T}_2\} && \leq \mathbf{0} \end{aligned} \tag{4.45}$$

Algorithm 4 Distributed H_∞ optimization algorithm

Require: $\boldsymbol{\Upsilon}^0$

Ensure: $\boldsymbol{\Upsilon}^0$ is within the relative interior [122].

Phase 1 - Finding relative interior

Solve (4.45) as a linear program since $\boldsymbol{\Upsilon}^0$ is given.

Result: γ^0, $\boldsymbol{X}^0$, ${}^i\boldsymbol{Z}^0$, ${}^c\boldsymbol{Z}^0$.

Phase 2 - Successive minimization

Let m be the iteration step of **Phase 2** and m_{max} be the maximum number of iterations.

Set $\boldsymbol{\Upsilon} \rightarrow \boldsymbol{\Upsilon}^{m=1}$, ${}^c\boldsymbol{Z}^0 \rightarrow {}^c\boldsymbol{Z}^{m=1}$

Define $\underline{\gamma}^m$, $\overline{\gamma}^m$ to be lower and upper bound to γ^m. Define $\gamma^m = (\overline{\gamma}^m + \underline{\gamma}^m)/2$ and set $\underline{\gamma}^{m=1} = \gamma^0 - E$, $\overline{\gamma}^{m=1} = \gamma^0 + E$, where $E > 0 \in \mathbb{R}$.

loop 1: Successive Minimization

  while $m < m_{max}$ & $\overline{\gamma}^m - \underline{\gamma}^m > tolerance$ **do**

    Solve **loop 2**

    if loop 2 is feasible **then**

      Set $\overline{\gamma}^{m+1} = \gamma^m$

    else

      Set $\underline{\gamma}^{m+1} = \gamma^m$

    end if

    Set $m = m + 1$

  end while

end loop

loop 2: Sequential Optimization

  Let k be iteration step of **loop 2** and k_{max} be the maximum number of iterations.

  Set $\boldsymbol{\Upsilon}^{k=1} = \boldsymbol{\Upsilon}^m$, ${}^c\boldsymbol{Z}^{k=1} = {}^c\boldsymbol{Z}^m$

  while $k < k_{max}$ & $\|\boldsymbol{\Upsilon}^k\|_2^2 - \|\boldsymbol{\Upsilon}^{k-1}\|_2^2 > tolerance$ **do**

    Solve (4.45) such that $\boldsymbol{\Upsilon}^k$, ${}^c\boldsymbol{Z}^k$ are the current values(k=1) and $\boldsymbol{\Upsilon}$ and ${}^c\boldsymbol{Z}$ are variables.

    Set $\boldsymbol{\Upsilon}^{k+1} = \boldsymbol{\Upsilon}$, ${}^c\boldsymbol{Z}^{k+1} = {}^c\boldsymbol{Z}$ of k^{th} iteration.

    Set $k = k + 1$

  end while

  Result: $\gamma^{k_{max}}$, $\boldsymbol{X}^{k_{max}}$, ${}^i\boldsymbol{Z}^{k_{max}}$, ${}^c\boldsymbol{Z}^{k_{max}}$ and $\boldsymbol{\Upsilon}^{k_{max}}$.

end loop

4.4 Verification

Methodology described under Section 4.3, is verified using a CS of six vehicles. Adjacency matrix for lumped and distributed cases are compared in terms of edge weights Υ and the robust performance level given by γ. Simulations are executed on a PC with Intel(R) Core(TM) i7-4720HQ CPU @2.6GHz, 16GB RAM running on Windows 10 OS and MATLAB 2019b.

Dynamics of the agents are defined with the state space representation given in (4.46).

$$ {}^{i}\boldsymbol{G} = \left[\begin{array}{cc|cc} 0 & 2 & 0 & 0 \\ -0.5 & -3 & 1 & 1 \\ \hline 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{array}\right] \tag{4.46} $$

Agents are connected to each other with an adjacency matrix, $\boldsymbol{\Upsilon}$, which is fully populated such that edge weights ν_{ij} for $i, j = [1, \cdots, N]$ and $i \neq j$ are defined by a uniformly distributed random number in the interval (0,1). Constructing edge weights in this form makes row sum of $\boldsymbol{\Upsilon} \geq 0$. Therefore before implementing the algorithm, a weighted $\boldsymbol{\Upsilon}$ is calculated. As a result, of that initial adjacency matrix is provided in (4.47).

$$ \boldsymbol{\Upsilon}_0 = \begin{bmatrix} 0 & 0.4546 & 0.2472 & 0.0530 & 0.1555 & 0.0897 \\ 0.1089 & 0 & 0.1705 & 0.2042 & 0.1772 & 0.3392 \\ 0.1571 & 0.2861 & 0 & 0.1934 & 0.2904 & 0.0730 \\ 0.2817 & 0.1204 & 0.2799 & 0 & 0.2896 & 0.0283 \\ 0.1091 & 0.0959 & 0.2859 & 0.3615 & 0 & 0.1475 \\ 0.3184 & 0.2755 & 0.0027 & 0.2456 & 0.1578 & 0 \end{bmatrix} \tag{4.47} $$

Using (4.46) as ${}^{i}\boldsymbol{G}$, a CS of six agents ($N = 6$) is created. Solver starts from the initial guess provided in (4.47).$\boldsymbol{\Upsilon}^0$ being the initial adjacency matrix for the Algorithm 4, $\boldsymbol{\Upsilon}^\infty$ is synthesized and given in (4.48). Initial robustness of the CS is calculated to be $\gamma = 2.05$ for $\boldsymbol{\Upsilon}^0$. CS with $\boldsymbol{\Upsilon}^\infty$ on the other hand provide a robustness $\gamma = 1.58$, which clearly represents an improvement in nominal H_∞ performance. Algorithm 4 has a mean calculation time of $t_\mu = 53.1028$s with a standard deviation of $\sigma = 12.6443$s for 100 runs, where $\boldsymbol{\Upsilon}^0$ is fully populated and constructed from uniformly distributed random edge weights every time. It should be noted that, topology is not synthesized and can be defined arbitrarily.

$$\boldsymbol{\Upsilon}_\infty = \begin{bmatrix} 0 & 0.9167 & 0.0431 & 0.0121 & 0.0124 & 0.0158 \\ 0.0117 & 0 & 0.0091 & 0.0404 & 0.0129 & 0.9259 \\ 0.0213 & 0.0935 & 0 & 0.0419 & 0.8313 & 0.0121 \\ 0.1156 & 0.0386 & 0.2211 & 0 & 0.6191 & 0.0056 \\ 0.0056 & 0.0138 & 0.5910 & 0.3791 & 0 & 0.0106 \\ 0.8617 & 0.0732 & 0.0000 & 0.0553 & 0.0097 & 0 \end{bmatrix} \tag{4.48}$$

$\boldsymbol{\Upsilon}^0$ and $\boldsymbol{\Upsilon}^\infty$ are provided in Figure 4.3 with the **black** and **red** color code. In this plot edge weights are provided, where thickness of edges are modified based on maximum edge weight of each adjacency matrix and only provided to visualize intensity of the edge ν_{ji} among the edge set of regarding adjacency matrix. Verification runs show that, adjacency matrices that resembles complete cycles are turned into directed path and cycles by minimizing weights on certain edges almost zero. In the example given in Figure 4.3, algorithm optimizes edge weights such that **Agents 1, 2 and 6** and **Agent 3, 4 and 5** become separate groups with high strength edge weights while edge weights between these groups are weakened. The only considerable weighting exist between **Agent 1 and Agent 4**, **Agent 6 and Agent 4** and **Agent 2 and Agent 3**.

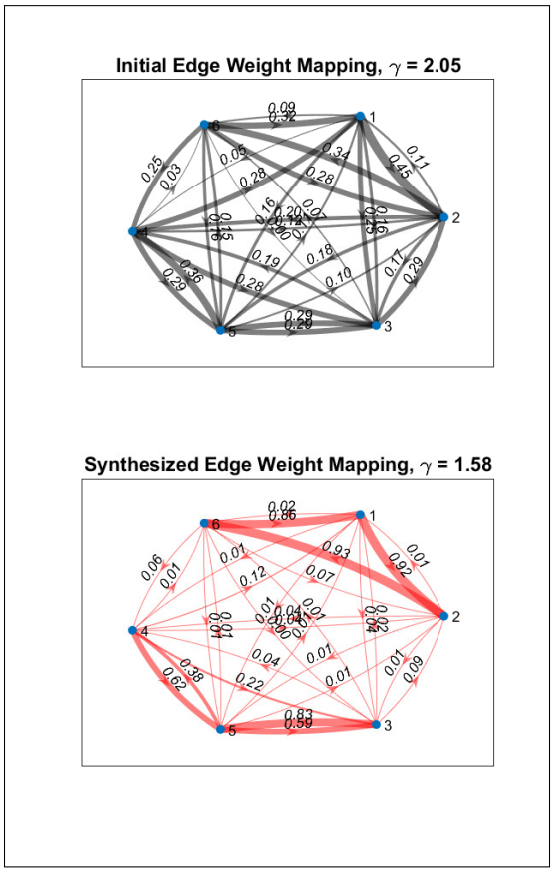


Figure 4.3. Comparison of the initial and synthesized adjacency matrices in a Graph plot. Initial and synthesized adjacency matrices $\boldsymbol{\Upsilon}^0$ and $\boldsymbol{\Upsilon}^\infty$ are provided in (4.47) and (4.48), respectively..

The time performance of these agents are verified by injecting disturbances at different nodes of a CS which is expected to have consensus. It should be noted that no controller is assigned to this CS for tracking and CS is constructed using the lumped model given in (4.6). The disturbances are calculated such that

$$ {}^i\Delta(t, {}^i\boldsymbol{z}_1) = \begin{cases} \underline{k}^i \boldsymbol{z}_1, & if \;\; t < {}^iT_1 < {}^iT_2 \\ \overline{k}^i \boldsymbol{z}_1, & if \;\; t > {}^iT_2 \end{cases} \tag{4.49} $$

, where $\underline{k}$ and $\overline{k}$ are defined as rectangular pulses with magnitude of -1 and 1 with duration of 0.5 s. The visualization of ${}^i\boldsymbol{\Delta}$ are provided in Figure 4.4. As illustrated in Figure 4.4,

perturbations are given until $t = 9$ s to every agent. Selection of $\underline{k}$, $\overline{k}$, ${}^{i}T_1$ and ${}^{i}T_2$ are arbitrary.

For the agent dynamics given in (4.46) and ${}^{i}\boldsymbol{\Delta}$ CS system is constructed with $\boldsymbol{\Upsilon}^0$ and $\boldsymbol{\Upsilon}^\infty$ and denoted as H_0 and H_∞.

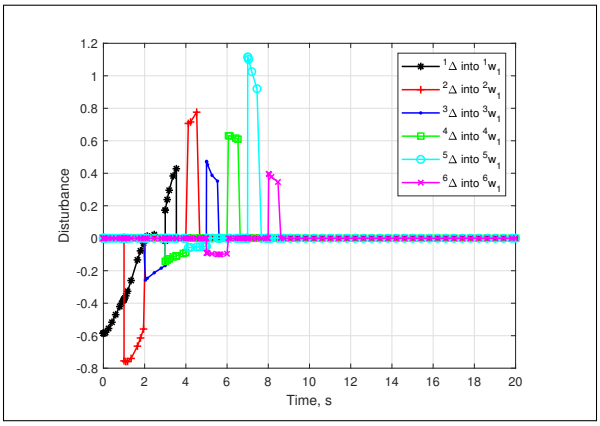


Figure 4.4. Inejcted disturbances ${}^{i}\Delta$ to the performance channel through ${}^{i}\boldsymbol{w}_1$..

The performance of H_0 and H_∞ are provided in Figure 4.5, where dashed red and solid black lines illustrate ${}^{i}\boldsymbol{z}_1$ of agents within H_0 and H_∞, respectively. Markers denoting each agent are provided in the plot for further elaboration. Figure 4.5 illustrates that injected disturbances are attenuated faster at every node in H_∞.

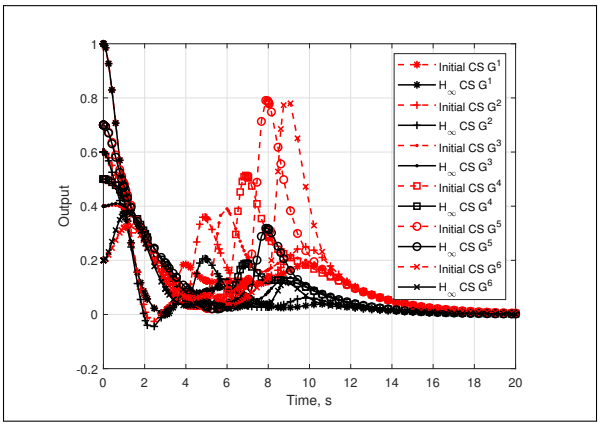


Figure 4.5. Initial and synthesized out-degrees of agent ${}^{i}\boldsymbol{G}$ denoted as ν_{ji}..

4.5 Conclusions

This paper focuses on finding the optimal values for edge weights in terms of enhancing H_∞ performance criteria of a CS in a distributed fashion. This problem is solved by modeling communication media as a synthetic agent denoted as ${}^{c}\boldsymbol{G}$ and making ideal interconnections between original agents and this synthetic agent in [73]. By this way, information on adjacency matrix is secured and distributed edge weight synthesis is executed in a distributed fashion. Yet This work improves the deficiencies in original methodology by minimizing the ${}^{i}\boldsymbol{Z}_{22}$ definition by finding a supremum to it with a successively magnifying quadratic term defined as $\gamma \boldsymbol{I}_{n_z}$ as given in **linear approximation of** ${}^{c}\boldsymbol{T}_{2,2}$ in (4.42). Here, a non-decreasing sequence is defined by γ which is being minimized. The off-diagonal terms in ${}^{i}\boldsymbol{Z}$ are given a linear approximation in **linear approximation of** ${}^{c}\boldsymbol{T}_{2,1}$ and this relies on convex-concave decomposition method. Finally, $L\{{}^{c}\boldsymbol{T}_{2,1}\}$ and $L\{{}^{c}\boldsymbol{T}_{2,2}\}$ are iterated in different layers of the optimization, where $L\{{}^{c}\boldsymbol{T}_{2,1}\}$ is being optimized in **loop 2**, whose steps are denoted with k, while $L\{{}^{c}\boldsymbol{T}_{2,2}\}$ is being optimized in **loop 1**, where γ is being minimized.

Chapter 5

Cooperative Model Predictive Control Strategy for Docking With Task Prioritization

5.1 Preliminaries

The notation of this work is as follows. $s \in \mathbb{R}$, $\boldsymbol{v} \in \mathbb{R}^{n_v}$ and $\boldsymbol{M} \in \mathbb{R}^{n_r \times n_c}$ represent arbitrary scalar, vector and matrix. $\mathcal{F}_I$ and $\mathcal{F}_B$ represent inertial and body frames, where the expression of a vector for these frames is written as ${}_i\boldsymbol{v},\ i = I, B$. Unit vectors in orthonormal frames are denoted as $\boldsymbol{u}_1 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$, $\boldsymbol{u}_2 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T$ and $\boldsymbol{u}_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$. Composite rotation matrix ${}_I\boldsymbol{C}_B \in SO(3)$ is written from $\mathcal{F}_B$ to $\mathcal{F}_I$ and rotation sequence is depicted as $z - y - x$ and associated Euler angles are denoted as ψ, θ and ϕ. $g = 9.81\ m/s^2$ is the gravitational acceleration. $col\{\cdot\}_{m=1}^{M}$, $row\{\cdot\}_{l=1}^{M}$ and $diag\{\cdot\}_{m=1}^{M}$ represents column, row and diagonal concatenation of the entity within the parenthesis.

5.1.1 Quadrotor Dynamics

A 6 Degree of Freedom (DoF) rigid body dynamics with mass (m_q) and moment of inertia ($\boldsymbol{J}_q$) express the dynamics of the quadcopter, as in (5.1). Model states are denoted as $\boldsymbol{x}_q = \begin{bmatrix} \boldsymbol{p}_q^T & \boldsymbol{v}_q^T & \boldsymbol{\theta}_q^T & \boldsymbol{w}_q^T \end{bmatrix}^T \in \mathbb{R}^{12}$, where $\boldsymbol{p}_q = {}_I\boldsymbol{p}_q \in \mathbb{R}^3$, $\boldsymbol{v}_q = {}_I\boldsymbol{v}_q \in \mathbb{R}^3$, $\boldsymbol{\theta}_q \in \mathbb{R}^3$ and $\boldsymbol{w}_q = {}_B\boldsymbol{w}_q \in \mathbb{R}^3$ are inertial position, inertial velocity, Euler angles and angular velocity, respectively. Given the Euler angles rotation matrix for quadrotor is denoted as ${}_I\boldsymbol{C}_q(\boldsymbol{\theta}_q) = {}_I\boldsymbol{C}_q$. Inputs of the system are denoted as $\boldsymbol{u}_q = \begin{bmatrix} f_q & \boldsymbol{t}_q^T \end{bmatrix}^T \in \mathbb{R}^4$, where f_q is the total thrust generated by the motors on $\mathcal{F}_B$ and $\boldsymbol{t}_q \in \mathbb{R}^3$ is the column matrix of moments generated on the body defined in $\mathcal{F}_B$. $\boldsymbol{E}_q$ is the mapping between $\boldsymbol{w}_q$ and $\dot{\boldsymbol{\theta}}_q$ such

that $\boldsymbol{w}_q = \boldsymbol{E}_q\dot{\boldsymbol{\theta}}_q$ for predefined rotation sequence. Quadcopter properties are taken from the work [133].

$$\begin{aligned}\dot{\boldsymbol{x}}_q &= \begin{bmatrix} \dot{\boldsymbol{p}}_q \\ \dot{\boldsymbol{v}}_q \\ \dot{\boldsymbol{e}}_q \\ \dot{\boldsymbol{w}}_q \end{bmatrix} = \boldsymbol{f}(\boldsymbol{x}_q, \boldsymbol{u}_q) \\ &= \begin{bmatrix} \boldsymbol{v}_q \\ g\boldsymbol{u}_3 - (1/m_q){}_I\boldsymbol{C}_q f_q \boldsymbol{u}_3 \\ \boldsymbol{E}^{-1}\boldsymbol{w}_q \\ \boldsymbol{J}_q^{-1}(-\boldsymbol{w}_q \times \boldsymbol{J}_q\boldsymbol{w}_q + \boldsymbol{t}_q) \end{bmatrix}\end{aligned} \tag{5.1}$$

5.1.2 Rover Dynamics

Dynamics of the rover is calculated with mass m_r and moment of inertia $\boldsymbol{J}_r$ assuming that it runs on a flat surface and provided in (5.2). Based on this assumption, model states are denoted as $\boldsymbol{x}_r = [p_{r,x}\ p_{r,y}\ \Psi_r\ v_{r,x}\ w_{r,z}]^T$, where $p_{r,x}$ and $p_{r,y}$ represents inertial position on $x-y$ plane, Ψ_r is the Euler angle about z axis, ${}_Bv_{r,x} = v_{r,x}$ is the horizontal velocity of the rover and ${}_Bw_{r,z} = w_{r,z}$ is the angular velocity of the rover. Composite rotation matrix for the is denoted as ${}_I\boldsymbol{C}_r = {}_I\boldsymbol{C}_r(\Psi_r)$. Inputs to the system are denoted as $\boldsymbol{u}_r = [f_{r,1}\ f_{r,2}]^T$, which represents the traction forces applied on the surface by a set of the wheel on the right and the left-hand side of the rover, respectively. m_r is taken as 1 kg and and $\boldsymbol{J}_r = diag\{0.1\}\ kg \cdot m^2$.

$$
\begin{aligned}
\dot{\boldsymbol{x}}_r \;&=\; \begin{bmatrix} \dot{p}_{r,x} \\ \dot{p}_{r,y} \\ \dot{\Psi}_r \\ \dot{v}_{r,x} \\ \dot{w}_{r,z} \end{bmatrix} \;=\; \boldsymbol{f}(\boldsymbol{x}_r, \boldsymbol{u}_r) \\
&= \begin{bmatrix} \boldsymbol{u}_1^T {}_I\boldsymbol{C}_r v_{r,x} \boldsymbol{u}_1 \\ \boldsymbol{u}_2^T {}_I\boldsymbol{C}_r v_{r,x} \boldsymbol{u}_1 \\ w_{r,z} \\ (1/m_r)(f_{r,1} + f_{r,2}) \\ \boldsymbol{J}^{-1}(\boldsymbol{r}_1 \times f_{r,1} + \boldsymbol{r}_2 \times f_{r,2}) \end{bmatrix}
\end{aligned}
\tag{5.2}
$$

5.1.3 Underlying Graph Structure

The communication among agents in the cooperative systems are described by Graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, which consists of node set $\mathcal{N}$ and edge set $\mathcal{E}$ [134]. Edge set $\mathcal{E} \subset \mathcal{N} \times \mathcal{N}$ is given between nodes $i \in \mathcal{N}$ and $j \in \mathcal{N}$ such that $(j, i) \in \mathcal{E}$ denotes node i receives information from j. Let n_w be an arbitrary signal dimension, then adjacency matrix[1] $\boldsymbol{A} = [a_{ij}] \otimes \boldsymbol{I}_{n_w} \in \mathbb{R}^{N \cdot n_w \times N \cdot n_w}$ of $\mathcal{G}$ is composed of weighting scalars a_{ij}, where a_{ij} quantifies the strength of the connection from node j to node i. N is the number of agents in the cooperative system (CS). Formally, A_{ij} is described as in the following equation.

$$
\boldsymbol{A}_{ij} = \begin{cases} a_{ij} > 0, \quad j \neq i, \;\; (j, i) \in \mathcal{E} \\ a_{ij} = 0, \quad otherwise \end{cases}
\tag{5.3}
$$

[1] Shared signal sizes among agents are assumed to be identical and equal to n_w.

5.2 Methodology

This section introduces a unified MPC strategy to maintain a docking approach for long-range and a finer docking maneuver in short-range by accommodating a non-linear and a linear MPC designed with edge weight information and task prioritization. This section is divided into four sections where Non-linear MPC (NMPC), Linear MPC (LMPC), cooperative task prioritization, and implementation of the control strategy is described in subsections section 5.2.1, section 5.2.2, section 5.2.3 and section 5.2.4, respectively.

Introduced method can be applied on all of the agents as formulation only considers local neighbor information therefore formulations will be provided for agent denoted as i, which is defined by the states $\boldsymbol{x}_i(t) \in \mathbb{R}^{n_x}$, inputs $\boldsymbol{u}_i(t) \in \mathbb{R}^{n_u}$ and outputs $\boldsymbol{y}_i(t) \in \mathbb{R}^{n_y}$. Given the states and inputs, non-linear dynamics of agent i are given in the form:

$$\begin{aligned} \dot{\boldsymbol{x}}_i(t) &= \boldsymbol{f}_i(\boldsymbol{x}_i(t),\ \boldsymbol{u}_i(t)) \\ \boldsymbol{y}_i &= \boldsymbol{h}_i(\boldsymbol{x}_i(t),\ \boldsymbol{u}_i(t)) \end{aligned} \tag{5.4}$$

Equation (5.5) provides the discrete linear representation of the agents' dynamics. This is obtained by linearizing about current state and input, which is denoted as $(\boldsymbol{x}_{i,c},\ \boldsymbol{u}_{i,c})$ and then by converting the continuous time system in discrete system using Euler discretization (see [135]).

$$\begin{aligned} \boldsymbol{\Delta x}_{i,k+1} &= \boldsymbol{A}_i \boldsymbol{\Delta x}_{i,k} + \boldsymbol{B}_i \boldsymbol{\Delta u}_{i,k} \\ \boldsymbol{\Delta y}_{i,k} &= \boldsymbol{C}_i \boldsymbol{\Delta x}_{i,k} + \boldsymbol{D}_i \boldsymbol{\Delta u}_{i,k} \end{aligned} \tag{5.5}$$

where k is the current sample such that, $\boldsymbol{\Delta x}_{i,k} = \boldsymbol{x}_{i,k} - \boldsymbol{x}_{i,c}$ and $\boldsymbol{\Delta u}_{i,k} = \boldsymbol{u}_{i,k} - \boldsymbol{u}_{i,c}$. In this work, we assumed full state information is shared among the agents. Therefore $\boldsymbol{C}_i$ and $\boldsymbol{D}_i$ matrices are assumed to be $\boldsymbol{I}$ and $\boldsymbol{0}$, respectively, with compatible sizes.

5.2.1 Non-linear MPC Formulation

Equation (5.6) represents the non-linear problem that runs in NMPC where $\mathcal{L}_i(\boldsymbol{x}_{i,k},\ \boldsymbol{u}_{i,k},\ \boldsymbol{x}_{i,ref})$ is the objective function, and $\mathcal{C}_{eq}(\boldsymbol{x}_{i,k},\ \boldsymbol{u}_{i,k})$ and $\mathcal{C}_{ineq}(\boldsymbol{x}_{i,k},\ \boldsymbol{u}_{i,k})$ are equality and inequality constraints, respectively. M in (5.6) denotes the prediction horizon for the NMPC.

$$\begin{aligned} \underset{\boldsymbol{x}_{i,k},\boldsymbol{u}_{i,k}}{\text{minimize}} \quad & \sum_{k=1}^{M} \mathcal{L}_i(\boldsymbol{x}_{i,k},\ \boldsymbol{u}_{i,k},\ \boldsymbol{x}_{i,ref}) \\ \text{s. t.} \quad & \dot{\boldsymbol{x}}_i(t) - \boldsymbol{f}_i(\boldsymbol{x}_i(t),\ \boldsymbol{u}_i(t)) \ =\ \mathbf{0} \\ & \mathcal{C}_{eq}(\boldsymbol{x}_{i,k},\ \boldsymbol{u}_{i,k}) \ =\ \mathbf{0} \\ & \mathcal{C}_{ineq}(\boldsymbol{x}_{i,k},\ \boldsymbol{u}_{i,k}) \ \leq\ \mathbf{0} \end{aligned} \tag{5.6}$$

The objective function implemented in this work has the following quadratic form.

$$\begin{aligned} & \mathcal{L}_i(\boldsymbol{x}_{i,k},\ \boldsymbol{u}_{i,k},\ \boldsymbol{x}_{i,ref}) \ = \\ & \sum_{k=1}^{M} \mathcal{L}_{i,tp}(\boldsymbol{x}_{i,ref},\ \boldsymbol{x}_{i,k}) + \boldsymbol{u}_{i,k}^T \boldsymbol{R} \boldsymbol{u}_{i,k} + \boldsymbol{x}_{i,k}^T \boldsymbol{Q} \boldsymbol{x}_{i,k} \end{aligned} \tag{5.7}$$

where $\boldsymbol{Q} \in \mathbb{R}^{n_x \times n_x}$ and $\boldsymbol{R} \in \mathbb{R}^{n_u \times n_u}$ are square matrices. This objective function drives system to achieve the cooperative task defined by $\mathcal{L}_{i,tp}(\boldsymbol{x}_{i,ref},\ \boldsymbol{x}_{i,k})$, which brings a nuance cooperativeness with task prioritization to the tracking and it is explained in section 5.2.3.

The equality and inequality constraints serve the following purposes. $\mathcal{C}_{eq}(\boldsymbol{x}_{i,k},\ \boldsymbol{u}_{i,k})$ is introduced to ensure initial conditions, which are $\boldsymbol{x}_{i,c} \ =\ \boldsymbol{x}_i(t=0)$ and $\boldsymbol{u}_{i,c} \ =\ \boldsymbol{u}_i(t=0)$, and $\mathcal{C}_{ineq}(\boldsymbol{x}_{i,k},\ \boldsymbol{u}_{i,k})$ is introduced to enforce states and inputs to stay in predefined ranges. These constraints are provided in [133].

5.2.2 Linear MPC Formulation

The LMPC method implemented in this work is implicit LMPC applied in [87, 133], and details can be found therein. Therefore the method is summarized here for completeness. The quadratic problem for the LMPC is provided in (5.8).

$$\begin{aligned} \underset{\boldsymbol{u}_{i,K}}{\text{minimize}} \quad & \overline{\mathcal{L}}_i(\boldsymbol{x}_{i,K},\ \boldsymbol{\Delta u}_{i,K},\ \boldsymbol{x}_{i,ref}) \\ & \overline{\mathcal{C}}_{eq}(\boldsymbol{x}_{i,K},\ \boldsymbol{u}_{i,K}) \ = \ \mathbf{0} \\ & \overline{\mathcal{C}}_{ineq}(\boldsymbol{x}_{i,K},\ \boldsymbol{u}_{i,K}) \ \leq \ \mathbf{0} \end{aligned} \tag{5.8}$$

The quadratic objective function is denoted as $\overline{\mathcal{L}}_i(\boldsymbol{x}_{i,K},\ \boldsymbol{\Delta u}_{i,K},\ \boldsymbol{x}_{i,ref})$ and given in (5.9), where decision variable is $\boldsymbol{\Delta u}_{i,K}$ and $\boldsymbol{x}_{i,K}$ is a function of $\boldsymbol{\Delta u}_{i,K}$.

$$\begin{aligned} & \overline{\mathcal{L}}_i(\boldsymbol{x}_{i,K},\ \boldsymbol{\Delta u}_{i,K},\ \boldsymbol{x}_{i,ref}) \ = \\ & \overline{\mathcal{L}}_{i,tp}(\boldsymbol{x}_{i,ref},\ \boldsymbol{x}_{i,K}) + \boldsymbol{u}_{i,K}^T \overline{\boldsymbol{R}} \boldsymbol{u}_{i,K} + \boldsymbol{x}_{i,K}^T \overline{\boldsymbol{Q}} \boldsymbol{x}_{i,K} \end{aligned} \tag{5.9}$$

where $\overline{\boldsymbol{R}} \ = \ diag\{\boldsymbol{R}\}_1^M$, $\overline{\boldsymbol{Q}} \ = \ diag\{\boldsymbol{Q}\}_1^M$ and $\overline{\mathcal{L}}_{i,tp}(\boldsymbol{x}_{i,ref},\ \boldsymbol{x}_{i,K})$ is calculated in section 5.2.3.

Due to the implicit formulation, state predictions are described with respect to the decision variables and combined in a lumped representation. Similarly, decision variables, which are the inputs, are also collected under a lumped term. These lumped states and inputs are denoted as $\boldsymbol{\Delta x}_{i,K}$ and $\boldsymbol{\Delta u}_{i,K}$ and provided in (5.10).

$$\begin{aligned} \boldsymbol{\Delta x}_{i,K} \ &= \ \begin{bmatrix} \boldsymbol{\Delta x}_{i,k} \\ \vdots \\ \boldsymbol{\Delta x}_{i,k+M-1} \end{bmatrix} \\ \boldsymbol{\Delta u}_{i,K} \ &= \ \begin{bmatrix} \boldsymbol{\Delta u}_{i,k} \\ \vdots \\ \boldsymbol{\Delta u}_{i,k+M-1} \end{bmatrix} \end{aligned} \tag{5.10}$$

Relationship between $\boldsymbol{\Delta x}_{i,K}$ and $\boldsymbol{\Delta u}_{i,K}$ is provided below with matrices $\boldsymbol{F}_i \in \mathbb{R}^{Mn_x \times n_x}$ and $\boldsymbol{H}_i \in \mathbb{R}^{Mn_x \times Mn_u}$ for prediction horizon of M.

$$
\begin{aligned}
\boldsymbol{\Delta x}_{i,K} &= \boldsymbol{F}_i \boldsymbol{x}_{i,c} + \boldsymbol{H}_i \boldsymbol{\Delta u}_{i,K} \\
\boldsymbol{F}_i &= col\{\boldsymbol{F}_{i,m}\}_{m=1}^{M} \\
\boldsymbol{F}_{i,m} &= \boldsymbol{A}_i^{(m-1)} && , \; m = [1, \cdots, M] \\
\boldsymbol{H}_i &= col\{\boldsymbol{H}_{i,m}\}_{m=1}^{M} \\
\boldsymbol{H}_{i,m} &= \boldsymbol{0} && m = 1 \\
\boldsymbol{H}_{i,m} &= row\{\boldsymbol{h}_{i,l}\} && m = [2, \cdots, M] \\
\boldsymbol{h}_{i,l} &= \boldsymbol{A}_i^{(m-l-1)} \boldsymbol{B}_i && , \; l = [1, \cdots, m-1]
\end{aligned}
\tag{5.11}
$$

Terminal states are given as a function of $\boldsymbol{x}_{i,k}$ and $\boldsymbol{\Delta u}_{i,K}$ below.

$$
\begin{aligned}
\boldsymbol{\Delta x}_{i,k+M} &= \boldsymbol{A}_i^{M} \boldsymbol{x}_{i,k} + \overline{\boldsymbol{B}}_i \boldsymbol{\Delta u}_{i,K} \\
\overline{\boldsymbol{B}}_i &= \begin{bmatrix} \boldsymbol{A}_i^{M-1}\boldsymbol{B}_i & \cdots & \boldsymbol{A}_i^{1}\boldsymbol{B}_i & \boldsymbol{B}_i \end{bmatrix}
\end{aligned}
\tag{5.12}
$$

$\overline{\mathcal{C}}_{eq}(\boldsymbol{x}_{i,K},\ \boldsymbol{u}_{i,K})$ and $\overline{\mathcal{C}}_{ineq}(\boldsymbol{x}_{i,K},\ \boldsymbol{u}_{i,K})$ serve the same purpose with $\mathcal{C}_{eq}(\boldsymbol{x}_{i,k},\ \boldsymbol{u}_{i,k})$ and $\mathcal{C}_{ineq}(\boldsymbol{x}_{i,k},\ \boldsymbol{u}_{i,k})$, however, they are modified to comply with LMPC notation. Calculation of these constraints are provided in [87].

5.2.3 Cooperative Task Prioritization

Let $\boldsymbol{\epsilon}_{i,t} \in \mathbb{R}^{n_{\epsilon t}}$ as provided in (5.13) represent a cooperative task for agent i based on local neighbor state information that is received from neighboring agents denoted as j. Assume that $\boldsymbol{\epsilon}_{i,t} \in \mathbb{R}^{n_{\epsilon t}}$ is defined for a subset of states denoted as $\hat{\boldsymbol{x}}_i \subset \boldsymbol{x}_i$ and $\hat{\boldsymbol{x}}_j \subset \boldsymbol{x}_j$, where $\hat{\boldsymbol{x}}_i \in \mathbb{R}^{n_s}$ and $\hat{\boldsymbol{x}}_j \in \mathbb{R}^{n_s}$. It should be noted $\hat{\boldsymbol{x}}_i$ and $\hat{\boldsymbol{x}}_j$ are assumed to be measured with respect to same coordinate frame. Otherwise the task $\boldsymbol{\epsilon}_{i,t}$ becomes non-linear.

$$
\boldsymbol{\epsilon}_{i,t} = a_{ij}(\hat{\boldsymbol{x}}_j - \hat{\boldsymbol{x}}_i), \;\; t = 1, \cdots, T \tag{5.13}
$$

where T is the maximum number of tasks. First derivative of $\boldsymbol{\epsilon}_{i,t}$ is calculated as follows

$$\begin{aligned} \dot{\boldsymbol{\epsilon}}_{i,t} &= a_{ij}(\dot{\hat{\boldsymbol{x}}}_j - \dot{\hat{\boldsymbol{x}}}_i) \\ \dot{\boldsymbol{\epsilon}}_{i,t} &= \begin{bmatrix} a_{ij}\boldsymbol{I}_{n_s} & -a_{ij}\boldsymbol{I}_{n_s} \end{bmatrix} \begin{bmatrix} \dot{\hat{\boldsymbol{x}}}_j \\ \dot{\hat{\boldsymbol{x}}}_i \end{bmatrix} \end{aligned} \tag{5.14}$$

Collecting the terms on the right-hand side as $\boldsymbol{M}_i(a_{ij}) = \begin{bmatrix} a_{ij}\boldsymbol{I}_{n_s} & -a_{ij}\boldsymbol{I}_{n_s} \end{bmatrix}$ and $\dot{\boldsymbol{S}}_{i,t} = \begin{bmatrix} \dot{\hat{\boldsymbol{x}}}_j^T & \dot{\hat{\boldsymbol{x}}}_i^T \end{bmatrix}^T \in \mathbb{R}^{n_{st}}$, the mapping in (5.14) between task space (TS) velocities and local state space (LSS) velocities takes the form

$$\dot{\boldsymbol{\epsilon}}_{i,t} = \boldsymbol{M}_i(a_{ij})\dot{\boldsymbol{S}}_{i,t} \tag{5.15}$$

Following that, the inverse mapping from TS to LSS is calculated as

$$\dot{\boldsymbol{S}}_{i,t} = \boldsymbol{M}_i^{+}\dot{\boldsymbol{\epsilon}}_{i,t} \tag{5.16}$$

where $\boldsymbol{M}_i^{+}$ is the pseudo inverse of $\boldsymbol{M}_{i,t}$ and a_{ij} is dropped from the expression for brevity. Note that $\boldsymbol{x}_j$ is the states of a neighboring agent.

As illustrated in [136] for robots, excessive(redundant) LSS can be utilized to manage secondary tasks. Let $\boldsymbol{\epsilon}_{i,1}$ and $\boldsymbol{S}_{i,1}$ denotes first TS and LSS and $\boldsymbol{\epsilon}_{i,t}$ and $\boldsymbol{S}_{i,t},\ t = 2, \cdots, T$ denotes the remaining, respectively, then following task management methodology can be used [137],

$$\begin{aligned} \dot{\boldsymbol{S}}_{i,1} &= \boldsymbol{M}_{i,1}^{+}\dot{\boldsymbol{\epsilon}}_1 \\ \dot{\boldsymbol{S}}_{i,t} &= \dot{\boldsymbol{S}}_{i,t-1} + (\boldsymbol{M}_{i,t}\boldsymbol{\Phi}_{i,t-1})^{+}(\dot{\boldsymbol{\epsilon}}_{i,t} - \boldsymbol{M}_{i,t}\dot{\boldsymbol{S}}_{i,t-1}) \end{aligned} \tag{5.17}$$

where $\boldsymbol{\Phi}_{i,t}$ is the null space projection matrix and calculated as follows,

$$\begin{aligned} \boldsymbol{\Phi}_{i,1} &= \boldsymbol{I}_{n_{st}} - \boldsymbol{M}_{i,1}^{+}\boldsymbol{M}_{i,1} \\ \boldsymbol{\Phi}_{i,t} &= \boldsymbol{\Phi}_{i,t-1} - (\boldsymbol{M}_{i,t}\boldsymbol{\Phi}_{i,t-1})^{+}(\boldsymbol{M}_{i,t}\boldsymbol{\Phi}_{i,t-1}) \end{aligned} \tag{5.18}$$

Formulation defined in (5.17) is collected in a minimal representation as in (5.19).

$$\dot{\boldsymbol{S}} \;=\; \boldsymbol{\Psi}\dot{\boldsymbol{\epsilon}} \tag{5.19}$$

where $\dot{\boldsymbol{S}}_i \;=\; \begin{bmatrix} \dot{\boldsymbol{S}}_{i,1}^T & \dot{\boldsymbol{S}}_{i,t}^T & \cdots & \dot{\boldsymbol{S}}_{i,T}^T \end{bmatrix}^T$ and $\dot{\boldsymbol{\epsilon}}_i \;=\; \begin{bmatrix} \dot{\boldsymbol{\epsilon}}_{i,1}^T & \dot{\boldsymbol{\epsilon}}_{i,t}^T & \cdots & \dot{\boldsymbol{\epsilon}}_{i,T}^T \end{bmatrix}^T$. Thus, a quadratic function can be written as in (5.20) based on (5.19).

$$L(\dot{\boldsymbol{\epsilon}}_i) := \dot{\boldsymbol{S}}^T\dot{\boldsymbol{S}} \;=\; \dot{\boldsymbol{\epsilon}}_i^T\boldsymbol{\Psi}^T\boldsymbol{\Psi}\dot{\boldsymbol{\epsilon}}_i \tag{5.20}$$

Finally, based on (5.20), $\mathcal{L}_{i,tp}(\boldsymbol{x}_{i,ref},\ \boldsymbol{x}_{i,k})$ and $\overline{\mathcal{L}}_{i,tp}(\boldsymbol{x}_{i,ref},\ \boldsymbol{x}_{i,K})$ are calculated as given in (5.21) and (5.25), respectively, where $\boldsymbol{Q}_s^T \;=\; \boldsymbol{Q}_s \geq \boldsymbol{0} \in \mathbb{R}^{n_{\epsilon t}\times n_{\epsilon t}}$.

$$\begin{aligned} \mathcal{L}_{i,tp}(\boldsymbol{x}_{i,ref},\ \boldsymbol{x}_{i,k}) \;&=\; \dot{\boldsymbol{\epsilon}}_i^T\boldsymbol{\Psi}^T \left[\begin{array}{c|c} \boldsymbol{Q}_s & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{Q}_s \end{array}\right] \boldsymbol{\Psi}\dot{\boldsymbol{\epsilon}}_i \\ &=\; \dot{\boldsymbol{\epsilon}}_i^T\boldsymbol{Q}_\epsilon\dot{\boldsymbol{\epsilon}}_i \end{aligned} \tag{5.21}$$

Remark 5.2.1 *If states of the agents i and j are linearly related, then $\hat{\boldsymbol{x}}_i$ and $\hat{\boldsymbol{x}}_j$ are linearly related, therefore $\boldsymbol{M}_{i,1}$ is constant. As a result,*

- *$L(\dot{\boldsymbol{\epsilon}}_i)$ outputs a quadratic function of $\dot{\boldsymbol{\epsilon}}_{i,t}$ with scalars γ_t such that $L(\dot{\boldsymbol{\epsilon}}_i) = \sum_{t=1}^{T}\gamma_t\dot{\boldsymbol{\epsilon}}_{i,t}^2$, where $\gamma_1 \geq \cdots \geq \gamma_t \geq \cdots \geq \gamma_T$.*
- *Relationship given for derivatives of state space and task space in* (5.20) *is also valid for state space and task space as follows*

$$L(\boldsymbol{\epsilon}_i) := \boldsymbol{S}^T\boldsymbol{S} \;=\; \boldsymbol{\epsilon}_i^T\boldsymbol{\Psi}^T\boldsymbol{\Psi}\boldsymbol{\epsilon}_i \tag{5.22}$$

which results in

$$\begin{aligned} \mathcal{L}_{i,tp}(\boldsymbol{x}_{i,ref},\ \boldsymbol{x}_{i,k}) \;&=\; \boldsymbol{\epsilon}_i^T\boldsymbol{\Psi}^T \left[\begin{array}{c|c} \boldsymbol{Q}_s & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{Q}_s \end{array}\right] \boldsymbol{\Psi}\boldsymbol{\epsilon}_i \\ &=\; \boldsymbol{\epsilon}_i^T\boldsymbol{Q}_\epsilon\boldsymbol{\epsilon}_i \end{aligned} \tag{5.23}$$

Let $\boldsymbol{Q}_\epsilon$ partitioned as in (5.24), and let $\boldsymbol{X}_{i,ref} := \overline{\boldsymbol{1}} \otimes \boldsymbol{x}_{i,ref}$, where $\overline{\boldsymbol{1}} = [1\ \cdots\ 1]^T \in \mathbb{R}^M$. Then overall task matrix $\overline{\boldsymbol{\epsilon}}_i$ can be written as an affine function of $\boldsymbol{X}_{i,ref}$ and $\boldsymbol{x}_{i,K}$) and $\overline{\mathcal{L}}_{i,tp}(\boldsymbol{x}_{i,ref},\ \boldsymbol{x}_{i,K})$ is calculated as provided in (5.25).

$$\boldsymbol{Q}_{\epsilon} \;=\; \left[\begin{array}{c|c} \boldsymbol{Q}_{\epsilon,11} & \boldsymbol{Q}_{\epsilon,12} \\ \hline \boldsymbol{Q}_{\epsilon,21} & \boldsymbol{Q}_{\epsilon,22} \end{array}\right] \tag{5.24}$$

$$\overline{\mathcal{L}}_{i,tp}(\boldsymbol{x}_{i,ref},\; \boldsymbol{x}_{i,K}) \;=\; \overline{\boldsymbol{\epsilon}}_i^T \overline{\boldsymbol{Q}}_{\epsilon} \overline{\boldsymbol{\epsilon}}_i \tag{5.25}$$

,where $\overline{\boldsymbol{Q}}_{\epsilon,\star *} \;=\; diag\{\boldsymbol{Q}_{\epsilon,\star *}\}_1^M$ and $\star, * \;=\; \{1,2\}$.

$$\overline{\boldsymbol{Q}}_{\epsilon} \;=\; \left[\begin{array}{c|c} \overline{\boldsymbol{Q}}_{\epsilon,11} & \overline{\boldsymbol{Q}}_{\epsilon,12} \\ \hline \overline{\boldsymbol{Q}}_{\epsilon,21} & \overline{\boldsymbol{Q}}_{\epsilon,22} \end{array}\right] \tag{5.26}$$

5.2.4 Implementation of the MPCs

The docking controller is designed to have several layers to decide how to perform the docking maneuver based on the distance between the agent i and neighboring agents. The layers, as mentioned earlier, are namely, Selector NMPC and LMPC. Essentially, Selector works as a governor and decides the reference trajectory that the LMPC tracks. If the absolute distance between agent i and neighboring agents is greater than a threshold, NMPC generates a trajectory towards the neighboring agents using neighboring state information. Note that NMPC is not aiming to achieve terminal state constraint, but it repeatedly minimizes the state error within the given prediction horizon. Then generated trajectory, which creeps towards the neighbors, is passed to LMPC. If the distance is smaller than the threshold, LMPC receives the neighboring state information to generate control inputs without needing NMPC. The distance and threshold are denoted as $\boldsymbol{e} \;=\; \| [\boldsymbol{p}_j] - [\boldsymbol{p}_i] \|_2^2 \in \mathbb{R}$ and $e_d \in \mathbb{R}$. Currently available state and input information are denoted as $\boldsymbol{x}_i(t-1)$, $\boldsymbol{x}_j(t-1)$ and $\boldsymbol{u}_i(t-1)$,where t represents current sample and $(t-1)$ represents previous sample. The pipeline for the control strategy described above is given in Figure 5.1.

When triggered, NMPC generates a rough trajectory towards agent j for given prediction horizon M based on the formulation given in (5.6). Then a portion of this trajectory that is determined by the control horizon M_u is extracted, and this is denoted as $\boldsymbol{x}_{nmpc}$.

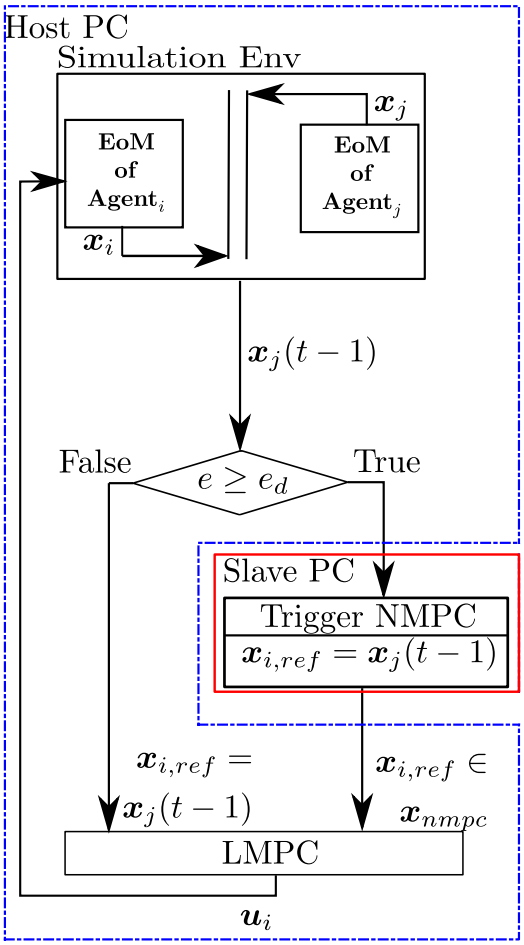


Figure 5.1. Unified MPC framework for docking..

Optimization parameters for NMPC such as M, t_f, and M_u are selected empirically to achieve a rough trajectory and allow a time interval to queue a new trajectory to guide the agent i to the vicinity of other agents. Selection of the optimization parameters with the right-hand-side sparsity template described in [135] provides a longer time to queue a new trajectory. Before LMPC receives the output of the NMPC or the neighboring state information that are $\boldsymbol{x}_{nmpc}$ and $\boldsymbol{x}_j(t-1)$, respectively, a line or a set of line segments are populated based on the prediction horizon of the LMPC. Finally, the tracking based on the optimization provided in (5.8) is done by the LMPC.

Property	Value
M	14
M_u	1
Q	$3\boldsymbol{I}$
R	$3\boldsymbol{I}$
Q_s	Solution to Discrete Algebraic Riccati Equation*

(a)

Property	Value
M	10
M_u	3
t_f	1 (s)
Q	$5\boldsymbol{I}$
R	$1\boldsymbol{I}$
Q_s	$40\boldsymbol{I}$
E_d	2 (m)

(b)

Table 5.1. (a) LMPC parameters. (b) NMPC parameters. t_f is the regarding time for prediction horizon. * for discrete algebraic Riccati solution, see [2].

5.3 Simulation and Results

This section presents the simulation results for the proposed strategy based on the simulation setup of a quadrotor and a rover, as illustrated in Fig. 5.5.

NMPC and LMPC parameters are provided in Table 5.1 and quadcopter properties are given in [133] in Table 1. Two scenarios are illustrated: 1) proximity docking on a moving agent and 2) large range docking on a moving agent, where LMPC and NMPC - LMPC strategies are tested, respectively. Let $\mathcal{P}_s = \{1, 2, 3, 4\}$ represent the priority of states, where 1 is the highest priority and 4 is the lowest priority, and $s = p, v, e, w$ represent the subset of states that $\mathcal{P}$ is representing. Based on this, LMPC uses the priority map of $\{\mathcal{P}_p, \mathcal{P}_v, \mathcal{P}_e, \mathcal{P}_w\} = \{2, 3, 1, 4\}$. Priority mapping is used to arrange tasks defined as $\boldsymbol{\epsilon}_{i,t}$, where $i \in \{1, 2, 3, 4\}$. This mapping is valid for both LMPC and NMPC.

5.3.1 Case Study 1: Proximity docking on the rover

Initial condition for quadcopter and the rover are

$$\begin{aligned} \boldsymbol{x}_{q,0} &= [0\ 0.5\ -10\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T \\ \boldsymbol{x}_{r,0} &= [0\ 0\ 0\ 0.5\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T \end{aligned} \tag{5.27}$$

, respectively. Initial inputs for the quadcopter is $\boldsymbol{u}_{i,c} = [9.81m_q\ 0\ 0\ 0]^T$ to temper the behavior of the $\boldsymbol{f}_q$. Given the initial values, the quadcopter is guided to the rover with -0.1 m offset in $z-$direction as illustrated in Fig. 5.2. Evaluating Fig. 5.2, the quadcopter makes a free fall $3/4^{th}$ of a second, then recovers. During this phase, it loses altitude and gains speed in the w component up to $6\ m/s$. The duration of the free fall is correlated to $\boldsymbol{R}$ in the cost function and inequality constraints provided to the LMPC, which is $\overline{C}_{ineq}(\boldsymbol{x}_{i,K},\ \boldsymbol{u}_{i,K})$. After the free fall approximately after $t = 1$ (s), controller makes a correction in $\boldsymbol{e}$ and starts approaching to the rover in x- and y-directions.

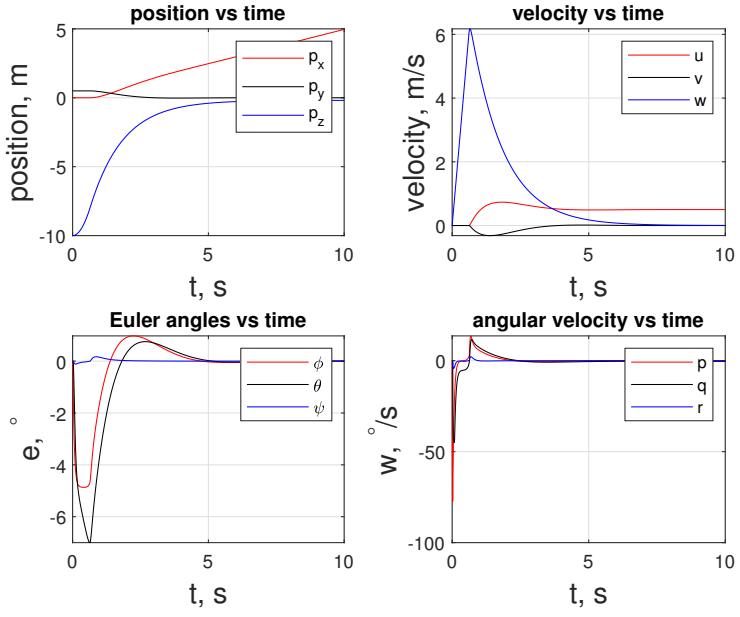


Figure 5.2. State trajectory of agent of the quadrotor..

Previously mentioned free fall can also be seen in Fig. 5.3 in the f_q subplot, where f_q stays at 0 N for that period. During the maneuver, f_q is upper bounded by 12 N. The maximums for $\boldsymbol{t}_q$ are reached in the x-direction, where -0.52 and 0.05 Nm of toque are observed at extremes.

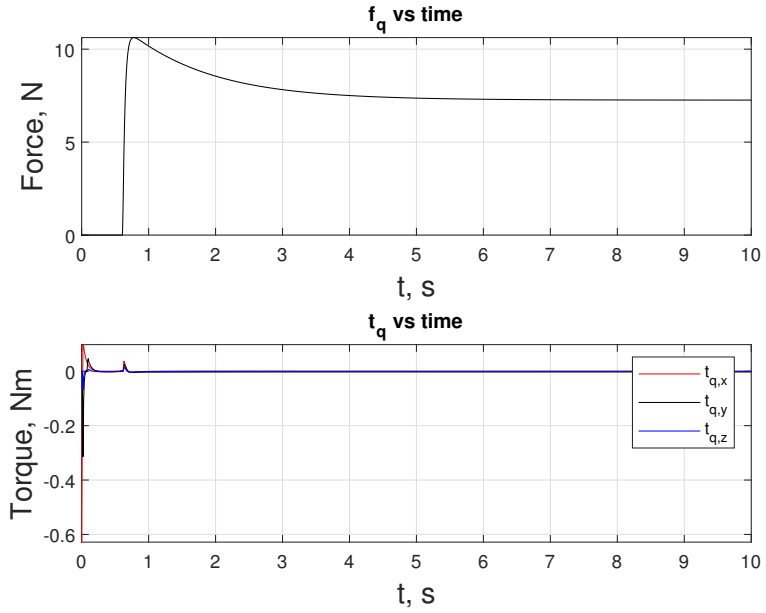


Figure 5.3. Inputs f_q and $\boldsymbol{t}_q$ over the duration of the docking maneuver..

Fig. 5.4 provides an overview of the performed trajectory along with the heading direction of the quadcopter, which is aligned with the $\boldsymbol{u}_1$ axis of $\mathcal{F}_B$.

5.3.2 Case Study 2: Long range docking on the rover

Initial conditions for the quadcopter and the rover are

$$\begin{aligned} \boldsymbol{x}_{q,0} &= [0\ 0\ -10\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T \\ \boldsymbol{x}_{r,0} &= [10\ 10\ 0\ 0.5\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T \end{aligned} \tag{5.28}$$

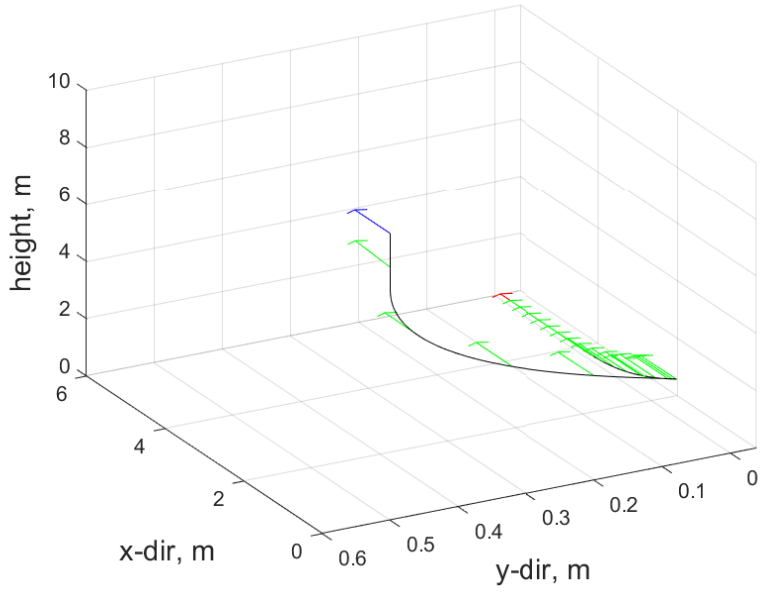


Figure 5.4. Trajectory and heading of the quadrotor.

, respectively. Initial inputs for the quadcopter is $\boldsymbol{u}_{i,c} = [9.81m_q\ 0\ 0\ 0]^T$. This case illustrates the long-range capability of the proposed control strategy. The overview of the scenario is provided as a set snapshot in Fig. 5.5, and the video of the performed scenario can be found in the following *link*.

In this scenario, LMPC tracks the $\boldsymbol{x}_{nmpc}$ the NMPC generates until $\boldsymbol{e}$ is less than $\boldsymbol{e}_d$. At about 41^{th} s a transition happens and LMPC starts tracking the $\boldsymbol{x}_j(t-1)$. Figures 5.6 and 5.8 reveal the difference in proximity and long-range docking maneuvers, to which LMPC is more sensitive to the magnitude of the tracking error $\boldsymbol{e}$. An important feature of this motion is that signal oscillates at velocity level both in linear and angular motion. The same oscillatory behavior is visible for inputs f_q and $\boldsymbol{t}_q$ as provided in Fig. 5.7, which is because the quadcopter's current states are changing as the $\boldsymbol{x}_{nmpc}$ is generated on NMPC side, sent to the LMPC and populated as line segments.

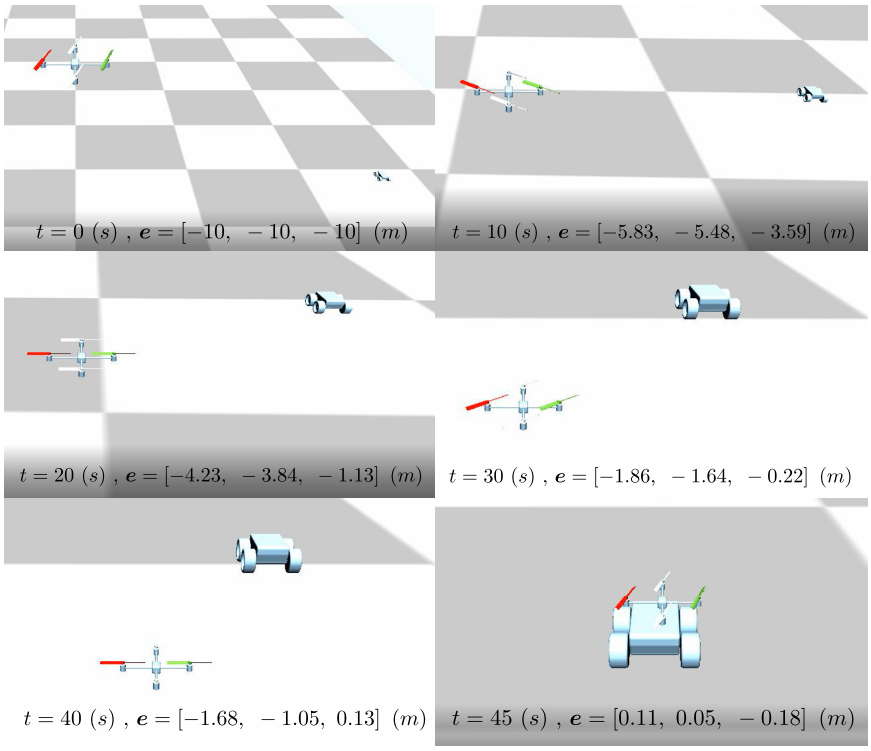


Figure 5.5. Snaps of the realized trajectory during long range docking maneuver..

The positional errors between rover and quadrotor in Fig. 5.8 illustrates positional errors in $x-$ and $y-$directions approach to zero, while the error in $z-$ direction approach to -0.1 m.

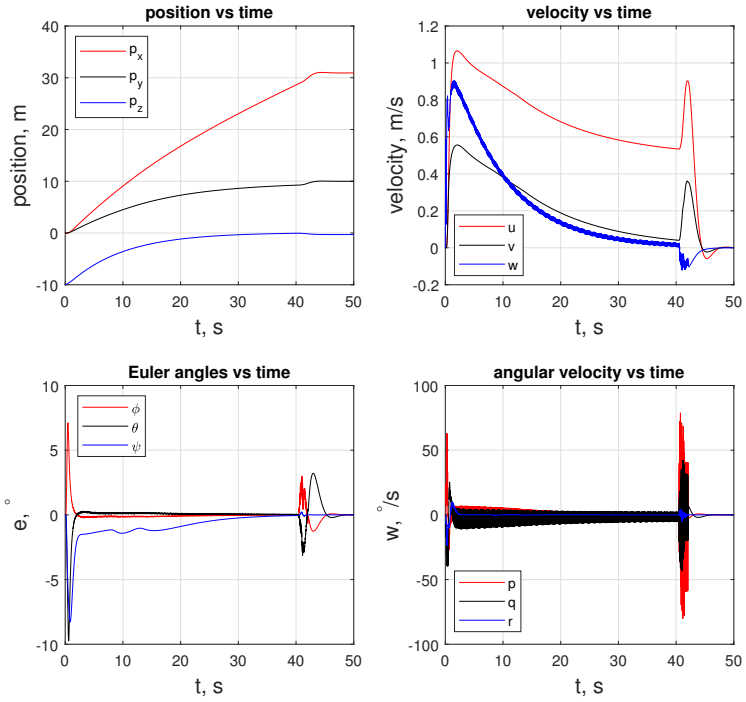


Figure 5.6. State trajectory of agent of the quadrotor..

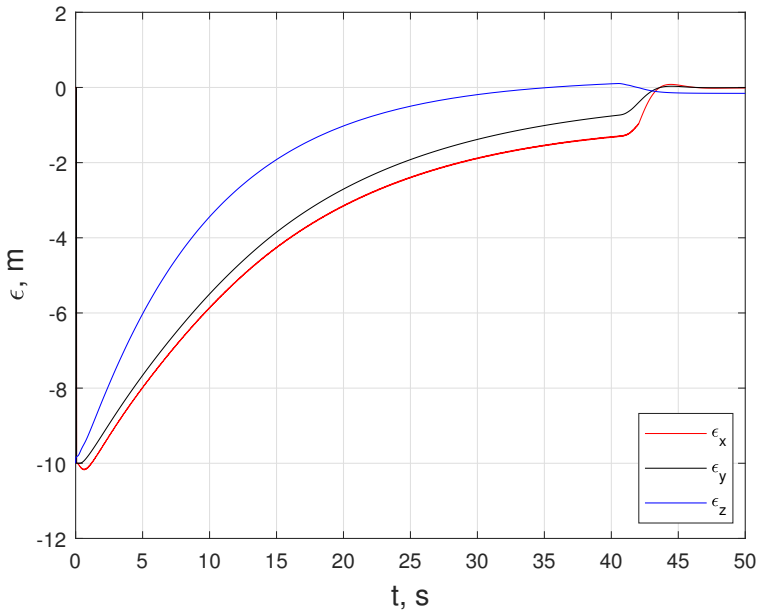


Figure 5.8. Position error between rover and quadrotor..

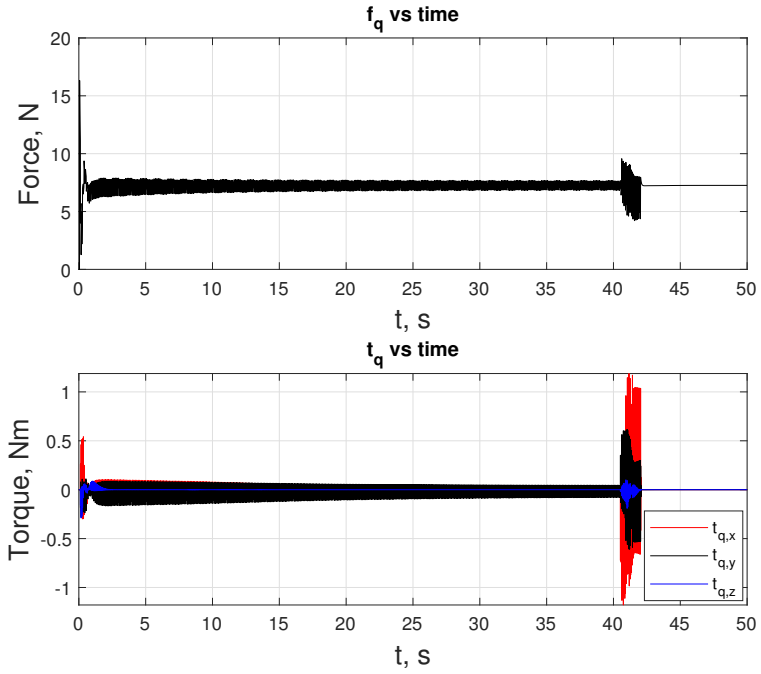


Figure 5.7. Inputs f_q and $\boldsymbol{t}_q$ over the duration of the docking maneuver..

Figure 5.9 illustrates calculation time of each NMPC trigger. Initially, generation costs approximately 0.1 s and as the quadcopter settles on the trajectory trajectory generation times reduces to 0.05 s. For given elapsed time profile, overall calculation time is calculated to be 13.91 s. The upper bound line illustrates the control horizon M_u for the NMPC.

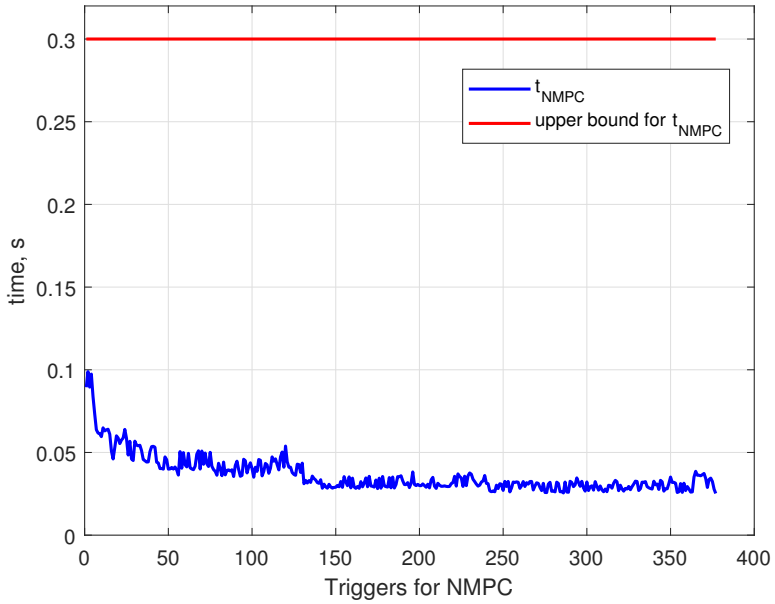


Figure 5.9. Calculation time for NMPC problem at every instant. .

5.4 Conclusions

This work proposes a unified MPC control strategy capable of handling long-range to proximity docking maneuvers. A rough but fast NMPC method is run to propagate the agent to a closer vicinity then LMPC handles more sensitive proximity motion. The sensitivity of the LMPC is apparent in Figures 5.6 and 5.8, where a smooth approach suddenly becomes relatively violent. Based on the simulations, the selection of the matrices $\boldsymbol{Q}$, $\boldsymbol{R}$ and $\boldsymbol{Q}_s$ makes the transition from NMPC-LMPC to LMPC smoother. Note that the docking of the quadrotor is almost tangential to the $x-y$ plane, and during the final section of the maneuver, it penetrates the docking platform. The latter problem can be resolved with an approach constraint to the MPC controllers. Most importantly, the proposed method is readily applicable among dynamic systems, including non-linear systems, since the

controller's structure stays the same. Using local neighbor information instead of external sensors or observers makes this control strategy decentralized naturally.

Chapter 6

Graph Theoretic Trajectory Optimization of ASLB Biped Robot

6.1 ASLB - A Bipedal Robot for Dynamics Locomotion

6.1.1 System Composition

ASLB is a floating bipedal platform that each leg is composed of hybrid three degrees of freedom (DoF) structure. Specifically, starting from the body, kinematic structure of legs are revolute joint (at shoulder) followed by a parallel 5R mechanism. This kinematics results in three actuated and two passive joint coordinates in each leg. A rendered 3D model and manufactured prototype of ASLB is provided in Fig. 6.1.

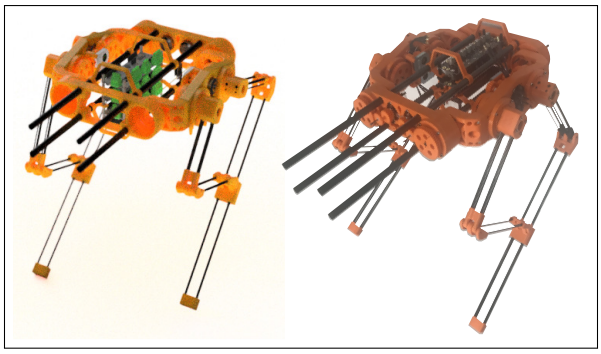


Figure 6.1. 3D model (left) and manufactured prototype of ASLB (right)..

6.1.2 Kinematics

Kinematic model of the floating platform starts with a set of unactuated joints that gives six DoF to the mobile platform. These joints are collected in column matrix $\boldsymbol{q}_B \in \mathbb{R}^3 \times SO(3)$. Starting from the inertial frame, joints are located in a sequence such that first translational joints $\boldsymbol{q}_{t,B} = [q_1, q_2, q_3]^T \in \mathbb{R}^3$ are located in respective x, y and z directions.

Assuming the euler sequence of YZX, rotational joints are $\boldsymbol{q}_{r,B} = [q_4, q_5, q_6]^T \in \mathbb{R}^3$. Body frame $\mathcal{F}_B$ is kinematically represented with respect to $\mathcal{F}_I$ by $\boldsymbol{q}_B = [\boldsymbol{q}_{t,B}^T, \boldsymbol{q}_{r,B}^T]^T$ such that $\boldsymbol{r}_B(\boldsymbol{q}_{t,B})$ and $\boldsymbol{C}_{IB}(\boldsymbol{q}_{r,B})$ are translation and rotation matrices from $\mathcal{F}_I$ to $\mathcal{F}_B$, respectively. Kinematic complexity starts after the origin of $\mathcal{F}_B$, where actuated legs are attached. There are two legs attached to the body and regarding joints are represented by $\boldsymbol{\theta}_i \in \mathbb{R}^{n_i}$, where $n_i = 5$ and $i = R, L$. In total, combined DoF for legs are given as $n_a = 10$. Composition of $\boldsymbol{\theta}_i$ for each leg is given as $\boldsymbol{\theta}_i = [\theta_{0,i}\ \theta_{1,i}\ \theta_{2,i}\ \theta_{3,i}\ \theta_{4,i}]^T$, where there are three active and two passive joints, respectively $\boldsymbol{\theta}_{a,i} = [\theta_{0,i}\ \theta_{1,i}\ \theta_{3,i}]^T \in \mathbb{R}^3$ and $\boldsymbol{\theta}_{p,i}(\boldsymbol{\theta}_{a,i}) = [\theta_{2,i}(\boldsymbol{\theta}_{a,i})\ \theta_{4,i}(\boldsymbol{\theta}_{a,i})]^T$. Passive joints can be written as a function of $\boldsymbol{\theta}_{a,i}$ based on a velocity contraint as described in § 6.1.2.1. As a result, total joint space of ASLB is given by $\boldsymbol{q} = [\boldsymbol{q}_B,\ \boldsymbol{\theta}_{a,R},\ \boldsymbol{\theta}_{a,L}]^T \in \mathbb{R}^{12}$. The kinamtic structure of right leg is illustrated in Fig. 6.2

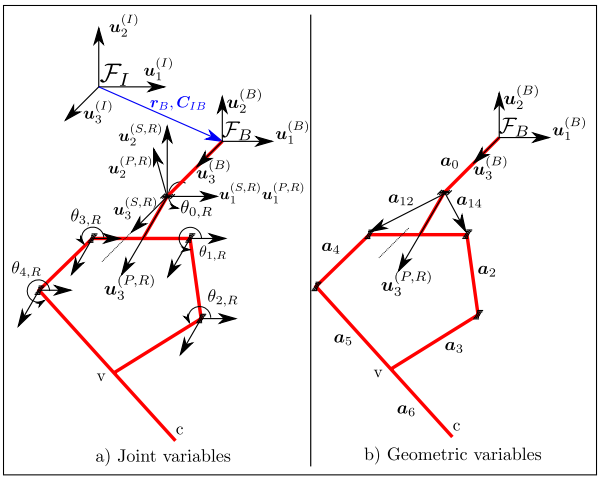


Figure 6.2. Kinematic structure of ASLB.

6.1.2.1 Passive - Active joint Relation

As described in [138] starting from the origin of the body frame $\mathcal{F}_B$, there are two chains to reach point $\boldsymbol{v}$ on both legs. Let $\boldsymbol{R}_*$ represent elementary rotation matrix, where $* = x, y, z$ are active axes. Then these two chains can be written as in Eq. (6.1).

$$\begin{aligned} \boldsymbol{r}_{v,1} &= \boldsymbol{a}_{12} + \boldsymbol{R}_z(\theta_{1,R})\boldsymbol{a}_2 + \boldsymbol{R}_z(\theta_{1,R} + \theta_{2,R})\boldsymbol{a}_2 \\ \boldsymbol{r}_{v,2} &= \boldsymbol{a}_{14} + \boldsymbol{R}_z(\theta_{3,R})\boldsymbol{a}_2 + \boldsymbol{R}_z(\theta_{3,R} + \theta_{4,R})\boldsymbol{a}_2 \end{aligned} \tag{6.1}$$

Differentiating Eq. (6.1) results in velocity equations of $\boldsymbol{v}_{r,1}$ and $\boldsymbol{v}_{r,2}$, which are equal. Writing this relationship as below relates the passive joints to active joints as described in Eq. (6.2). Let $\overline{\boldsymbol{\theta}}_{a,i} = [\theta_{1,i}\ \theta_{3,i}]^T$ denote the set of active joints related to the closed loop, then $\boldsymbol{J}_{a,i} \in \mathbb{R}^2$ becomes a square matrix.

$$\begin{aligned} \boldsymbol{v}_{r,1} &= \boldsymbol{J}_{r,1} \begin{bmatrix} \overline{\boldsymbol{\theta}}_{a,i}^T & \boldsymbol{\theta}_{p,i}^T \end{bmatrix}^T \\ \boldsymbol{v}_{r,2} &= \boldsymbol{J}_{r,2} \begin{bmatrix} \overline{\boldsymbol{\theta}}_{a,i}^T & \boldsymbol{\theta}_{p,i}^T \end{bmatrix}^T \\ \boldsymbol{0} &= (\boldsymbol{J}_{r,1} - \boldsymbol{J}_{r,2}) \begin{bmatrix} \overline{\boldsymbol{\theta}}_{a,i}^T & \boldsymbol{\theta}_{p,i}^T \end{bmatrix}^T \\ &= \left[\begin{array}{c|c} \boldsymbol{J}_a & \boldsymbol{J}_p \end{array}\right] \begin{bmatrix} \overline{\boldsymbol{\theta}}_{a,i}^T & \boldsymbol{\theta}_{p,i}^T \end{bmatrix}^T \end{aligned} \tag{6.2}$$

Finally passive joints are related to active joints as given in Eq. (6.3).

$$\begin{aligned} \boldsymbol{\theta}_{p,i} &= \boldsymbol{J}_{pa}\overline{\boldsymbol{\theta}}_{a,i} \\ \boldsymbol{J}_{pa} &= -\boldsymbol{J}_{p,i}^{-1}\boldsymbol{J}_{a,i} \end{aligned} \tag{6.3}$$

6.1.2.2 Forward Kinematics

Forward kinematics for each leg calculates the position of the contact point c with respect to origin of $\mathcal{F}_B$, as illustrated in Fig. 6.2. The aforementioned position vector is denoted as $\boldsymbol{r}_c$. As mentioned in § 6.1.2.1 the active and passive joint angles are related to

each other and unless the passive joints are measured by sensors, they have to be calculated from this relationship. To do that position vectors $\boldsymbol{r}_{v,1}$ and and $\boldsymbol{r}_{v,2}$ are arranged as below.

$$\begin{aligned}
&\begin{bmatrix} cos(\theta_{3,R}+\theta_{4,R}) \\ sin(\theta_{3,R}+\theta_{4,R}) \\ 0 \end{bmatrix} a_{5,x} = \\
&- \begin{bmatrix} cos(\theta_{1,R}+\theta_{2,R}) \\ sin(\theta_{1,R}+\theta_{2,R}) \\ 0 \end{bmatrix} a_{3,x} + \begin{bmatrix} a_{12,x}-a_{14,x} \\ a_{12,y}-a_{14,y} \\ a_{12,z}-a_{14,z} \end{bmatrix} \\
&+ \begin{bmatrix} cos(\theta_{1,R}) \\ sin(\theta_{1,R}) \\ 0 \end{bmatrix} a_{2,x} - \begin{bmatrix} cos(\theta_{3,R}) \\ sin(\theta_{3,R}) \\ 0 \end{bmatrix} a_{4,x}
\end{aligned} \tag{6.4}$$

First two rows of Eq. (6.4) can be written in a minimal form as provided in Eq. (6.5) such that terms including $\theta_{3,R}$, $\theta_{4,R}$ are left alone.

$$\begin{aligned}
\begin{bmatrix} cos(\theta_{3,R}+\theta_{4,R}) \\ sin(\theta_{3,R}+\theta_{4,R}) \end{bmatrix} &= \begin{bmatrix} T_x \\ T_y \end{bmatrix} + \frac{a_{3,x}}{a_{5,x}} \begin{bmatrix} cos(\theta_{1,R}+\theta_{2,R}) \\ sin(\theta_{1,R}+\theta_{2,R}) \end{bmatrix} \\
T_x &= \frac{1}{a_{5,x}}(a_{12,x}-a_{14,x}+cos(\theta_{1,R})+cos(\theta_{3,R})) \\
T_y &= \frac{1}{a_{5,x}}(a_{12,y}-a_{14,y}+sin(\theta_{1,R})+sin(\theta_{3,R}))
\end{aligned} \tag{6.5}$$

Elements of Eq. (6.5) are squared and summed to obtain Eq.(6.6). Then trigonometric expressions are written in terms of tangents of the half angles, which leads to the a solution to $\theta_{12,R} = \theta_{1,R} + \theta_{2,R}$ as provided in Eq. (6.7).

$$T_x^2 + T_y^2 + \frac{a_{3,x}}{a_{5,x}}^2 + 2T_x\frac{a_{3,x}}{a_{5,x}}cos(\theta_{12,R}) + 2Bsin(\theta_{12,R}) = 1 \tag{6.6}$$

$$
\begin{aligned}
\theta_{12,R} &= 2atan(T_{12}) \\
T_{12} &= \frac{-2C_3 \pm \sqrt{(2C_3)^2 - 4(C_1 - C_2)(C_1 + C_2)}}{2(C_1 - C_2)} \\
C_1 &= T_x^2 + T_y^2 + \frac{a_{3,x}}{a_{5,x}}^2 - 1 \\
C_2 &= 2T_x \frac{a_{3,x}}{a_{5,x}} \\
C_3 &= 2T_y \frac{a_{3,x}}{a_{5,x}}
\end{aligned}
\tag{6.7}
$$

Using $\theta_{12,R}$ one can write Eq. (6.8) and solve for $\theta_{34,R} = \theta_{3,R} + \theta_{4,R}$. Finally, tip point location $\boldsymbol{r}_c$ is calculated as provided in Eq. (6.9).

$$
\begin{aligned}
\theta_{34,R} &= atan2(s_{34,R}, c_{34,R}) \\
\begin{bmatrix} c_{34,R} \\ s_{34,R} \end{bmatrix} &= \begin{bmatrix} T_x + \frac{a_{3,x}}{a_{5,x}} cos(\theta_{12,R}) \\ T_y + \frac{a_{3,x}}{a_{5,x}} sin(\theta_{12,R}) \end{bmatrix}
\end{aligned}
\tag{6.8}
$$

$$
\begin{aligned}
\boldsymbol{r}_c = \boldsymbol{a}_0 + \boldsymbol{R}_x(\theta_{0,R})(\boldsymbol{a}_{14} + \boldsymbol{R}_z(\theta_{3,R})\boldsymbol{a}_4 \\
+ \boldsymbol{R}_z(\theta_{34,R})(\boldsymbol{a}_5 + \boldsymbol{a}_6))
\end{aligned}
\tag{6.9}
$$

6.1.2.3 Inverse Kinematics

Inverse kinematics is illustrated on right leg and calculations provided here can be duplicated for left leg. Inverse kinematics solution in this work relies on geometric calculation of $\theta_{0,R}$ as provided in Eq. (6.10). The geometric entities are illustrated in Figure 6.3.

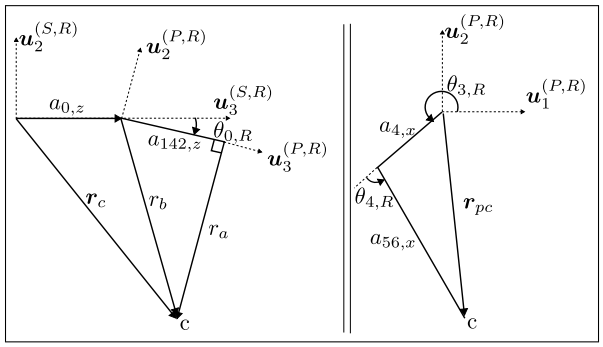


Figure 6.3. Geometric entities realted to calculation of $\theta_{0,R}$..

$$
\begin{aligned}
\theta_0 &= atan2(sin(\theta_{0,R}), cos(\theta_{0,R})) \\
\begin{bmatrix} cos(\theta_0) \\ sin(\theta_0) \end{bmatrix} &= \begin{bmatrix} a_{142,z} & -r_a \\ -r_a & -a_{142,z} \end{bmatrix}^{-1} \begin{bmatrix} r_z - a_{0,z} \\ r_y - a_{0,y} \end{bmatrix} \\
\boldsymbol{r}_1 &= \boldsymbol{r} - \boldsymbol{a}_0 \\
r_a &= \sqrt{r_{1,y}^2 + r_{1,z}^2 - a_{14,z}^2} \\
a_{142,z} &= a_{14,z} + a_{2,z}
\end{aligned}
\tag{6.10}
$$

To calculate active joints $\theta_{1,R}$ and $\theta_{3,R}$ x- and y- components of the position vector $\boldsymbol{r}_{pc}$ as given in Eq. (6.11) is exploited as in Eq. (6.12).

$$
\begin{aligned}
r_{pc,x} &= a_{4,x}cos(\theta_{3,R}) + a_{56,x}cos(\theta_{3,R} + \theta_{4,R}) \\
r_{pc,y} &= a_{4,x}sin(\theta_{3,R}) + a_{56,x}sin(\theta_{3,R} + \theta_{4,R})
\end{aligned}
\tag{6.11}
$$

$$
\begin{aligned}
r_{pc,x} &= a_{4,x}cos(\theta_{3,R}) + a_{56,x}(cos(\theta_{3,R})cos(\theta_{4,R}) - sin(\theta_{3,R})sin(\theta_{4,R})) \\
r_{pc,y} &= a_{4,x}sin(\theta_{3,R}) + a_{56,x}(cos(\theta_{3,R})sin(\theta_{4,R}) + sin(\theta_{3,R})cos(\theta_{4,R}))
\end{aligned}
\tag{6.12}
$$

Equation (6.12) is rewritten as in Eq. (6.13) and each scalar equation can be squared and summed as in Eq. (6.14).

$$
\begin{aligned}
cos(\theta_{3,R}) &= (r_{pc,x}/a_{4,x}) + (a_{56,x}/a_{4,x})cos(\theta_{3,R}+\theta_{4,R}) \\
&= C_4 + C_5 cos(\theta_{3,R}+\theta_{4,R}) \\
sin(\theta_{3,R}) &= (r_{pc,y}/a_{4,x}) + (a_{56,x}/a_{4,x})(sin(\theta_{3,R}+\theta_{4,R}) \\
&= C_6 + C_5 sin(\theta_{3,R}+\theta_{4,R})
\end{aligned}
\tag{6.13}
$$

$$
\begin{aligned}
1 &= C_4^2 + C_6^2 + C_5^2 + 2C_4C_5cos(\theta_{3,R}+\theta_{4,R}) + 2C_6C_5sin(\theta_{3,R}+\theta_{4,R}) \\
0 &= (C_4^2 + C_6^2 + C_5^2 - 1) + (2C_4C_5)cos(\theta_{3,R}+\theta_{4,R}) + (2C_6C_5)sin(\theta_{3,R}+\theta_{4,R}) \\
0 &= C_7 + C_8 cos(\theta_{3,R}+\theta_{4,R}) + C_9 sin(\theta_{3,R}+\theta_{4,R})
\end{aligned}
\tag{6.14}
$$

Then using tangents of half angle, Eq. (6.14) can be further manipulated in Eq. (6.15). The solution to $\theta_{34,R}$ can be calculated by solving the quadratic problem for T_{34} as illustrated in Eq. (6.7).

$$
0 = (C_7 - C_8)T_{34}^2 + 2C_9T_{34} + (C_7 + C_8) \tag{6.15}
$$

After finding solution to $\theta_{34,R}$, these values are inserted into Eq. (6.11) to calculate $\theta_{3,R}$. This completes the solution to one chain of the leg mechanism.

Geometric definitions such as $\boldsymbol{r}_{pcl}$, $\boldsymbol{r}_{cl}$ are provided in Figure 6.4 to calculate the joint variables on the other chain that contains $\boldsymbol{a}_2$ and $\boldsymbol{a}_3$.

Using the known joint variables, point v is represented from origin of the joint $\mathbf{3}$ with a vector denoted as $\boldsymbol{r}_{pcl}$. Alternatively, point v can also be represented from origin of the joint $\mathbf{1}$ with a vector that is $\boldsymbol{r}_{cl}$. These vectors are defined on (P, R) plane, where the 5R mechanism lies on, and provided in Eq. (6.16) and Eq. (6.17), respectively.

$$
\begin{aligned}
r_{pcl,x} &= r_{pc,x} - a_{6,x}cos(\theta_{3,R}+\theta_{4,R}) \\
r_{pcl,y} &= r_{pc,y} - a_{6,x}sin(\theta_{3,R}+\theta_{4,R})
\end{aligned}
\tag{6.16}
$$

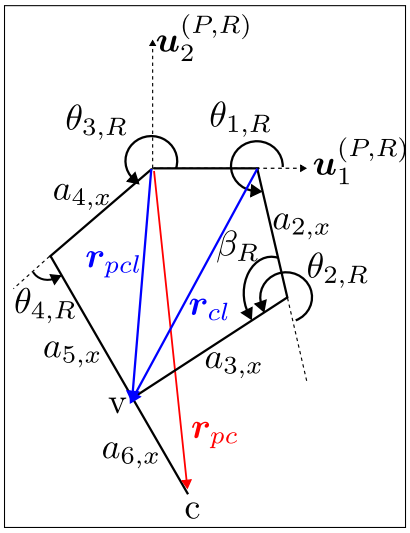


Figure 6.4. Geometric entities realted to calculation of $\theta_{0,R}$..

$$\begin{aligned} r_{cl,x} &= a_{2,x}cos(\theta_{1,R}) + a_{3,x}cos(\theta_{1,R} + \theta_{2,R}) \\ r_{cl,y} &= a_{2,x}sin(\theta_{1,R}) + a_{3,x}sin(\theta_{1,R} + \theta_{2,R}) \end{aligned} \tag{6.17}$$

Scalar equations $r_{cl,x}$ and $r_{cl,y}$ of Eq. (6.17) are squared, summed and reorganized as provided in Eq. (6.18) to calculate β_R, which leads to $\theta_{2,R}$.

$$\begin{aligned} cos(\beta_R) &= (-1)\frac{r_{cl,x}^2 + r_{cl,y}^2 - a_{2,x}^2 - a_{3,x}^2}{2a_{2,x}a_{3,x}} \\ \beta_R &= acos((-1)\frac{r_{cl,x}^2 + r_{cl,y}^2 - a_{2,x}^2 - a_{3,x}^2}{2a_{2,x}a_{3,x}}) \\ \theta_{2,R} &= \pi + \beta_R \end{aligned} \tag{6.18}$$

To calculate $\theta_{1,R}$, $r_{cl,x}$ and $r_{cl,y}$ is represented as a function of r_{pcl}. This is given in Eq. (6.19) in matrix form, where terms with $\theta_{1,R}$ are the unknown variables. Solution to Eq. (6.19) for $\theta_{1,R}$ finishes the inverse kinematic solution of the right leg. Procedure for left leg is identital.

$$\begin{bmatrix} r_{pcl,x} + (a_{14,x} - a_{12,x}) \\ r_{pcl,y} \end{bmatrix} = \begin{bmatrix} a_{2,x} + a_{3,x}cos(\theta_{2,R}) & -a_{3,x}sin(\theta_{2,R}) \\ a_{3,x}sin(\theta_{2,R}) & a_{2,x} + a_{3,x}cos(\theta_{2,R}) \end{bmatrix} \begin{bmatrix} cos(\theta_1) \\ sin(\theta_1) \end{bmatrix} \tag{6.19}$$

$$\theta_{1,R} = atan2(sin(\theta_1), cos(\theta_1))$$

6.1.3 Dynamics

Based on the generalized coordinates, multi-body dynamics of ASLB is formulated as in (Eq. 6.20)

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{S}\boldsymbol{\tau} + \boldsymbol{J}_{C,i}^T \boldsymbol{F}_{C,i} \tag{6.20}$$

,where $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{12\times12}$, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{12}$, $\boldsymbol{G}(\boldsymbol{q}) \in \mathbb{R}^{12}$ are generalized mass, Coriolis & centrifugal term and gravitational term matrices. $\boldsymbol{S} \in \mathbb{R}^{12\times6}$, $\boldsymbol{\tau} \in \mathbb{R}^6$, $\boldsymbol{F}_{C,i} \in \mathbb{R}^3$ and $\boldsymbol{J}_{C,i} \in \mathbb{R}^{3\times12}$ are the selection matrix for the actuated joints of regarding legs, actuated joint torques, external force on the tip of i^{th} leg and geometric Jacobian of the tip point of the i^{th} leg.

Let $\boldsymbol{x} = [\boldsymbol{q},\ \dot{\boldsymbol{q}}]^T \in \mathbb{R}^{24}$, $\boldsymbol{u}_\tau = \boldsymbol{\tau}$ and $\boldsymbol{u}_c = \boldsymbol{F}_{C,i}\ \forall i$ are the states, inputs to motors and external forces acting on tip point of the legs, respectively, then non-linear dynamics of the robot can be written as in Eq. 6.21.

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}_\tau, \boldsymbol{u}) \\ \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}_\tau, \boldsymbol{u}) &= \begin{bmatrix} \dot{\boldsymbol{q}} \\ \boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{\Theta} \end{bmatrix} \\ \boldsymbol{\theta} &= -\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{S}\boldsymbol{\tau} + \sum_i \boldsymbol{J}_{C,i}^T \boldsymbol{F}_{C,i} \end{aligned} \tag{6.21}$$

6.2 Graph Theoretic Modeling of ASLB

Separating legs, which are defined as **Agent 2** and **Agent 3**, from the floating base introduces simplifications to the overall complexity of the model and optimization. One of the simplifications appears in modeling as leg dynamics is not necessarily needs to be modeled with respect to the $\mathcal{F}_I$ using $\boldsymbol{q}_B$ but rather preferred to be defined with respect to the $\mathcal{F}_B$. This local representation of leg dynamics leads to following outcomes for the dynamics of the agents:

- Leg is only used to find adjacency and contact forces on the floating base and ground,
- State space representation of the leg dynamics can be kept at velocity level.
- Contact forces and geometric properties of contact point must be converted to $\mathcal{F}_B$

Following subsections explain underlying Graph structure, kinematics, and dynamics of the agents.

6.2.1 Underlying Graph Structure

The interaction among agents in the cooperative systems are described by Graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, which consists of a node set $\mathcal{N}$ and an edge set $\mathcal{E}$ [109]. Edge set $\mathcal{E} \subset \mathcal{N} \times \mathcal{N}$ is given between nodes $m \in \mathcal{N}$ and $n \in \mathcal{N}$ such that $(n, m) \in \mathcal{E}$ denotes node m receives information from n. Adjacency matrix is composed of edge weights such that $\mathbb{A} = [a_{mn}] \in \mathbb{R}^{N \times N}$ of $\mathcal{G}$, where a_{mn} is the strength of the connection between node n and node m. N is the number agents in the cooperative system (CS). A_{mn} represents element in the m^{th} row n^{th} column of matrix $\boldsymbol{A}$. Formal description of $\mathbb{A}_{mn}$ is given as follows.

$$\mathbb{A}_{mn} = \begin{cases} a_{mn} = 1, & n \neq m, \ \ (n, m) \in \mathcal{E} \\ a_{mn} = 0, & otherwise \end{cases} \tag{6.22}$$

Floating base of the robot is denoted as $\mathbf{Agent\ 1}$, right leg is denoted as $\mathbf{Agent\ 2}$ and left leg is denoted as $\mathbf{Agent\ 3}$. $\mathbf{Agent\ 1}$ is defined with two nodes that are $\mathbf{node\ 1}$

and $\mathbf{node\ 2}$, $\mathbf{Agent\ 2}$ is defined with $\mathbf{node\ 3}$ and $\mathbf{Agent\ 3}$ is defined with $\mathbf{node\ 4}$. The location of these nodes are illustrated in Fig. 6.5 and it should be noted that $\mathbf{node\ 1}$ and $\mathbf{node\ 2}$ are coincident. Without any interconnection constraint, agents are independent of each other , however, there are rigid joints connecting them. In this case there are two bi-directional edges and these are $(\mathbf{node\ 3}, \mathbf{node\ 1}) \in \mathcal{E}$ and $(\mathbf{node\ 2}, \mathbf{node\ 4}) \in \mathcal{E}$. Connection between nodes are assumed to be rigid then adjacency matrix is composed of edge weights $a_{mn} = 1$. This yields an adjacency matrix as in (6.23).

$$\mathbb{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{6.23}$$

Finally, relationship between cooperative agents are given in (6.24) by Laplacian matrix. $\boldsymbol{I}_p$ represents identity matrix with size p.

$$\mathbb{L} = \boldsymbol{I}_4 - \mathbb{A} \tag{6.24}$$

Let $\boldsymbol{W} \in \mathbb{R}^p$ be the signal that is being shared between agents then $\mathbb{L}$ can be expanded as in (6.25) to comply with the signal dimension. $\otimes$ is the Kronecker product operator.

$$\boldsymbol{L} = \mathbb{L} \otimes \boldsymbol{I}_p \tag{6.25}$$

Recalling that $\mathbf{node\ 1}$ and $\mathbf{node\ 2}$ are coincident, one can define following adjacency constraints using the extended Laplacian definition given in (6.25)

$$\begin{aligned} \mathcal{L}_x &= \boldsymbol{L} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \end{bmatrix} = \mathbf{0} \\ \mathcal{L}_w &= \boldsymbol{L} \begin{bmatrix} \boldsymbol{W}_{A,1} \\ \boldsymbol{W}_{A,2} \\ \boldsymbol{W}_{A,3} \\ \boldsymbol{W}_{A,4} \end{bmatrix} = \mathbf{0} \end{aligned} \tag{6.26}$$

Another aspect of splitting a lumped multi-body model into distributed cooperating multi-body models is generalized coordinates. This operation necessarily duplicates the generalized coordinates of the floating base in lumped model to the distributed models. In addition to that $\mathbf{Agent\ 2}$ and $\mathbf{Agent\ 3}$ have actuated joints of $\boldsymbol{q}_{a,i}$, $i = 2, 3$. Due to this, generalized coordinates of the agents are defined as $\boldsymbol{q}_1 = \begin{bmatrix} \boldsymbol{q}_{t,1}^T \ \boldsymbol{q}_{r,1}^T \end{bmatrix}^T \in \mathbb{R}^6$, $\boldsymbol{q}_2 = \begin{bmatrix} \boldsymbol{q}_{t,2}^T \ \boldsymbol{q}_{r,2}^T \ \boldsymbol{q}_{a,2}^T \end{bmatrix}^T \in \mathbb{R}^9$ and $\boldsymbol{q}_3 = \begin{bmatrix} \boldsymbol{q}_{t,3}^T \ \boldsymbol{q}_{r,3}^T \ \boldsymbol{q}_{a,3}^T \end{bmatrix}^T \in \mathbb{R}^9$, respectively for $\mathbf{Agent\ 1}$, $\mathbf{Agent\ 2}$ and $\mathbf{Agent\ 3}$. This is illustrated in Fig. 6.5.

6.2.2 Agent Kinematics

YZX Euler sequence is used in defining the composite rotations $\boldsymbol{C}_{Ii}(\boldsymbol{q}_{r,i})$ from $\mathcal{F}_I$ to $\mathcal{F}_i$, where subscript i is agent index and $\mathcal{F}_i$ is the local origin of the agent. Contact point position and velocity of $\mathbf{Agent\ 2}$ and $\mathbf{Agent\ 3}$ are calculated with respect to $\mathcal{F}_B$, which is denoted as in (6.27). $\mathcal{K}^+(\boldsymbol{q}_i)$ represents forward kinematics of the leg i in Eq. (6.27), which is explained in § 6.1.2.

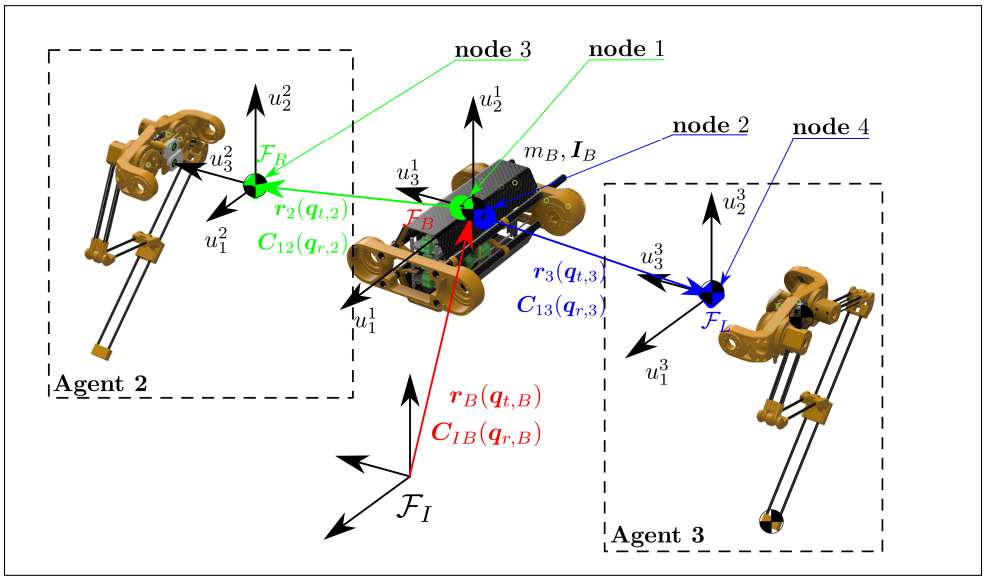


Figure 6.5. Cooperative interconnection between agents and generalized coordinates at every agent..

$$\begin{aligned} \boldsymbol{r}_{c,i}(\boldsymbol{q}_i) &= \mathcal{K}^+(\boldsymbol{q}_i) \\ \dot{\boldsymbol{r}}_{c,i}(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i) &= \boldsymbol{J}_{c,i}\dot{\boldsymbol{q}}_i \end{aligned} \tag{6.27}$$

6.2.3 Agent Dynamics

Dynamics of agents yield a similar equation as given in (6.21) and for brevity a representative Equation of Motion (EoM) is given in this section.

Generalized velocities are assigned to states of each agent as $\boldsymbol{x}_i = \dot{\boldsymbol{q}}_i$. External forces in distributed notation is divided into two, where first one is denoted as $\boldsymbol{F}_{C,i}$ and acting on the agents as a result of ground contact. The second external force is denoted as $\boldsymbol{F}_{A,m}$ and exerted on the agents from the adjacent nodes. Adjacent nodes also transmit moment, $\boldsymbol{M}_{A,m}$, therefore it is convenient to collect forces and moments at adjacent nodes as a wrench denoted as $\boldsymbol{W}_{A,m} = \left[\boldsymbol{F}_{A,m}^T \ \boldsymbol{M}_{A,m}^T\right]^T$. As a result, non-linear dynamics of each agent is written as given in (6.28). Let $\boldsymbol{x}_i = \dot{\boldsymbol{q}} \in \mathbb{R}^{n_{xi}}$, $\boldsymbol{u}_{\tau,i} = \boldsymbol{\tau} \in \mathbb{R}^{n_{ti}}$, $\boldsymbol{u}_{c,i} = \boldsymbol{F}_{C,i} \in \mathbb{R}^{n_{fi}}$, and $\boldsymbol{u}_{w,i} = \boldsymbol{W}_{A,i} \in \mathbb{R}^{n_{wi}}$ are the states, torques, contact forces,

and adjacency wrench, respectively, then non-linear dynamics of the robot can be written as in Eq. 6.21. Signal sizes for **Agent 1** is $n_{x1} = 6$, $n_{t1} = 0$, $n_{f1} = 0$, $n_{w1} = 6$, while **Agent 2** and **Agent 3** has signal sizes of $n_{xi} = 9$, $n_{ti} = 3$, $n_{fi} = 3$, $n_{wi} = 6$ for $i = 2, 3$.

$$\begin{aligned} \dot{\boldsymbol{x}}_i &= \boldsymbol{f}_i(\boldsymbol{q}_i, \boldsymbol{u}_{\tau,i}, \boldsymbol{u}_{c,i}) \\ \boldsymbol{f}_i &= \boldsymbol{M}_i(\boldsymbol{q}_i)^{-1}\boldsymbol{\Theta}_i \\ \boldsymbol{\Theta}_i &= -\boldsymbol{C}_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i) - \boldsymbol{G}_i(\boldsymbol{q}_i) \\ &\quad + \boldsymbol{S}_i\boldsymbol{\tau}_i + \boldsymbol{J}_{C,i}^T\boldsymbol{F}_{C,i} + \sum_m \boldsymbol{J}_{A,m}^T\boldsymbol{W}_{A,m} \end{aligned} \tag{6.28}$$

6.3 Graph Theoretic Online Trajectory Generation

Graph theoretic online trajectory generation relies on cooperative modeling of the robot and composed of a series of optimizations. These optimizations are, contact phase, swing phase and force optimizations, where contact optimization finds optimal finite horizon for the current contact phase and a sequence of contact trajectories that will keep the robot states bounded for defined phase horizon. Due to this reason, resultant contact phase trajectories except for the one associated with the current contact leg are not passed to the next optimization. Another cardinal data set that is passed to the next optimization from contact phase is the initial point of the subsequent contact phase. This information is required to generate a rough swing trajectory for the leg that is in the air. It should be noted that, both contact and swing trajectories are calculated to provide an initial trajectory for the force optimization, where trajectories are refined using cooperative system framework. Regarding trajectories to previously defined optimizations are illustrated in Figure 6.6.

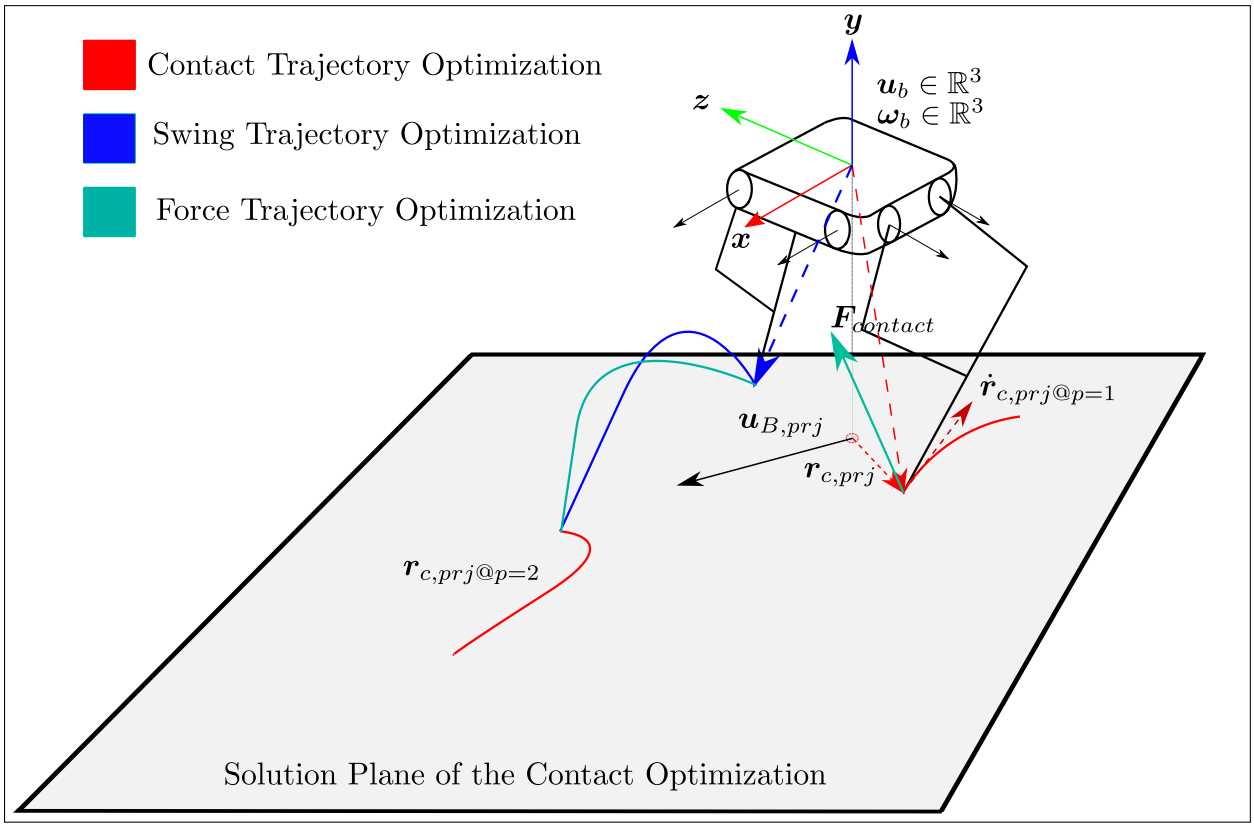


Figure 6.6. Cooperative interconnection between agents and generalized coordinates at every agent..

6.3.1 Contact Phase Optimization

Contact phase optimization calculates a set of trajectories using Linear Inverted Pendulum Model (LIPM) and contact constraints. LIPM dynamics we used in this work is widely used in vast majority of the literature. As denoted in § 6.2, dynamics are written with respect to body frame $\mathcal{F}_B$. Besides that, dynamics are also kept at velocity level. Under these circumstances the contact conditions for the leg in contact need to be defined accordingly.

Contact Condition Under contact conditions with no slip assumption, $r_{c,i}$ has no relative motion with respect to the ground if this condition is observed from the inertial frame

$\mathcal{F}_I$. This condition is observed from $\mathcal{F}_B$ as if $\dot{\boldsymbol{r}}_{c,i} = -\boldsymbol{u}_B$, which is illustrated in Eq. (6.29) for agents $i = 2, 3$. Recall that states of agents $i = 2, 3$ are denoted as $\boldsymbol{x}_i = \left[\boldsymbol{q}_{t,i}^T \, \boldsymbol{q}_{r,i}^T \, \boldsymbol{q}_{a,i}^T\right]^T$, $i = 2, 3$ and $\dot{\boldsymbol{q}}_{t,i}$, $i = 2, 3$ is duplicate of $\boldsymbol{u}_B$ assuming that connection between nodes are as defined in Eq. (6.22).

$$\mathbf{0} = \begin{bmatrix} -\boldsymbol{I} & -\boldsymbol{J}_{r,i} & \boldsymbol{J}_{a,i} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{q}}_{t,i} \\ \dot{\boldsymbol{q}}_{r,i} \\ \dot{\boldsymbol{q}}_{a,i} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{I} & -\boldsymbol{J}_{r,i} & \boldsymbol{J}_{a,i} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_B \\ \dot{\boldsymbol{q}}_{r,i} \\ \dot{\boldsymbol{q}}_{a,i} \end{bmatrix} \tag{6.29}$$

Assuming body is slowly rotating, $\dot{\boldsymbol{q}}_{r,i} \approx \mathbf{0}$, $i = 2, 3$, Eq. (6.29)

LIPM Model Implementation of the LIPM model in this work have some nuances compared to the the work where it is proposed [139, 140]. Let $\boldsymbol{F}_{xz} \in \mathbb{R}^2$ be the virtual force created on $x - z$ plane due to the displacement between origin and contact locations ${}_{xz}\boldsymbol{r}_{c,2}$ and ${}_{xz}\boldsymbol{r}_{c,3}$. Note that origin represents the center of mass (CoM) and is not the origin of the $\mathcal{F}_B$. Although CoM moves with respect to the origin of $\mathcal{F}_B$, in practice it is assumed to be fixed with an offset from $\mathcal{F}_B$.

Under these assumptions the LIPM dynamics is provided as in Eq. (6.30) in state space form. States for this system is denoted as $\boldsymbol{x}_c$ and defined as the contact point and this point is denoted as $\boldsymbol{\Delta r}_{xz}$ as any of the two contact locations can be assigned to it, which are ${}_{xz}\boldsymbol{r}_{c,2}$ and ${}_{xz}\boldsymbol{r}_{c,3}$, respectively. Explicitly states are defined as $\overline{\boldsymbol{x}}_{c,1} = \boldsymbol{\Delta r}_{xz} \in \mathbb{R}^2$, $\overline{\boldsymbol{x}}_{c,2} = \dot{\boldsymbol{\Delta r}}_{xz} \in \mathbb{R}^2$, which are combined as $\overline{\boldsymbol{x}}_c = \left[\overline{\boldsymbol{x}}_{c,1}^T \; \overline{\boldsymbol{x}}_{c,2}^T\right]^T$.

$$\dot{\overline{\boldsymbol{x}}}_c = \begin{bmatrix} \dot{\overline{\boldsymbol{x}}}_{c,1} \\ \dot{\overline{\boldsymbol{x}}}_{c,2} \end{bmatrix} = \left[\begin{array}{c|c} \mathbf{0} & \boldsymbol{I} \\ \hline \boldsymbol{I}\omega_0^2 & \mathbf{0} \end{array}\right] \begin{bmatrix} \overline{\boldsymbol{x}}_{c,1} \\ \overline{\boldsymbol{x}}_{c,2} \end{bmatrix} = \boldsymbol{A}_{lipm}\overline{\boldsymbol{x}}_c \tag{6.30}$$

This system is an inherently unstable system therefore what is being pursued with this system is to find a set of initial conditions denoted as ${}^0\overline{\boldsymbol{x}}_c$, that will propel the CoM of the robot towards the desired velocity vector. While doing that, a set of state bounds

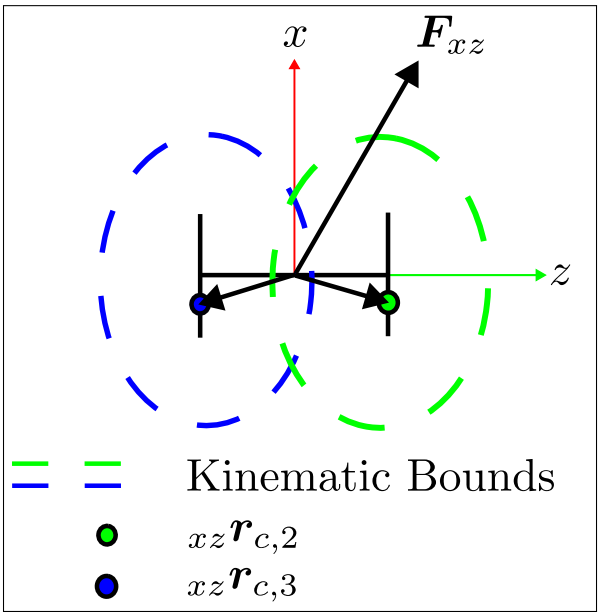


Figure 6.7. LIPM dynamics projected on $x - z$ plane.

are also satisfied. Equation (6.30) is dicretized using Euler propogation as provided below, where Δt is the sampling time.

$$\overline{\boldsymbol{x}}_{c,k+1} = (\boldsymbol{I} + \Delta t \boldsymbol{A}_{lipm})\overline{\boldsymbol{x}}_{c,k} \tag{6.31}$$

Contact Phase Optimization This method relies on finding set of trajectories that will keep proceeding steps within bounds, therefore a phase horizon is defined as $N_p \in \mathbb{N}$, which represents the number of phases to be calculated during the optimization inlcuding the current phase. A finite horizon for each phase is defined as $N_n \in \mathbb{N}$, which will be minimized for $N_p = 1$ and kept at its nominal for $N_p > 1$. States of the LIPM dynamics in each phase are denoted as ${}^{p}\overline{\boldsymbol{x}}_{c,k}$, where $p = [1, \cdots, N_p]$ and $k = [1, \cdots, N_n]$. Combined states for each phase are denoted as ${}^{p}\boldsymbol{x}_{c,K}$ as provided below.

$$
{}^{p}\overline{\boldsymbol{x}}_{c,K} \;=\; \begin{bmatrix} {}^{p}\overline{\boldsymbol{x}}_{c,1} \\ \vdots \\ {}^{p}\overline{\boldsymbol{x}}_{c,N_n} \end{bmatrix}, \;\; \forall\, p \tag{6.32}
$$

Equation (6.33) represents the quadratic problem that runs in contact optimization phase, where $\mathcal{L}_c({}^{p}\overline{\boldsymbol{x}}_{c,1},\; \boldsymbol{u}_{B,ref})$ is the objective function, and $\mathcal{C}_{cont}({}^{p}\overline{\boldsymbol{x}}_{c,1})$ and $\mathcal{C}_{bounds}({}^{p}\overline{\boldsymbol{x}}_{c,1})$ are equality and inequality contstraints to ensure continuity of the states between phases and to keep states within predefined bounds.

$$
\begin{aligned}
\underset{{}^{p}\overline{\boldsymbol{x}}_{c,1}}{\text{minimize}} \quad & \sum_{k=1}^{M} \mathcal{L}_c({}^{p}\overline{\boldsymbol{x}}_{c,1},\; \boldsymbol{u}_{B,ref}) \\
\text{s. t.} \quad & \mathcal{C}_{cont}({}^{p}\overline{\boldsymbol{x}}_{c,1}) \;=\; \mathbf{0} \\
& \mathcal{C}_{bounds}({}^{p}\overline{\boldsymbol{x}}_{c,1}) \;\leq\; \mathbf{0}
\end{aligned} \tag{6.33}
$$

Continuity Constraint As explained earlier, contact optimization seeks to find several contact phase trajectories and these trajectories should be continuous ideally. This is achieved by an equaltiy constraint defined as in Eq. (6.34) for $p \;=\; 1,\; \cdots,\; N_p - 1$.

$$
\begin{aligned}
{}^{p}\overline{\boldsymbol{x}}_{c,N_n} \;&=\; {}^{p+1}\overline{\boldsymbol{x}}_{c,1} \\
\begin{bmatrix} \mathbf{0} & \boldsymbol{I} \end{bmatrix} {}^{p}\overline{\boldsymbol{x}}_{c,N_n} \;&=\; \begin{bmatrix} \mathbf{0} & \boldsymbol{I} \end{bmatrix} {}^{p+1}\overline{\boldsymbol{x}}_{c,1}
\end{aligned} \tag{6.34}
$$

Constraint for State Bounds State bounds are defined based on the leg in contact, therefore state bounds are switching between the bounds of **Agent 2** and **Agent 3**. Let $c_2 \;=\; [0, 1]$ and $c_3 \;=\; [0, 1]$ be the contact indicators of **Agent 2** and **Agent 3**, respectively. Under contact $c_i \;=\; 1$ otherwise $c_i \;=\; 0$ for $i = 2, 3$ at phase p. Bounds for **Agent 2** and **Agent 3** are defined as $\boldsymbol{S}_i \;=\; \left[\boldsymbol{S}_{i,UB}^{T},\; \boldsymbol{S}_{i,LB}^{T}\right]^{T}$ and assigned to $\boldsymbol{S}_p$ such that $\boldsymbol{S}_p \;=\; \boldsymbol{S}_i$ if $c_i = 1$. Note that we are assuming single point contact. Using the bounds state boundary constraints are defined as in Eq. (6.35).

$$\begin{bmatrix} \boldsymbol{I} \\ -\boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \end{bmatrix} {}^{p}\overline{\boldsymbol{x}}_{c,k} \leq \begin{bmatrix} \boldsymbol{S}_{p,UB} \\ \boldsymbol{S}_{p,LB} \end{bmatrix} \tag{6.35}$$

Cost Function Contact phase optimization desires to reach the $x-z$ projection of reference velocity that is provided by the user $\boldsymbol{u}_{B,ref}$, which is $\boldsymbol{u}_{B,xz}$. Recall that $\dot{\boldsymbol{r}}_{c,i} = -\boldsymbol{u}_B$, therefore, cost function is written as in Eq. (6.36).

$$\begin{aligned} J_{contact} = & \left[\boldsymbol{u}_{B,xz} + \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} {}^{p}\overline{\boldsymbol{x}}_{c,K}\right]^{T} \boldsymbol{Q}_s \left[\boldsymbol{u}_{B,xz} + \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} {}^{p}\overline{\boldsymbol{x}}_{c,K}\right]^{T} + \\ & \left[\begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} {}^{p}\overline{\boldsymbol{x}}_{c,K}\right]^{T} \boldsymbol{Q}_p \left[\begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} {}^{p}\overline{\boldsymbol{x}}_{c,K}\right]^{T} \end{aligned} \tag{6.36}$$

6.3.2 Swing Phase Optimization

Swing pahse optimization calculates a rough trajectory for the swinging leg by connecting current position of the tip of the swinging leg to the initial point of the proceeding contact phase trajectory with a bezier curve. Instances of the swing trajectory is denoted as ${}^{p}\boldsymbol{x}_{s,k}$, current and final positions of the swing trajectory are denoted as ${}^{p}\boldsymbol{x}_{s,1}$ and ${}^{p}\boldsymbol{x}_{s,N_n}$, respectively. ${}^{p}\overline{\boldsymbol{x}}_{s,1}$ and ${}^{p}\overline{\boldsymbol{x}}_{s,N_n}$ are the projections of the vectors on $x-z$ plane, and initial point of the proceeding contact phase trajectory is ${}^{p+1}\overline{\boldsymbol{x}}_{c,1}$. Note that, contact phase optimization generates a planar trajectory. Therefore, implementation to swing phase trajectory optimization requires a modification of these vectors by adding height of the CoM to y axis of any projected vector if it needs to be passed to swing trajectory. Figure 6.8 illustrates previously mentioned vectors. Dashed lines represent projection of the swing trajectory, while solid lines show contact trajectory. $\boldsymbol{n}_0$, $\boldsymbol{n}_f$ and $\boldsymbol{L}$ represents unit vector to CoM, unit vector from CoM, and straight line on $x-z$ plane between initial and final positions of the swing trajectory. Formal definitions for $\boldsymbol{n}_0$, $\boldsymbol{n}_f$ and $\boldsymbol{L}$ are provided in Eq. (6.37).

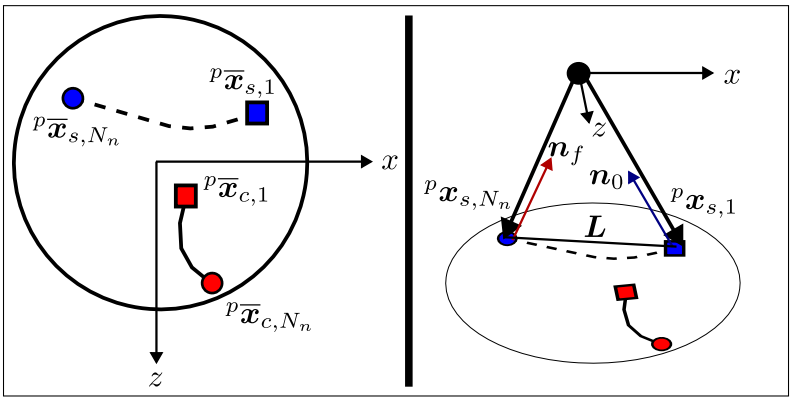


Figure 6.8. Swing phase trajectory generation.

$$\begin{aligned} \boldsymbol{n}_0 &= -\frac{^{p=1}\boldsymbol{x}_{s,1}}{|^{p=1}\boldsymbol{x}_{s,1}|} \\ \boldsymbol{n}_f &= \frac{^{p=1}\boldsymbol{x}_{s,N_n}}{|^{p=1}\boldsymbol{x}_{s,N_n}|} \\ \boldsymbol{L} &= {}^{p=1}\boldsymbol{x}_{s,N_n} - {}^{p=1}\boldsymbol{x}_{s,1} \end{aligned} \tag{6.37}$$

Swing Phase Optimization Swing phase optimization is run for the current phase, therefore unlike the contact phase optimization $p = 1$. Finite horizon for this phase is the N_n at $p = 1$. Note that at $p = 1$, N_n is being optimized at contact phase optimization.

Let $b_{x,j}$, $b_{y,j}$, and $b_{z,j}$ be the coefficients of the bezier curve, where $j = 1, \cdots, nb$. Define $\boldsymbol{c}_b$ and $\boldsymbol{v}_b$, which are sorted collections of $b_{x,j}$, $b_{y,j}$, and $b_{z,j}$. This classification collects coefficients related to initial and final positions of the curve under $\boldsymbol{c}_b$ and coefficients thatare being optimized under $\boldsymbol{v}_b$. Specifically $\boldsymbol{c}_b$ is defined as,

$$\boldsymbol{c}_b = \begin{bmatrix} ^{p=1}\boldsymbol{x}_{s,1}^T & ^{p=1}\boldsymbol{x}_{s,N_n} \end{bmatrix}^T \tag{6.38}$$

. Based on previously described notation, Bezier curves for swing trajectory are defined as in Eq.(6.40), where $\boldsymbol{J}_{b,c}$ and $\boldsymbol{J}_{b,v}$ are matrix valued functions of k, which can be populated for $k = 1 < \cdots, N_n$ and maps $\boldsymbol{c}_b$ and $\boldsymbol{v}_b$ to $^{p=1}\boldsymbol{x}_{s,k} \in \mathbb{R}^3$. Similarly, $\boldsymbol{c}_b$ and $\boldsymbol{v}_b$ are mapped to $^{p=1}\dot{\boldsymbol{x}}_{s,k} \in \mathbb{R}^3$ using ${}_d\boldsymbol{J}_{b,c}$ and ${}_d\boldsymbol{J}_{b,v}$.

Equation (6.39) represents the quadratic problem that runs in swing optimization phase, where $\mathcal{L}_s(\boldsymbol{v}_b)$ is the objective function, and $\mathcal{S}_{bounds}(\boldsymbol{v}_b)$ is the set of inequality contstraints to keep states within predefined bounds.

$$\begin{aligned} \underset{\boldsymbol{v}_b}{\text{minimize}} \quad & \sum_{k=1}^{M} \mathcal{L}_s(\boldsymbol{v}_b) \\ \text{s. t.} \quad & \mathcal{C}_{bounds}(\boldsymbol{v}_b) \leq \mathbf{0} \end{aligned} \tag{6.39}$$

$$\begin{aligned} {}^{p=1}\boldsymbol{x}_{s,k} &= \begin{bmatrix} \boldsymbol{J}_{b,c}(k) & \boldsymbol{J}_{b,v}(k) \end{bmatrix} \begin{bmatrix} \boldsymbol{c}_b \\ \boldsymbol{v}_b \end{bmatrix} \\ {}^{p=1}\dot{\boldsymbol{x}}_{s,k} &= \begin{bmatrix} {}_d\boldsymbol{J}_{b,c}(k) & {}_d\boldsymbol{J}_{b,v}(k) \end{bmatrix} \begin{bmatrix} \boldsymbol{c}_b \\ \boldsymbol{v}_b \end{bmatrix} \end{aligned} \tag{6.40}$$

Implementation of swing phase optimization requires calculation of ${}^0_d\boldsymbol{J}_{b,c}$, ${}^f_d\boldsymbol{J}_{b,c}$, ${}^0_d\boldsymbol{J}_{b,v}$, ${}^f_d\boldsymbol{J}_{b,v}$ using Eq. (6.40) at $k = 1$ and $k = N_n$.

Constraint for State Bounds State bounds are defined based on the leg in swing, therefore state bounds are switching between the bounds of **Agent 2** and **Agent 3**. Bounds for **Agent 2** and **Agent 3** are defined as $\boldsymbol{S}_i = \left[\boldsymbol{S}_{i,UB}^T,\ \boldsymbol{S}_{i,LB}^T\right]^T$ and assigned to $\boldsymbol{S}_p$ such that $\boldsymbol{S}_p = \boldsymbol{S}_i$ if $c_i = 0$. Using the bounds state boundary constraints are defined as in Eq. (6.41).

$$\begin{aligned} \begin{bmatrix} \boldsymbol{J}_{b,c}(k)\boldsymbol{c}_b + \boldsymbol{J}_{b,v}(k)\boldsymbol{v}_b \\ -\boldsymbol{J}_{b,c}(k)\boldsymbol{c}_b - \boldsymbol{J}_{b,v}(k)\boldsymbol{v}_b \end{bmatrix} &\leq \boldsymbol{S}_p \\ \begin{bmatrix} \boldsymbol{J}_{b,v}(k) \\ -\boldsymbol{J}_{b,v}(k) \end{bmatrix} \boldsymbol{v}_b &\leq \boldsymbol{S}_p - \begin{bmatrix} \boldsymbol{J}_{b,c}(k) \\ -\boldsymbol{J}_{b,c}(k) \end{bmatrix} \boldsymbol{c}_b \end{aligned} \tag{6.41}$$

Cost Function Swing phase optimization desires to pull swinging leg to the CoM in the beginning of the swing motion and then pushes it towards the final position. Along with these, it also tries to approach the straight line $\boldsymbol{L}$. Therefore, cost function of swing phase optimization is written as in Eq. (6.42).

$$
\begin{aligned}
J_{swing} \;=\; & \left(\boldsymbol{n}_0 - \begin{bmatrix} {}^{0}_{d}\boldsymbol{J}_{b,c} & {}^{0}_{d}\boldsymbol{J}_{b,v} \end{bmatrix} \begin{bmatrix} \boldsymbol{c}_b \\ \boldsymbol{v}_b \end{bmatrix} \right)^T \boldsymbol{Q}_0 \left(\boldsymbol{n}_0 - \begin{bmatrix} {}^{0}_{d}\boldsymbol{J}_{b,c} & {}^{0}_{d}\boldsymbol{J}_{b,v} \end{bmatrix} \begin{bmatrix} \boldsymbol{c}_b \\ \boldsymbol{v}_b \end{bmatrix} \right) + \\
& \left(\boldsymbol{n}_f - \begin{bmatrix} {}^{f}_{d}\boldsymbol{J}_{b,c} & {}^{f}_{d}\boldsymbol{J}_{b,v} \end{bmatrix} \begin{bmatrix} \boldsymbol{c}_b \\ \boldsymbol{v}_b \end{bmatrix} \right)^T \boldsymbol{Q}_f \left(\boldsymbol{n}_f - \begin{bmatrix} {}^{f}_{d}\boldsymbol{J}_{b,c} & {}^{f}_{d}\boldsymbol{J}_{b,v} \end{bmatrix} \begin{bmatrix} \boldsymbol{c}_b \\ \boldsymbol{v}_b \end{bmatrix} \right) + \\
& \left(\boldsymbol{L} - \begin{bmatrix} \boldsymbol{J}_{b,c} & \boldsymbol{J}_{b,v} \end{bmatrix} \begin{bmatrix} \boldsymbol{c}_b \\ \boldsymbol{v}_b \end{bmatrix} \right)^T \boldsymbol{Q}_t \left(\boldsymbol{L} - \begin{bmatrix} \boldsymbol{J}_{b,c} & \boldsymbol{J}_{b,v} \end{bmatrix} \begin{bmatrix} \boldsymbol{c}_b \\ \boldsymbol{v}_b \end{bmatrix} \right)
\end{aligned}
\tag{6.42}
$$

6.3.3 Cooperative Force Optimization

Approximate trajectories are obained in contact and swing phase optimizations and these trajectories are used in cooperative force optimization as initial trajectory. In order to follow the method easily agent dynamics are rewritten in Eq. (6.44), where $\boldsymbol{M}_h$ and $\boldsymbol{C}_h$ are partitioned as given in Eq. (6.43). In addition to that $\boldsymbol{M}_i$, $\boldsymbol{C}_i$, and $\boldsymbol{G}_i$ for $i = 2, 3$ are assigned to $\boldsymbol{M}_h$, $\boldsymbol{C}_h$ and $\boldsymbol{G}_h$, where $h = S$ represents swing, $h = C$ represents contact and $h = B$ represents floating base matrices. Minimal representation of the dynamics are provided in Eq. (6.45). Floating base dynamics do not switch, however, **Agent 2** and **Agent 3** dynamics are assigned to $h = C$ or $h = S$ depending on c_i for $i = 2, 3$. Similarly, wrenches $\boldsymbol{W}_{A,m}$ are assigned to $\boldsymbol{W}_{A,h}$ based on c_i such that if $c_2 = 1$, then $\boldsymbol{W}_{A,C} = \boldsymbol{W}_{A,2}$

and if $c_3 = 1$ then $\boldsymbol{W}_{A,C} = \boldsymbol{W}_{A,3}$ for $i = 2, 3$. Finally, $\tilde{\boldsymbol{r}}_C$ is the tip point position $\boldsymbol{r}_{c,i}$ of the leg with $c_i = 1$. $\boldsymbol{F}_C$ is the interaction between contact leg and ground.

$$\begin{aligned} \boldsymbol{M}_h &= \left[\begin{array}{c|c} {}_{bb}\boldsymbol{M}_h & {}_{bq}\boldsymbol{M}_h \\ \hline {}_{qb}\boldsymbol{M}_h & {}_{qq}\boldsymbol{M}_h \end{array}\right] \\ \boldsymbol{C}_h &= \left[\begin{array}{c|c} {}_{bb}\boldsymbol{C}_h & {}_{bq}\boldsymbol{C}_h \\ \hline {}_{qb}\boldsymbol{C}_h & {}_{qq}\boldsymbol{C}_h \end{array}\right] \end{aligned} \tag{6.43}$$

$$\begin{aligned} {}_{bb}\boldsymbol{M}_B\dot{\boldsymbol{x}}_B + {}_{bb}\boldsymbol{C}_B\boldsymbol{x}_B + {}_{bb}\boldsymbol{G}_B &= \overline{\boldsymbol{W}}_{A,C} + \overline{\boldsymbol{W}}_{A,S} \\ \left[\begin{array}{c|c} {}_{bb}\boldsymbol{M}_S & {}_{bq}\boldsymbol{M}_S \end{array}\right]\dot{\boldsymbol{x}}_S + \left[\begin{array}{c|c} {}_{bb}\boldsymbol{C}_S & {}_{bq}\boldsymbol{C}_S \end{array}\right]\boldsymbol{x}_S + {}_{bb}\boldsymbol{G}_S &= \boldsymbol{W}_{A,S} \\ \left[\begin{array}{c|c} {}_{bb}\boldsymbol{M}_C & {}_{bq}\boldsymbol{M}_C \end{array}\right]\dot{\boldsymbol{x}}_C + \left[\begin{array}{c|c} {}_{bb}\boldsymbol{C}_C & {}_{bq}\boldsymbol{C}_C \end{array}\right]\boldsymbol{x}_S + {}_{bb}\boldsymbol{G}_C &= \boldsymbol{W}_{A,C} + \left[\begin{array}{c} \boldsymbol{I} \\ \hline \tilde{\boldsymbol{r}}_C \end{array}\right]\boldsymbol{F}_C \end{aligned} \tag{6.44}$$

$$\begin{aligned} \hat{\boldsymbol{M}}_B\dot{\boldsymbol{x}}_B + \hat{\boldsymbol{C}}_B\boldsymbol{x}_B + \hat{\boldsymbol{G}}_B &= \overline{\boldsymbol{W}}_{A,C} + \overline{\boldsymbol{W}}_{A,S} \\ \hat{\boldsymbol{M}}_S\dot{\boldsymbol{x}}_S + \hat{\boldsymbol{C}}_S\boldsymbol{x}_S + \hat{\boldsymbol{G}}_S &= \boldsymbol{W}_{A,S} \\ \hat{\boldsymbol{M}}_C\dot{\boldsymbol{x}}_C + \hat{\boldsymbol{C}}_C\boldsymbol{x}_S + \hat{\boldsymbol{G}}_C &= \boldsymbol{W}_{A,C} + \left[\begin{array}{c} \boldsymbol{I} \\ \hline \tilde{\boldsymbol{r}}_C \end{array}\right]\boldsymbol{F}_C \end{aligned} \tag{6.45}$$

Then continuous models in Eq. (6.44) are converted to discrete models using Euler discretization and Eq. (6.46) provides the discrete system model that is used in cooperative force optimization. Current states are denoted as $\boldsymbol{x}_{h,k}$, where h and k represent model identifier and prediction step, respectively.

$$\begin{bmatrix} \Delta t\hat{\boldsymbol{C}}_{B,k} - \hat{\boldsymbol{M}}_{B,k} & \hat{\boldsymbol{M}}_{B,k} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{B,k} \\ \boldsymbol{x}_{B,k+1} \end{bmatrix} - \Delta t\overline{\boldsymbol{W}}_{S,k} - \Delta t\overline{\boldsymbol{W}}_{C,k} = \Delta t\hat{\boldsymbol{G}}_B$$

$$\begin{bmatrix} \Delta t\hat{\boldsymbol{C}}_{S,k} - \hat{\boldsymbol{M}}_{S,k} & \hat{\boldsymbol{M}}_{S,k} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{S,k} \\ \boldsymbol{x}_{S,k+1} \end{bmatrix} - \Delta t\boldsymbol{W}_{S,k} = \Delta t\hat{\boldsymbol{G}}_{S,k} \quad (6.46)$$

$$\begin{bmatrix} \Delta t\hat{\boldsymbol{C}}_{C,k} - \hat{\boldsymbol{M}}_{C,k} & \hat{\boldsymbol{M}}_{C,k} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{C,k} \\ \boldsymbol{x}_{C,k+1} \end{bmatrix} - \Delta t\boldsymbol{W}_{C,k} - \Delta t\hat{\boldsymbol{J}}_{C,k}\boldsymbol{F}_{C,k} = \Delta t\hat{\boldsymbol{G}}_{C,k}$$

For brevity Eq. (6.46) are represented with a minimal representation as follows.

$$\begin{aligned} \tilde{\boldsymbol{M}}_{h,k} &= \begin{bmatrix} \tilde{\boldsymbol{M}}_{h_1,k} & \tilde{\boldsymbol{M}}_{h_2,k} \end{bmatrix} = \begin{bmatrix} \Delta t\hat{\boldsymbol{C}}_{B,k} - \hat{\boldsymbol{M}}_{B,k} & \hat{\boldsymbol{M}}_{B,k} \end{bmatrix} \\ \tilde{\boldsymbol{J}}_{C,k} &= -\Delta t\hat{\boldsymbol{J}}_{C,k} \\ \tilde{\boldsymbol{G}}_{h,k} &= \Delta t\hat{\boldsymbol{G}}_{h,k} \\ \tilde{\boldsymbol{P}}_{h,k} &= -\Delta t\hat{\boldsymbol{I}} \end{aligned} \quad (6.47)$$

Cooperative Force Optimization Problem This method relies on initially provided trajectories that is provided by contact and swing phase. In this phase, decision variables are defined as corrections to the nominal trajectories and a complete trajectory is defined such that. Nominal trajectories for states are denoted with ${}_0\boldsymbol{x}_{h,k}$ and corrections to the nominal trajectories at every intant is denoted as $\Delta\boldsymbol{x}_{h,k}$. Similarly, force trajectorries are defined in the same fashion such that ${}_0\boldsymbol{F}_{C,k}$ and ${}_0\boldsymbol{W}_{A,m}$ are the nominal force trajectories, while $\Delta\boldsymbol{F}_{C,k}$ and $\Delta\boldsymbol{W}_{A,m}$ are the corrections to regarding trajectories.

$$\begin{aligned} \boldsymbol{x}_{h,k}^T &= {}_0\boldsymbol{x}_{h,k} + \Delta\boldsymbol{x}_{h,k} \\ \boldsymbol{F}_{C,k}^T &= {}_0\boldsymbol{F}_{C,k} + \Delta\boldsymbol{F}_{C,k} \\ \boldsymbol{W}_{h,k}^T &= {}_0\boldsymbol{W}_{h,k} + \Delta\boldsymbol{W}_{h,k} \end{aligned} \quad (6.48)$$

States, contact forces and wrenches for the entire trajectory are combined as follows.

$$
\begin{aligned}
\boldsymbol{x}_{h,K}^T &= \begin{bmatrix} \boldsymbol{x}_{h,1}^T & \cdots & \boldsymbol{x}_{h,N_n}^T \end{bmatrix}^T \\
\boldsymbol{F}_{C,K}^T &= \begin{bmatrix} \boldsymbol{F}_{C,1}^T & \cdots & \boldsymbol{F}_{C,N_n}^T \end{bmatrix}^T \\
\boldsymbol{W}_{h,K}^T &= \begin{bmatrix} \boldsymbol{W}_{h,1}^T & \cdots & \boldsymbol{W}_{h,N_n}^T \end{bmatrix}^T
\end{aligned} \tag{6.49}
$$

Relationship between $\boldsymbol{x}_{h,K}$, $\boldsymbol{F}_{C,K}$ and $\boldsymbol{W}_{h,K}$ can be written for the entire trajectory using combined matrices as provided in Eq. (6.50).

$$
\begin{aligned}
\tilde{\boldsymbol{M}}_{h,K} &= \begin{bmatrix} \tilde{\boldsymbol{M}}_{h_1,1} & \tilde{\boldsymbol{M}}_{h_2,2} & \boldsymbol{0} & \cdots \\ \boldsymbol{0} & \tilde{\boldsymbol{M}}_{h_1,2} & \tilde{\boldsymbol{M}}_{h_2,3} & \\ \vdots & & \ddots & \ddots \end{bmatrix} \\
\tilde{\boldsymbol{J}}_{C,K} &= \begin{bmatrix} -\Delta t \hat{\boldsymbol{J}}_{C,1} & \boldsymbol{0} & \cdots \\ \boldsymbol{0} & -\Delta t \hat{\boldsymbol{J}}_{C,2} & \\ \vdots & & \ddots \end{bmatrix} \\
\tilde{\boldsymbol{G}}_{h,K} &= \begin{bmatrix} \tilde{\boldsymbol{G}}_{h,1} \\ \tilde{\boldsymbol{G}}_{h,2} \\ \vdots \end{bmatrix} \\
\tilde{\boldsymbol{P}}_{h,K} &= \begin{bmatrix} \tilde{\boldsymbol{P}}_{h,1} & \boldsymbol{0} & \cdots \\ \boldsymbol{0} & \tilde{\boldsymbol{P}}_{h,2} & \\ \vdots & & \ddots \end{bmatrix}
\end{aligned} \tag{6.50}
$$

Dynamics for the floating base, contacting and swinging bodies are written as provided in Eq. (6.51).

$$
\begin{aligned}
\begin{bmatrix} \tilde{\boldsymbol{M}}_{B,K} & \tilde{\boldsymbol{P}}_{S,K} & \tilde{\boldsymbol{P}}_{C,K} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{B,K} \\ \overline{\boldsymbol{W}}_{S,K} \\ \overline{\boldsymbol{W}}_{C,K} \end{bmatrix} &= \boldsymbol{G}_{B,K} \\
\begin{bmatrix} \tilde{\boldsymbol{M}}_{S,K} & \tilde{\boldsymbol{P}}_{S,K} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{S,K} \\ \boldsymbol{W}_{S,K} \end{bmatrix} &= \boldsymbol{G}_{S,K} \\
\begin{bmatrix} \tilde{\boldsymbol{M}}_{C,K} & \tilde{\boldsymbol{P}}_{C,K} & \tilde{\boldsymbol{J}}_{C,K} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{C,K} \\ \boldsymbol{W}_{C,K} \\ \boldsymbol{F}_{C,K} \end{bmatrix} &= \boldsymbol{G}_{C,K}
\end{aligned} \tag{6.51}
$$

Optimization for the three agents are represented in a single objective Eq. (6.52) and a set of constraints in this work, however, problem is readily available for distributed optimization.

$$
\begin{aligned}
\underset{\Delta \boldsymbol{x}_{h,k}, \Delta \boldsymbol{F}_{C,k}, \Delta \boldsymbol{W}_{h,k}}{\text{minimize}} \quad & \sum_{k=1}^{M} \mathcal{L}_c(\boldsymbol{x}_{h,k}, \boldsymbol{F}_{C,k}, \boldsymbol{W}_{h,k}) \\
\text{s. t.} \quad & \mathcal{C}_{dyn}(\Delta \boldsymbol{x}_{h,k}, \Delta \boldsymbol{F}_{C,k}, \Delta \boldsymbol{W}_{h,k}) &&= \boldsymbol{0} \\
& \mathcal{C}_{coop}(\Delta \boldsymbol{x}_{h,k}, \Delta \boldsymbol{F}_{C,k}, \Delta \boldsymbol{W}_{h,k}) &&= \boldsymbol{0} \\
& \mathcal{C}_{cntct}(\Delta \boldsymbol{x}_{h,k}) &&= \boldsymbol{0} \\
& \mathcal{C}_{fc}(\Delta \boldsymbol{F}_{C,k}) &&\leq \boldsymbol{0}
\end{aligned} \tag{6.52}
$$

Dynamics Based on the dynamics given in Eq. (6.50) and representation of the trajectories as provided in Eq. (6.48), matrices for equality constraints are denoted as $\boldsymbol{A}_{h,dyn}$ and $\boldsymbol{B}_{h,dyn}$. The equality constraint is provided for only floating base for brevity.

$$\begin{bmatrix} \tilde{\boldsymbol{M}}_{B,K} & \tilde{\boldsymbol{P}}_{S,K} & \tilde{\boldsymbol{P}}_{C,K} \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{x}_{B,K} \\ \Delta\overline{\boldsymbol{W}}_{S,K} \\ \Delta\overline{\boldsymbol{W}}_{C,K} \end{bmatrix} = \boldsymbol{G}_{B,K} - \begin{bmatrix} \tilde{\boldsymbol{M}}_{B,K} & \tilde{\boldsymbol{P}}_{S,K} & \tilde{\boldsymbol{P}}_{C,K} \end{bmatrix} \begin{bmatrix} {}_0\boldsymbol{x}_{B,K} \\ {}_0\overline{\boldsymbol{W}}_{S,K} \\ {}_0\overline{\boldsymbol{W}}_{C,K} \end{bmatrix} \tag{6.53}$$

Contact Constraint Contact constraint is provided previously in Eq. (6.29). This condition is modified for the definition of the trajectory that is provided in Eq. (6.48). Simpliy contact point velocity has to be equal to the body velocity in ooposite direction in no slip condition and Eq. (6.54) projects the relationship on the decision variables for the quaratic optimization.

$$\begin{aligned} \boldsymbol{0} &= \begin{bmatrix} -\boldsymbol{I} & \boldsymbol{J}_{a,i} \end{bmatrix} \boldsymbol{x}_{B,K} \\ \boldsymbol{0} &= \begin{bmatrix} -\boldsymbol{I} & \boldsymbol{J}_{a,i} \end{bmatrix} \Delta\boldsymbol{x}_{B,K} + \begin{bmatrix} -\boldsymbol{I} & \boldsymbol{J}_{a,i} \end{bmatrix} {}_0\boldsymbol{x}_{B,K} \\ -\begin{bmatrix} -\boldsymbol{I} & \boldsymbol{J}_{a,i} \end{bmatrix} \Delta\boldsymbol{x}_{B,K} &= \begin{bmatrix} -\boldsymbol{I} & \boldsymbol{J}_{a,i} \end{bmatrix} {}_0\boldsymbol{x}_{B,K} \end{aligned} \tag{6.54}$$

Force Cone Constraint Cooperative force optimization phase is designed as a quadratic problem therefore constraints have to be set accordingly. Contact constraints are dedicated to keep tangential forces small so that no sliping occurs. To do that a friction pyramid is created inside a friction cone. The friction cone is geometric interpretation of the magnitude of the allowable tangential force that can be applied on the ground. The allowable limit is calculated simply multiplying the normal component of the applied force by the contact point with the friction coefficient. Normal component of the force is denoted as ${}_Bf_{c,n}$ and the tangential components of the applied force are denoted as ${}_Bf_{c,t}$, ${}_Bf_{c,s}$, respectively. It should be noted that applied force is defined with respect to $\mathcal{F}_B$. The friction coefficient is denoted as μ. Radius of the friction cone is define as $\boldsymbol{r}_{fc} = \mu_B f_{c,n}$. The friction pyramid

is defined such that it is always upper bounded by the $\boldsymbol{r}_{fc}$ and this is achieved by setting linear bounds that are denoted as $r_{fc,s}$ and $r_{fc,t}$. These bounds are calculated such that $|r_{fc,s}| \approx 0.707\boldsymbol{r}_{fc}$ and $|r_{fc,s}| \approx 0.707\boldsymbol{r}_{fc}$. It should be noted that bounding $\boldsymbol{r}_{fc}$ creates a non-linear relationship and make optimizationa no-linear problem, however, bounds that are defined for tangetial components can be implemented in linear fashion. The geometric interpretation of the friction cone (red solid line) and pyramid (blue solid line) is provided in Figure 6.9.

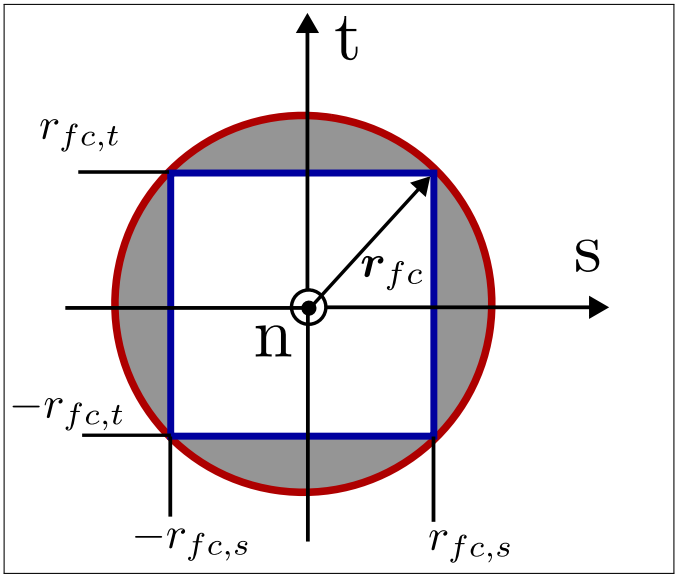


Figure 6.9. Geometric interpretation of the friction pyramid.

Based on the linear and conservative bounds following linear constraints are defined for the tangential forces.

$$\begin{aligned} -0.707\mu_B f_{c,n} &\leq {}_B f_{c,t} \leq 0.707\mu_B f_{c,n} \\ -0.707\mu_B f_{c,n} &\leq {}_B f_{c,s} \leq 0.707\mu_B f_{c,n} \\ -2 \times 0.707\mu_B f_{c,n} &\leq {}_B f_{c,t} + {}_B f_{c,s} \leq 2 \times 0.707\mu_B f_{c,n} \end{aligned} \tag{6.55}$$

The relationship given in Eq. (6.55) is written compactly as provided in Eq.. (**??**), where ${}_B\boldsymbol{F}_{C,k}$ is the vector containing the decision variables. ${}_B\boldsymbol{n}$, ${}_B\boldsymbol{t}$ and ${}_B\boldsymbol{s}$ are the unit vectors attached on the contact points and defined in $\mathcal{F}_B$. As a practical note, calculating these unit vectors with respect to $\mathcal{F}_B$ is straight forward calculation when there are passive joints at the ankle of the contact legs. Depending on the kinamatic structure of the leg certain unit vectors can be assumed to be in the same direction with the axes of the body frame.

$$\begin{aligned}\left(-2\times 0.707\mu_B\boldsymbol{n}^T - {}_B\boldsymbol{t}^T - {}_B\boldsymbol{s}^T\right){}_B\boldsymbol{F}_{C,k} &\leq \mathbf{0}\\ \left(\ 2\times 0.707\mu_B\boldsymbol{n}^T + {}_B\boldsymbol{t}^T + {}_B\boldsymbol{s}^T\right){}_B\boldsymbol{F}_{C,k} &\leq \mathbf{0}\end{aligned} \tag{6.56}$$

Within current work ${}_B\boldsymbol{n}$, ${}_B\boldsymbol{t}$ and ${}_B\boldsymbol{s}$ are assumed to be constant and defined as ${}_B\boldsymbol{n}^T = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T$, ${}_B\boldsymbol{t} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$ and ${}_B\boldsymbol{s} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$.

Cost Function Force phase optimization tries to keep the commanded motion intact while minimizing the disturbance injected on the system due to the joint accelerations. The joint accelerations are affecting each agent due to the $\mathcal{C}_{coop}$, which is provided in Eq. (6.26). The constraints that are introduced in this chapter previously maintain contact, force cone constraints satisfied. Cost function for this optimization is defined as in Eq. (6.57).

$$\begin{aligned}&\mathcal{L}_c(\boldsymbol{x}_{h,k}, \boldsymbol{F}_{C,k}, \boldsymbol{W}_{h,k}) =\\ &\left(-\boldsymbol{u}_{B,xz} + \left[\begin{array}{ccc|c|c} 1 & 0 & 0 & \mathbf{0} & \mathbf{0} \\ 0 & 0 & 1 & \mathbf{0} & \mathbf{0} \end{array}\right]\boldsymbol{x}_{B,k}\right)^T \boldsymbol{Q}_T \left(-\boldsymbol{u}_{B,xz} + \left[\begin{array}{ccc|c|c} 1 & 0 & 0 & \mathbf{0} & \mathbf{0} \\ 0 & 0 & 1 & \mathbf{0} & \mathbf{0} \end{array}\right]\boldsymbol{x}_{B,k}\right) +\\ &\boldsymbol{x}_{B,k}^T \boldsymbol{I}I_Q^T \boldsymbol{Q}_W \boldsymbol{I}I_Q \boldsymbol{x}_{B,k}\end{aligned} \tag{6.57}$$

where $\boldsymbol{I}I_Q$ is a matrix that selects the states that are joint velocities of contact and swing legs and give an approximation of acceleration of these selected states. Joint velocities

within states $\boldsymbol{x}_{h,k}$ are denoted as $\boldsymbol{q}_{h,k}$ for $h = C, S$ and the selection of these states is defined as in Eq. (6.58).

$$\begin{aligned} \dot{\boldsymbol{q}}_{h,k} &= \left[\begin{array}{cc|ccc} \mathbf{0} & \mathbf{0} & 1 & 0 & 0 \\ \mathbf{0} & \mathbf{0} & 0 & 1 & 0 \\ \mathbf{0} & \mathbf{0} & 0 & 0 & 1 \end{array}\right] \boldsymbol{x}_{h,k} \\ \dot{\boldsymbol{q}}_{h,k} &= \boldsymbol{I} I_q \boldsymbol{x}_{h,k} \end{aligned} \tag{6.58}$$

The approximation for acceleration is given as below, where states are assumed to propogate with first order Euler method.

$$\ddot{\boldsymbol{q}}_{h,k} = \boldsymbol{I} I_q \frac{(\boldsymbol{x}_{h,k+1} - \boldsymbol{x}_{h,k})}{\Delta t} \tag{6.59}$$

Finally, accelerations $\ddot{\boldsymbol{q}}_{h,k}$ for $h = C, S$ for $k = 1, \cdots, N_n$ is given as provided in Eq. (6.60)

$$\begin{aligned} \ddot{\boldsymbol{q}}_{h,k} &= \begin{bmatrix} -\Delta t \boldsymbol{I} I_q & \Delta t \boldsymbol{I} I_q & & \\ & -\Delta t \boldsymbol{I} I_q & \Delta t \boldsymbol{I} I_q & \\ & & & \ddots \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{h,1} \\ \boldsymbol{x}_{h,2} \\ \vdots \\ \boldsymbol{x}_{h,N_n} \end{bmatrix} \\ \ddot{\boldsymbol{q}}_{h,k} &= \boldsymbol{I} I_Q \boldsymbol{x}_{h,K} \end{aligned} \tag{6.60}$$

6.4 Results

This section presents the simulation of the algorithm for the proposed method. The simulation is not executed in a physics environment and trajectories illustrated in the section are extimated trajectories only. Solutions are obtained on *Intel(R) Core i7-4720HQ CPU @2.60 GHz 16GB RAM PC* with Matlab 2019b software.

The sampling time Δt for the discrete model is selected to be $\Delta t = 0.05$s. Nominal values for N_n and N_p are selected as $N_n = 20$ and $N_p = 4$, respectively. $\boldsymbol{Q}_s$ and $\boldsymbol{Q}_p$ for contact phase optimization is selected to be $\boldsymbol{Q}_s = 5$ and $\boldsymbol{Q}_p = 2$. $\boldsymbol{Q}_0$, $\boldsymbol{Q}_f$ and $\boldsymbol{Q}_t$ for swing phase optimization is selected to be $\boldsymbol{Q}_0 = 12.5$, $\boldsymbol{Q}_f = 12.5$ and $\boldsymbol{Q}_t = 30$. Finally, $\boldsymbol{Q}_u$ and $\boldsymbol{Q}_w$ for cooperativev force optimization is selected to be $\boldsymbol{Q}_u = 100$ and $\boldsymbol{Q}_w = 5$.

Based on these settings and from initial conditions of $\boldsymbol{u}_B(t=0) = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$, $\boldsymbol{r}_2(t=0) = \begin{bmatrix} 0.01 & -0.344 & 0.063 \end{bmatrix}^T$ and $\boldsymbol{r}_3(t=0) = \begin{bmatrix} 0.045 & -0.344 & -0.239 \end{bmatrix}^T$, ASLB is asked to move forward by $\boldsymbol{u}_{B,ref} = \begin{bmatrix} 0.1 & 0 & 0 \end{bmatrix}^T$. The results are provided in the following figures.

Figure 6.10 illustrates calculated swing and contact trajectories. N_n for the first optimization is minimized to $N_n = 6$ from nominal 20 as contact initial position for ${}^{p=1}\boldsymbol{r}_{c,1}$ is already provided by the sensor information. For a feasible finite horizon $N_n = 6$, contact leg, which is the right leg or **Agent 2**, calculates an approximate trajectory for $\boldsymbol{r}_c$ starting from $\boldsymbol{r}_2(t=0)$ and diverge from the CoM towards (+)x and (+)z directions. Note that this trajectory is calculated with resect to $\mathcal{F}_B$. Swing phase optimization connects $\boldsymbol{r}_3(t=0)$ to the ${}^{p=2}\boldsymbol{r}_{c,1}$ without violating the kinematic bounds.

Figure 6.11 illustrates nominal force trajectory for contact leg as it interacts with the ground. ${}_0\boldsymbol{F}_C$ is the initial trajectory that is calculated by substituting $\boldsymbol{q}_{a,i}$ and $\boldsymbol{x}_i$ into the agent dynamics. ${}_{opt}\boldsymbol{F}_C$ is the resultant contact force trajectory. It is visible from the figure that, there is not a significant change in y-axis. However, ${}_{opt}\boldsymbol{F}_C$ is shifted in (+)x direction approximately 0.4 N.

Figure 6.12 shows the contact and swing trajectories for the second step, where N_n is minimized to $N_n = 16$. Similar to the first step, ${}^{p=1}\boldsymbol{r}_{c,1}$ for contact phase optimization is

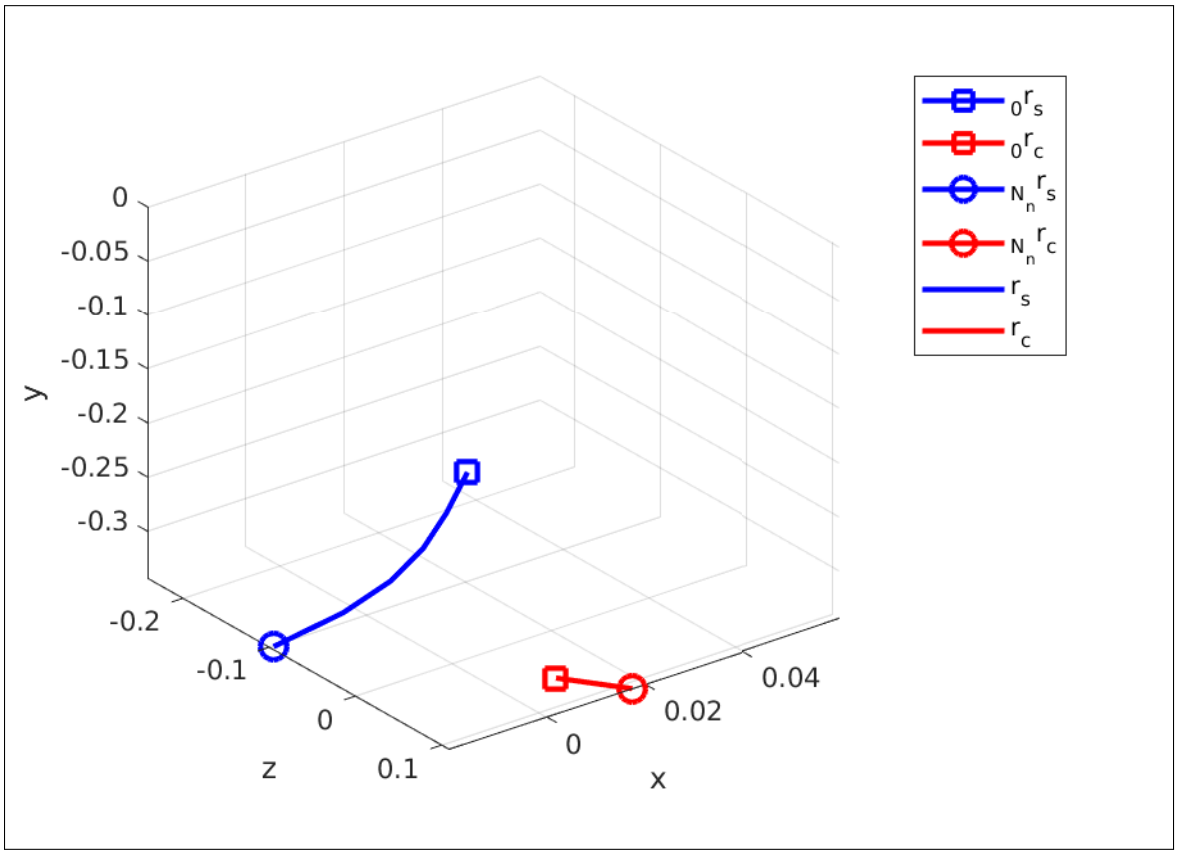


Figure 6.10. Calculated swing and contact leg trajectories for first step.

the initial position of the current contact leg, which is **Agent 3**. $\boldsymbol{r}_c$ in Figure 6.12 converges to the CoM in the beginning of this phase then pushes away towards (-)x and (-)z directions.

A smaller correction occurs in contact force as ${}_0\boldsymbol{F}_C$ and ${}_{opt}\boldsymbol{F}_C$ have slight differences in all three directions.

Figure 6.14 provides $\boldsymbol{u}_B$ trajectory throughout the walking simulation along with the $\boldsymbol{r}_{c,i}$ for $i = 2, 3$. This figure illustrates the characteristic difference of the method, which is calculating contact positions and forces to track a reference velocity $\boldsymbol{u}_{B,ref}$. Black circle in Figure 6.14 indicates the CoM and decision variables are defined with respect to that. Trajectory of $\boldsymbol{u}_B$, which is given with blue line, settles in a cyclic pattern. The mean of the magnitude in x direction is approximately 0.28 m/s, while it is almost zero for z direction.

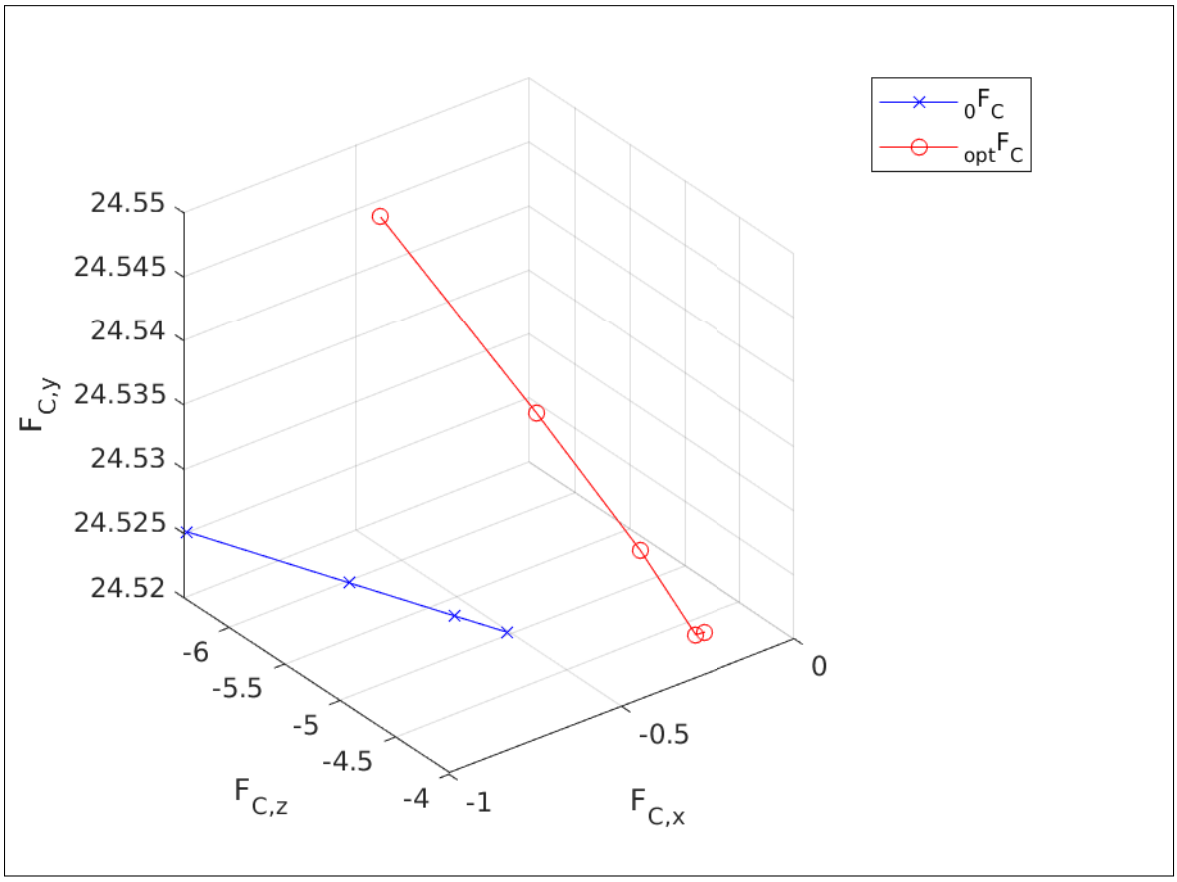


Figure 6.11. Calculated force trajectory for first step.

Figure 6.15 provides a more intuitive illustration of the motion of the robot as ${}_I\boldsymbol{r}_B$ is calculated from $\boldsymbol{u}_B$. ${}_I\boldsymbol{r}_B$ gives the position of the origin of the $\mathcal{F}_B$ with respect to the $\mathcal{F}_I$. Trajectory of ${}_I\boldsymbol{r}_B$ reveals that body drifts away from the line $z = 0$, while achieving forward motion as desired. The drift is due to the first contact leg, which is the **Agent 2**, and cannot be corrected.

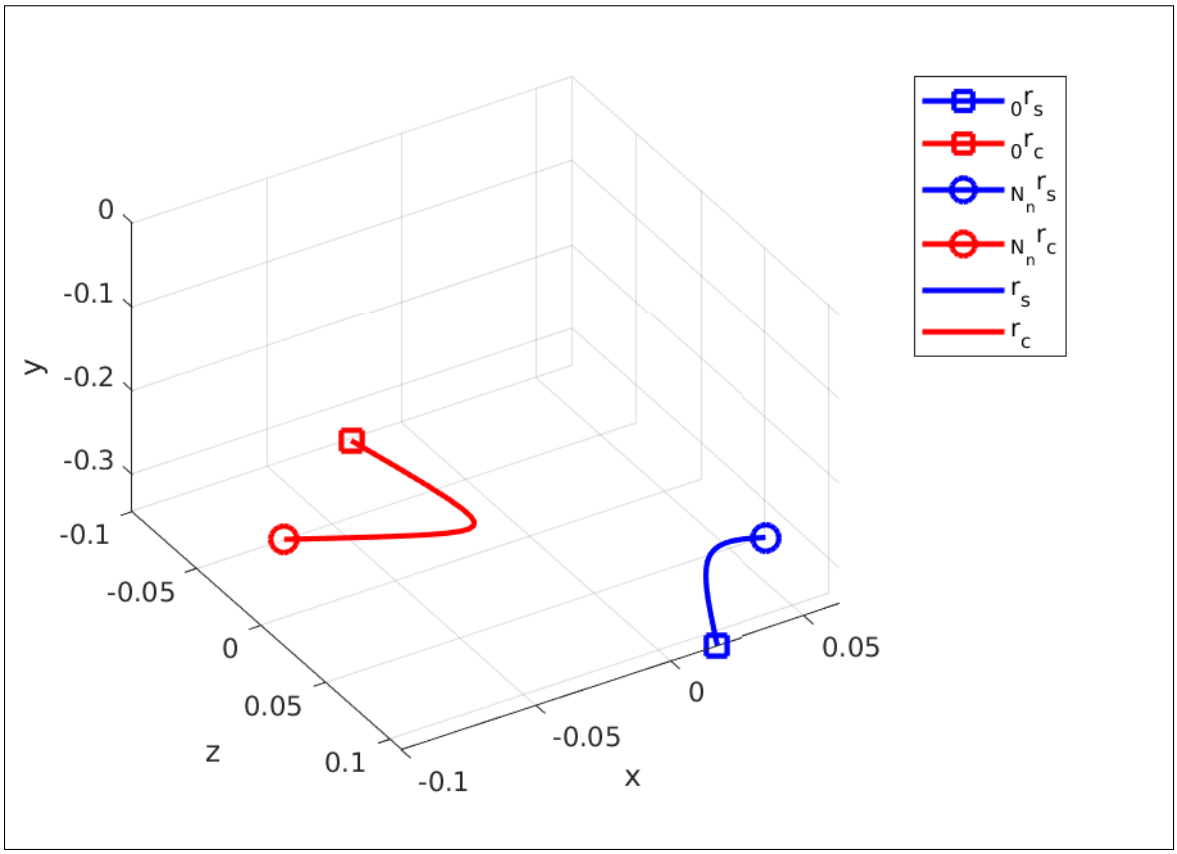


Figure 6.12. Calculated swing and contact leg trajectories for second step.

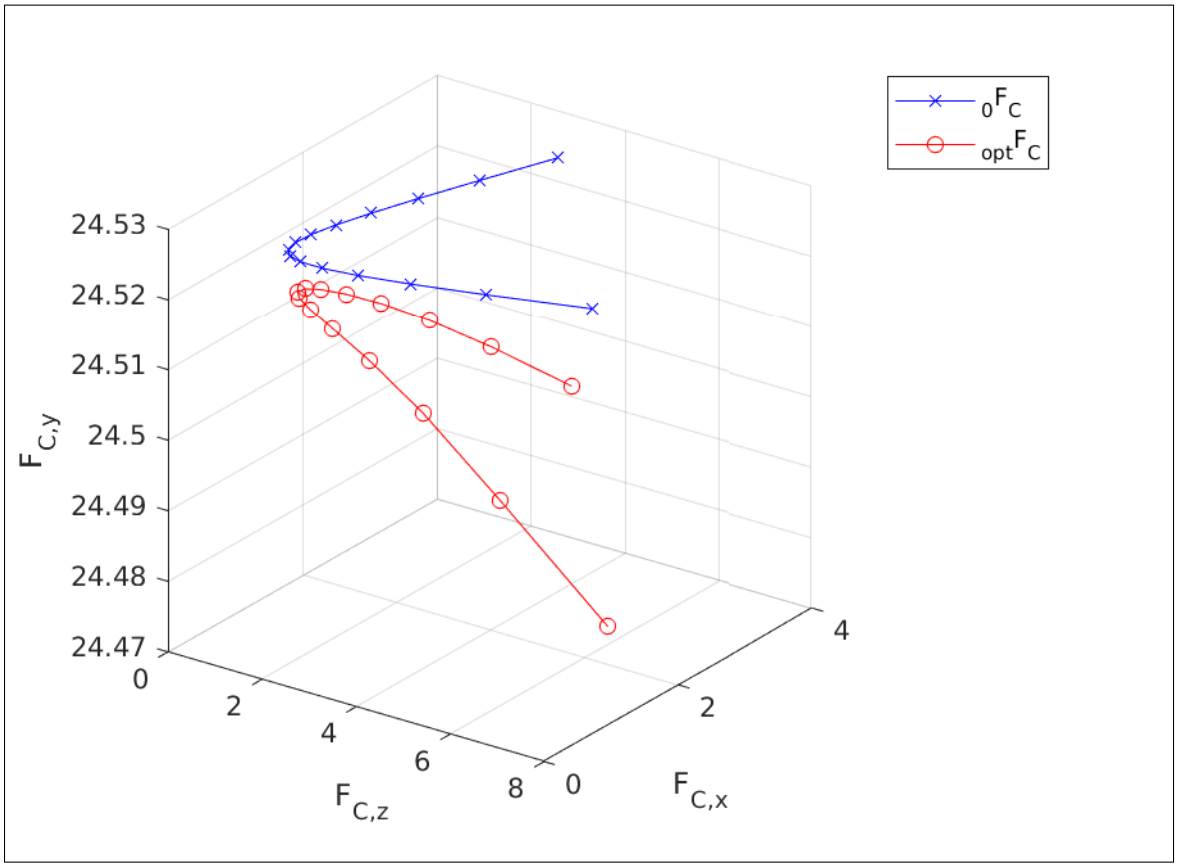


Figure 6.13. Calculated force trajectory for second step.

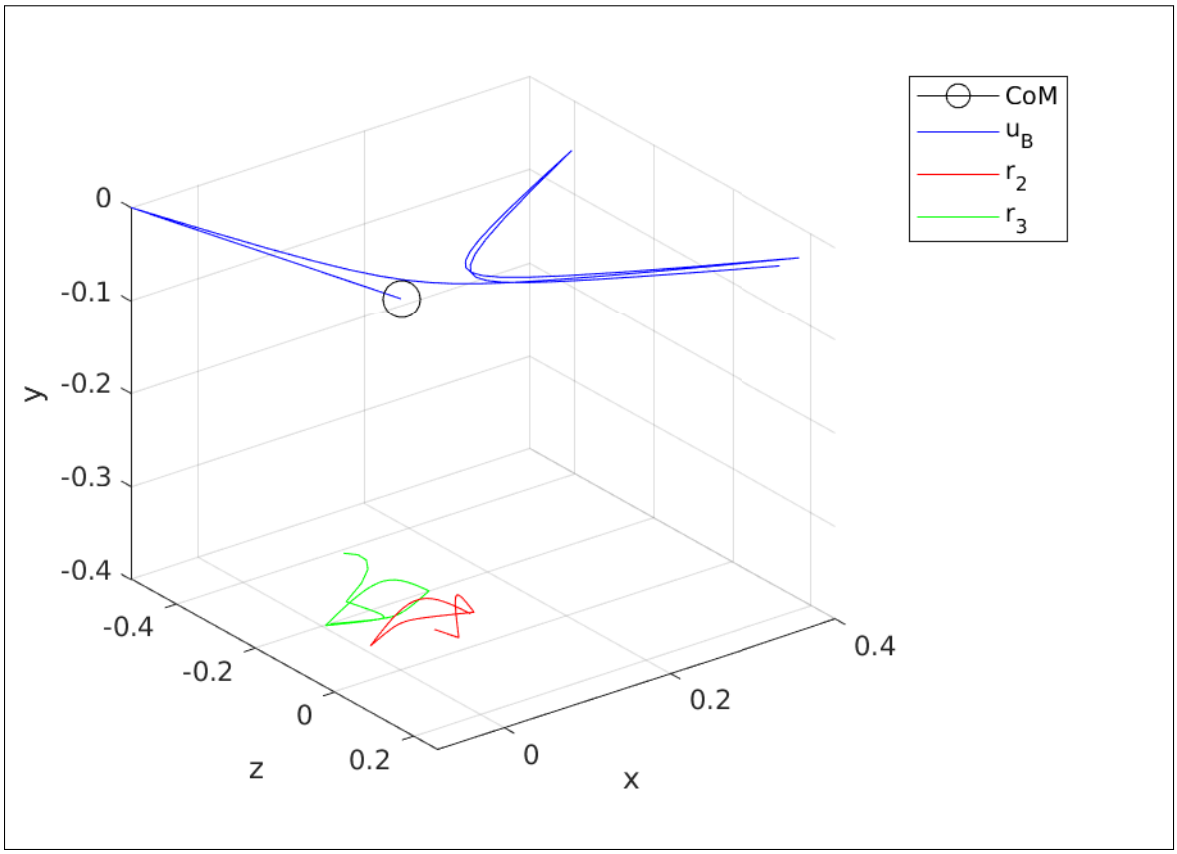


Figure 6.14. $\boldsymbol{u}_B$ trajectory plot along with leg motion with respect to CoM.

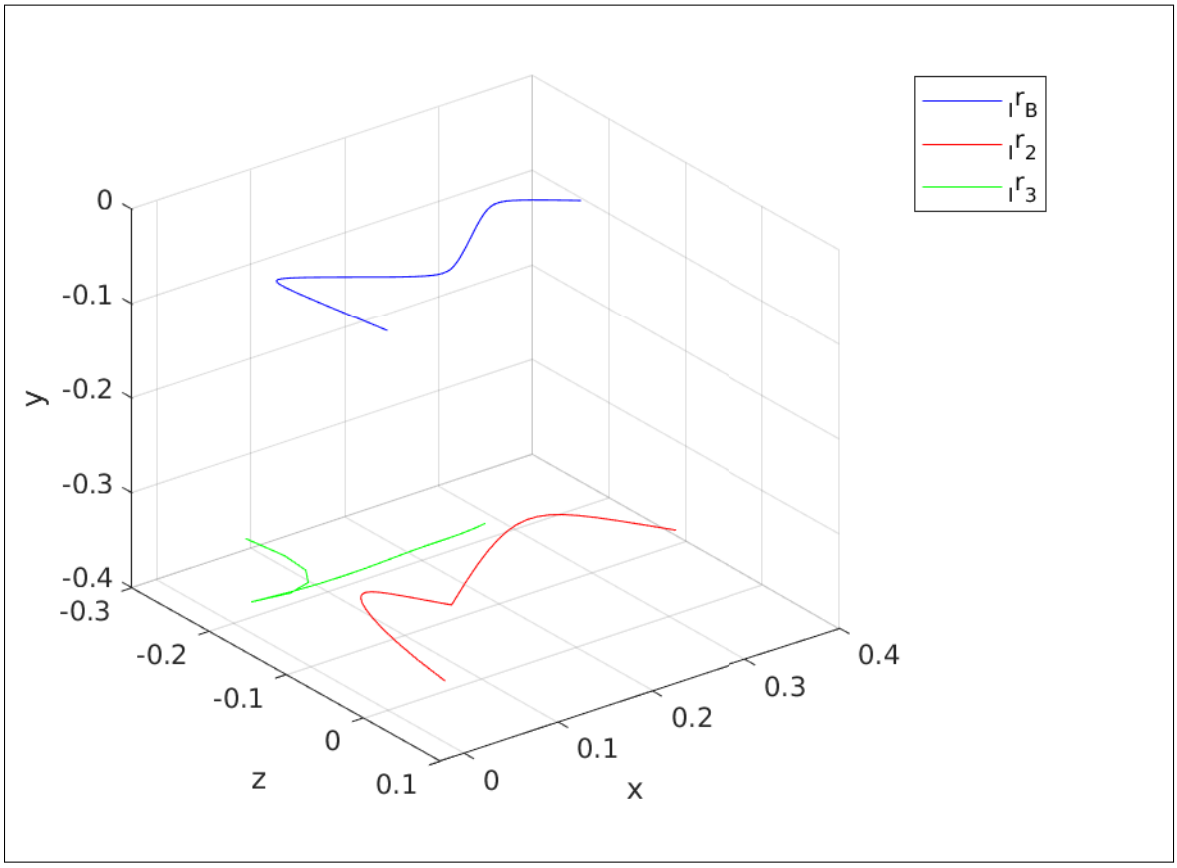


Figure 6.15. $\boldsymbol{r}_B$ plot along with leg motion with respect to $\mathcal{F}_I$.

6.5 Conclusions

This work proposes a cooperative online trajectory generation algorithm. The proposed algorithm utilizes distributed modeling of ASLB robot to generate contact and swing trajectories. This is done by a series of optimization that generates approximate trajectoiries and passes them to a final optimization that refines the motion with cooperative system framework. With the proposed method given velocity command was able to be tracked with some offset. During simulation of the algorithm, it is observed that relaxation of the state bounds decreases the offsets between velocity command and mean of the actual velocity. In addition to that, since this method is designed for velocity level dynamics, a drift in walking line occurs. However, extracting dynamics from kinematics and developing a controller for this simpler set equations allows faster calculation of the future steps.

6.6 Future Work

This method is not simulated on a physics environment or on the real robot. Therefore, in this we can only illustrate the concept of making optimization lightweight such that we do not include kinematics into an optimization that needs to be calculated several times a second or define dynamics with respect to the inertial frame, which creates complexity due to additional Euler angles. Therefore, proposed method has to be implemeted on a physics simulation.

Chapter 7

Summary and Closing Remarks

Contributions listed in § 1.3 led to computationally efficient robust synthesis methods for cooperative system that suffers uncertainty in the Dissertation. Besides that, this Dissertation studies possible applications for cooperative synthesis methods for practical state of the art applications such as docking on a cooperating target and online trajectory generation for ASLB, a biped robot.

Outcomes of this dissertation was that recasting of the dynamics of the single and mutli-agent systems, in a form such that uncertainties and controllers are connected to the system in the same manner makes method applicable as a nested controller. The nested implementation of single and coopeartive controller allows distribution of the uncertainties, which leads to faster computation for a lumped model. The performance of the method shows dramatic increase with the distributed adaptation of the same nested method.

Second outcome of this dissertation is that cooperative systems' robust performance can be increased by finding locally optimal edge weights. By this way complexity in synthesis of a cooperative controller can be eliminated. This dissertation shows that there is an LPV edge weight mapping that increases H_∞ performance of an LPV cooperative system. The method used to obtain this result includes implementation of IQC analysis with valid time domain IQCs into LPV robust H_∞ synthesis.

Third outcome is that a distributed method for H_∞ edge weight systhesis is introduced based on neurality of the interconnections and dissipativity. Neutrality condition in interconnections are implemented by introducing a synthetic agent into the system, which secures the communication

topology and edge weight information within. With the proposed method calculation time for edge weight are reduced compared to the lumped counterpart.

Fourth outcome of this dissertation illustrates practical implementations of cooperative control framework on a cooperative system that executes docking. A unified framework for cooperative docking is proposed as an MPC, which prioritize certain set of states. The prioritization allows designer of the controller to dedicate each docking party to handle certain sub-task of the docking allowing a greater DoF to handle docking.

Finally, cooperative control framework is used in calculation of a walking trajectory for a biped robot. The proposed method models ASLB as cooperative agents and executes a series of optimizations to calculate contact, swing and force trajectories. The force optimization phase uses cooperative control framework to refine contact and swing trajectories. By this way, a nonlinear problem is converted into smaller quadratic problems that is efficiently solved in a pace that is suitable for online implementations.

References

[1] S. Wang, H. Pfifer, and P. Seiler, "Robust synthesis for linear parameter varying systems using integral quadratic constraints," Automatica, vol. 68, pp. 111–118, 2016.

[2] A. Ferrante and L. Ntogramatzidis, "The generalised discrete algebraic riccati equation in linear-quadratic optimal control," Automatica, vol. 49, no. 2, pp. 471–478, feb 2013. [Online]. Available: https://doi.org/10.1016%2Fj.automatica.2012.11.006

[3] W. Yu, Y. Yao, and W. Chen, "Analytical cooperative entry guidance for rendezvous and formation flight," Acta Astronautica, vol. 171, pp. 118–138, 2020, doi: https://doi.org/10.1016/j.actaastro.2020.02.044.

[4] L. Bai, L. Zhu, X. Zhang, W. Zhang, and Q. Yu, "Multi-satellite relay transmission in 5g: Concepts, techniques, and challenges," IEEE Network, vol. 32, no. 5, pp. 38–44, 2018, doi: http://doi.org/10.1109/MNET.2018.1800038.

[5] H. X. Pham, H. M. La, D. Feil-Seifer, and M. C. Deans, "A distributed control framework of multiple unmanned aerial vehicles for dynamic wildfire tracking," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 50, no. 4, pp. 1537–1548, 2020.

[6] S. Zhang, H. Zhang, B. Di, and L. Song, "Cellular cooperative unmanned aerial vehicle networks with sense-and-send protocol," IEEE Internet of Things Journal, vol. 6, no. 2, pp. 1754–1767, 2019.

[7] H. Zhang and P. Gurfil, "Distributed control for satellite cluster flight under different communication topologies," Journal of Guidance, Control, and Dynamics, vol. 39, no. 3, pp. 617–627, 2016.

[8] F. Yao, J. Li, Y. Chen, X. Chu, and B. Zhao, "Task allocation strategies for cooperative task planning of multi-autonomous satellite constellation," Advances in Space Research, vol. 63, no. 2, pp. 1073–1084, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0273117718307713

[9] R. Casado-Vara, F. Prieto-Castrillo, and J. M. Corchado, "A game theory approach for cooperative control to improve data quality and false data detection in wsn," International Journal of Robust and Nonlinear Control, vol. 28, no. 16, pp. 5087–5102, 2018.

[10] F. de Ponte Müller, E. M. Diaz, and I. Rashdan, "Cooperative positioning and radar sensor fusion for relative localization of vehicles," in 2016 IEEE Intelligent Vehicles Symposium (IV), 2016, doi: http://doi.org/10.1109/IVS.2016.7535520, pp. 1060–1065.

[11] M. W. Khan, J. Wang, L. Xiong, and M. Ma, "Modelling and optimal management of distributed microgrid using multi-agent systems," Sustainable Cities and Society, vol. 41, pp. 154 – 169, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2210670718302312

[12] Y. Zhang, Y. Zheng, and S. Li, "Enhancing cooperative distributed model predictive control for the water distribution networks pressure optimization," Journal of Process Control, vol. 84, pp. 70–88, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0959152418303780

[13] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," IEEE transactions on automatic control, vol. 49, no. 9, pp. 1465–1476, 2004.

[14] V. Shaferman and T. Shima, "Cooperative optimal guidance laws for imposing a relative intercept angle," AIAA Journal of Guidance, Control, and Dynamics, vol. 38, no. 8, pp. 1395–1408, 2015.

[15] W. Yu, G. Chen, and M. Cao, "Distributed leader–follower flocking control for multi-agent dynamical systems with time-varying velocities," Systems & Control Letters, vol. 59, no. 9, pp. 543–552, 2010.

[16] S. B. Sarsilmaz, T. Yucelen, and T. Oswald, "A distributed adaptive control approach for heterogeneous uncertain multiagent systems," in 2018 AIAA Guidance, Navigation, and Control Conference, January 2018.

[17] D. Ding, Q. Han, Z. Wang, and X. Ge, "A survey on model-based distributed control and filtering for industrial cyber-physical systems," IEEE Transactions on Industrial Informatics, vol. 15, no. 5, pp. 2483–2499, 2019, doi: http://doi.org/10.1109/TII.2019.2905295.

[18] C. Poignard, T. Pereira, and J. P. Pade, "Spectra of laplacian matrices of weighted graphs: Structural genericity properties," SIAM Journal on Applied Mathematics, vol. 78, no. 1, pp. 372–394, 2018. [Online]. Available: https://doi.org/10.1137/17M1124474

[19] A. Arenas, A. Díaz-Guilera, J. Kurths, Y. Moreno, and C. Zhou, "Synchronization in complex networks," Physics Reports, vol. 469, no. 3, pp. 93–153, 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0370157308003384

[20] H.-J. Li, Z. Bu, Z. Wang, J. Cao, and Y. Shi, "Enhance the performance of network computation by a tunable weighting strategy," IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 2, no. 3, pp. 214–223, 2018.

[21] A. Ghosh, S. Boyd, and A. Saberi, "Minimizing effective resistance of a graph," SIAM review, vol. 50, no. 1, p. 37–66, 2008 doi: .

[22] Y. Zou, Z. Zhou, X. Dong, and Z. Meng, "Distributed formation control for multiple vertical takeoff and landing uavs with switching topologies," IEEE/ASME Transactions on Mechatronics, vol. 23, no. 4, pp. 1750–1761, 2018.

[23] L. El Ghaoui and S.-l. Niculescu, Advances in linear matrix inequality methods in control. SIAM, 2000.

[24] A. Ataei and Q. Wang, "An ellipsoid algorithm for linear optimization with uncertain lmi constraints," in 2012 American Control Conference (ACC), 2012, pp. 857–862.

[25] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, Linear matrix inequalities in system and control theory. SIAM, 1994.

[26] V. A. Yakubovich, "A frequency theorem for the case in which the state and control spaces are hilbert spaces, with an application to some problems of synthesis of optimal controls, ii," Siberian Mathematical Journal, vol. 16, no. 5, pp. 828–845, 1975.

[27] J. G. VanAntwerp and R. D. Braatz, "A tutorial on linear and bilinear matrix inequalities," Journal of process control, vol. 10, no. 4, pp. 363–385, 2000.

[28] G.-R. Duan and H.-H. Yu, LMIs in control systems: analysis, design and applications. CRC press, 2013.

[29] F. Wang and V. Balakrishnan, "Improved stability analysis and gain-scheduled controller synthesis for parameter-dependent systems," IEEE Transactions on Automatic Control, vol. 47, no. 5, pp. 720–734, 2002.

[30] Y. Yang and S. Dubljevic, "Linear matrix inequalities (lmis) observer and controller design synthesis for parabolic pde," European Journal of Control, vol. 20, no. 5, pp. 227–236, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0947358014000430

[31] I. Masubuchi, A. Ohara, and N. Suda, "Lmi-based controller synthesis: A unified formulation and solution," International Journal of Robust and Nonlinear Control, vol. 8, no. 8, pp. 669–686, 1998.

[32] P. Gahinet and P. Apkarian, "An lmi-based parametrization of all $h\infty$ controllers with applications," in Proceedings of 32nd IEEE Conference on Decision and Control, 1993, pp. 656–661 vol.1.

[33] A. Nobakhti and H. Wang, "Noniterative $h\infty$-based model order reduction of lti systems using lmis," IEEE Transactions on Control Systems Technology, vol. 17, no. 2, pp. 494–501, 2009.

[34] M. Fu and Z.-Q. Luo, "Computational complexity of a problem arising in fixed order output feedback design," Systems & Control Letters, vol. 30, no. 5, pp. 209–215, 1997. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167691197000145

[35] M. Chilali, P. Gahinet, and C. Scherer, "Multi-objective output-feedback control via lmi optimization," IFAC Proceedings Volumes, vol. 29, no. 1, pp. 1691–1696, 1996.

[36] C. Scherer, P. Gahinet, and M. Chilali, "Multiobjective output-feedback control via lmi optimization," IEEE Transactions on Automatic Control, vol. 42, no. 7, pp. 896–911, 1997.

[37] Q. Tran Dinh, S. Gumussoy, W. Michiels, and M. Diehl, "Combining convex–concave decompositions and linearization approaches for solving bmis, with application to static output feedback," IEEE Transactions on Automatic Control, vol. 57, no. 6, pp. 1377–1390, 2012.

[38] A. Bidram, F. L. Lewis, and A. Davoudi, "Distributed control systems for small-scale power networks: Using multiagent cooperative control theory," IEEE Control systems magazine, vol. 34, no. 6, pp. 56–77, 2014.

[39] A. R. Girard, J. B. de Sousa, J. A. Misener, and J. K. Hedrick, "A control architecture for integrated cooperative cruise control and collision warning systems," in Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No. 01CH37228), vol. 2. IEEE, 2001, pp. 1491–1496.

[40] A. Helmersson, "μ synthesis and lft gain scheduling with real uncertainties," International Journal of Robust and Nonlinear Control, vol. 8, no. 7, pp. 631–642, 1998. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291099-1239%28199806%298%3A7%3C631%3A%3AAID-RNC335%3E3.0.CO%3B2-D

[41] H. Bevrani, M. R. Feizi, and S. Ataee, "Robust frequency control in an islanded microgrid: H_∞ and μ -synthesis approaches," IEEE Transactions on Smart Grid, vol. 7, no. 2, pp. 706–717, 2016.

[42] A. Megretski and A. Rantzer, "System analysis via integral quadratic constraints," IEEE Transactions on Automatic Control, vol. 42, no. 6, pp. 819–830, June 1997.

[43] Z. Wang and T. Zhou, "Iqc based robust stability verification for a networked system with communication delays," Science China Information Sciences, vol. 61, no. 12, p. 122201, 2018.

[44] P. Seiler, "Stability analysis with dissipation inequalities and integral quadratic constraints," IEEE Transactions on Automatic Control, vol. 60, no. 6, pp. 1704–1709, June 2015.

[45] H. Pfifer and P. Seiler, "Robustness analysis of linear parameter varying systems using integral quadratic constraints," International Journal of Robust and Nonlinear Control, vol. 25, no. 15, pp. 2843–2864, 2015.

[46] C. Scherer and S. Weiland, "Linear matrix inequalities in control," Lecture Notes, Dutch Institute for Systems and Control, Delft, The Netherlands, vol. 3, no. 2, 2000.

[47] F. Wu, "Control of linear parameter varying systems," Ph.D. dissertation, Department of Mechanical Engineering, University of California, Berkeley, January 1995.

[48] C.-Y. Kao, A. Megretski, and U. Jönsson, "Specialized fast algorithms for iqc feasibility and optimization problems," Automatica, vol. 40, no. 2, pp. 239–252, 2004.

[49] A. Hansson and L. Vandenberghe, "A primal-dual potential reduction method for integral quadratic constraints," in Proceedings of the 2001 American Control Conference.(Cat. No. 01CH37148), vol. 4. IEEE, 2001, pp. 3013–3018.

[50] C. Langbort, R. S. Chandra, and R. D'Andrea, "Distributed control design for systems interconnected over an arbitrary graph," IEEE Transactions on Automatic Control, vol. 49, no. 9, pp. 1502–1519, 2004.

[51] P. Viccione, C. W. Scherer, and M. Innocenti, "Lpv synthesis with integral quadratic constraints for distributed control of interconnected systems," IFAC Proceedings Volumes, vol. 42, no. 6, pp. 13–18, 2009.

[52] T. Zhou and Y. Zhang, "On the stability and robust stability of networked dynamic systems," IEEE Transactions on Automatic Control, vol. 61, no. 6, pp. 1595–1600, 2015.

[53] M. S. Andersen, S. K. Pakazad, A. Hansson, and A. Rantzer, "Robust stability analysis of sparsely interconnected uncertain systems," IEEE Transactions on Automatic Control, vol. 59, no. 8, pp. 2151–2156, 2014.

[54] C. Mu, K. Wang, Z. Ni, and C. Sun, "Cooperative differential game-based optimal control and its application to power systems," IEEE Transactions on Industrial Informatics, vol. 16, no. 8, pp. 5169–5179, 2020.

[55] E. P. van Horssen and S. Weiland, "Synthesis of distributed robust h-infinity controllers for interconnected discrete time systems," IEEE Transactions on Control of Network Systems, vol. 3, no. 3, pp. 286–295, 2016.

[56] Y. Kim, "On the stability margin of networked dynamical systems," IEEE Transactions on Automatic Control, vol. 62, no. 10, pp. 5451–5456, 2017.

[57] Z. Li, Z. Duan, G. Chen, and L. Huang, "Consensus of multiagent systems and synchronization of complex networks: A unified viewpoint," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 57, no. 1, pp. 213–224, 2010.

[58] K. Zhou and J. C. Doyle, Essentials of robust control. Prentice hall Upper Saddle River, NJ, 1998, vol. 104.

[59] S. Z. Khong, E. Lovisari, and A. Rantzer, "A unifying framework for robust synchronization of heterogeneous networks via integral quadratic constraints," IEEE Transactions on Automatic Control, vol. 61, no. 5, pp. 1297–1309, 2016.

[60] T. Feng, H. Zhang, Y. Luo, and H. Liang, "Globally optimal distributed cooperative control for general linear multi-agent systems," Neurocomputing, vol. 203, pp. 12 – 21, 2016.

[61] A. Ghosh, S. Boyd, and A. Saberi, "Minimizing effective resistance of a graph," SIAM Rev., vol. 50, no. 1, p. 37–66, Feb. 2008.

[62] S. Y. Shafi, M. Arcak, and L. El Ghaoui, "Designing node and edge weights of a graph to meet laplacian eigenvalue constraints," pp. 1016–1023, 2010.

[63] R. Dai, J. Maximoff, and M. Mesbahi, "Optimal trajectory generation for establishing connectivity in proximity networks," IEEE Transactions on Aerospace and Electronic Systems, vol. 49, no. 3, pp. 1968–1981, 2013.

[64] S. C. and D. R., "Identification of network topology via quadratic optimization," pp. 5752–5757, 2015.

[65] S. Wang, H. Pfifer, and P. Seiler, "Robust synthesis for linear parameter varying systems using integral quadratic constraints," Automatica, vol. 68, pp. 111 – 118, 2016.

[66] D. Noll, "Local convergence of an augmented lagrangian method for matrix inequality constrained programming," Optimisation Methods and Software, vol. 22, no. 5, pp. 777–802, 2007.

[67] P. Apkarian, D. Noll, and O. Prot, "A proximity control algorithm to minimize nonsmooth and nonconvex semi-infinite maximum eigenvalue functions," J. Convex Anal, vol. 16, no. 3-4, pp. 641–666, 2009.

[68] L. Ding, Q. Han, L. Y. Wang, and E. Sindi, "Distributed cooperative optimal control of dc microgrids with communication delays," IEEE Transactions on Industrial Informatics, vol. 14, no. 9, pp. 3924–3935, 2018, doi: http://doi.org/10.1109/TII.2018.2799239.

[69] P. B. g. Dohmann and S. Hirche, "Distributed control for cooperative manipulation with event-triggered communication," IEEE Transactions on Robotics, vol. 36, no. 4, pp. 1038–1052, 2020, doi: http://doi.org/10.1109/TRO.2020.2973096.

[70] S. Hassan-Moghaddam and M. R. Jovanović, "Topology design for stochastically forced consensus networks," IEEE Transactions on Control of Network Systems, vol. 5, no. 3, pp. 1075–1086, 2018, doi: http://doi.org/10.1109/TCNS.2017.2674962.

[71] D. Yuan, D. W. C. Ho, and S. Xu, "Regularized primal–dual subgradient method for distributed constrained optimization," IEEE Transactions on Cybernetics, vol. 46, no. 9, pp. 2109–2118, 2016, doi: http://doi.org/10.1109/TCYB.2015.2464255.

[72] S. Huang, P. Li, Q. Wu, F. Li, and F. Rong, "Admm-based distributed optimal reactive power control for loss minimization of dfig-based wind farms," International Journal of Electrical Power & Energy Systems, vol. 118, p. 105827, 2020, doi: https://doi.org/10.1016/j.ijepes.2020.105827.

[73] B. Taner and K. Subbarao, "Distributed h_∞ edge weight synthesis for cooperative systems," in 2021 60th IEEE Conference on Decision and Control (CDC), 2021, pp. 6652–6658.

[74] M. Babazadeh and A. Nobakhti, "Direct synthesis of fixed-order h_∞ controllers," IEEE Transactions on Automatic Control, vol. 60, no. 10, pp. 2704–2709, 2015, doi: http://doi.org/10.1109/TAC.2015.2461832.

[75] C. Langbort, R. S. Chandra, and R. D'Andrea, "Distributed control design for systems interconnected over an arbitrary graph," IEEE Transactions on Automatic Control, vol. 49, no. 9, pp. 1502–1519, 2004, doi: http://doi.org/10.1109/TAC.2004.834123.

[76] E. T. Efaz, M. M. Mowlee, J. Jabin, I. Khan, and M. R. Islam, "Modeling of a high-speed and cost-effective FPV quadcopter for surveillance," in 2020 23rd International Conference on Computer and Information Technology (ICCIT). IEEE, dec 2020. [Online]. Available: https://doi.org/10.1109%2Ficcit51783.2020.9392696

[77] O. Andersson, P. Doherty, M. Lager, J.-O. Lindh, L. Persson, E. A. Topp, J. Tordenlid, and B. Wahlberg, "WARA-PS: a research arena for public safety demonstrations and autonomous collaborative rescue robotics experimentation," Autonomous Intelligent Systems, vol. 1, no. 1, nov 2021. [Online]. Available: https://doi.org/10.1007%2Fs43684-021-00009-9

[78] H. Saha, S. Basu, S. Auddy, R. Dey, A. Nandy, D. Pal, N. Roy, S. Jasu, A. Saha, S. Chattopadhyay, and T. Maity, "A low cost fully autonomous GPS (global positioning system) based quad copter for disaster management," in 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC). IEEE, jan 2018. [Online]. Available: https://doi.org/10.1109%2Fccwc.2018.8301782

[79] D. Guo and K. K. Leang, "Image-based estimation, planning, and control of a cable-suspended payload for package delivery," IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 2698–2705, apr 2020. [Online]. Available: https://doi.org/10.1109%2Flra.2020.2972855

[80] C. Estrada and L. Sun, "Trajectory tracking control of a drone-guided hose system for fluid delivery," in AIAA Scitech 2021 Forum. American Institute of Aeronautics and Astronautics, jan 2021. [Online]. Available: https://doi.org/10.2514%2F6.2021-1003

[81] T.-M. Nguyen, Z. Qiu, M. Cao, T. H. Nguyen, and L. Xie, "An integrated localization-navigation scheme for distance-based docking of UAVs," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, oct 2018. [Online]. Available: https://doi.org/10.1109%2Firos.2018.8594251

[82] B. Caruso, M. Fatakdawala, A. Patil, G. Chen, and M. Wilde, "Demonstration of in-flight docking between quadcopters and fixed-wing UAV," in 2021 IEEE Aerospace Conference (50100). IEEE, mar 2021. [Online]. Available: https://doi.org/10.1109%2Faero50100.2021.9438229

[83] D. Löbl, F. Holzapfel, M. Weiss, and T. Shima, "Cooperative docking guidance and control with application to civil autonomous aerial refueling," Journal of Guidance, Control, and Dynamics, vol. 44, no. 9, pp. 1638–1648, sep 2021. [Online]. Available: https://doi.org/10.2514%2F1.g004425

[84] H. L. N. N. Thanh, N. N. Phi, and S. K. Hong, "Simple nonlinear control of quadcopter for collision avoidance based on geometric approach in static environment," International Journal of Advanced Robotic Systems, vol. 15, no. 2, p. 172988141876757, mar 2018. [Online]. Available: https://doi.org/10.1177%2F1729881418767575

[85] L. D. DeVries and D. A. Paley, "Wake estimation and optimal control for autonomous aircraft in formation flight," in AIAA Guidance, Navigation, and Control (GNC) Conference. American Institute of Aeronautics and Astronautics, aug 2013. [Online]. Available: https://doi.org/10.2514%2F6.2013-4705

[86] A. Matus-Vargas, G. Rodriguez-Gomez, and J. Martinez-Carranza, "Ground effect on rotorcraft unmanned aerial vehicles: a review," Intelligent Service Robotics, vol. 14, no. 1, pp. 99–118, jan 2021. [Online]. Available: https://doi.org/10.1007%2Fs11370-020-00344-5

[87] P. Ru and K. Subbarao, "Nonlinear model predictive control for unmanned aerial vehicles," Aerospace, vol. 4, no. 2, p. 31, jun 2017. [Online]. Available: https://doi.org/10.3390%2Faerospace4020031

[88] D. Falanga, P. Foehn, P. Lu, and D. Scaramuzza, "PAMPC: Perception-aware model predictive control for quadrotors," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, oct 2018. [Online]. Available: https://doi.org/10.1109%2Firos.2018.8593739

[89] A. Weiss, M. Baldwin, R. S. Erwin, and I. Kolmanovsky, "Model predictive control for spacecraft rendezvous and docking: Strategies for handling constraints and case studies," IEEE Transactions on Control Systems Technology, vol. 23, no. 4, pp. 1638–1647, jul 2015. [Online]. Available: https://doi.org/10.1109%2Ftcst.2014.2379639

[90] F. Stesina, "Tracking model predictive control for docking maneuvers of a CubeSat with a big spacecraft," Aerospace, vol. 8, no. 8, p. 197, jul 2021. [Online]. Available: https://doi.org/10.3390%2Faerospace8080197

[91] K. Dong, J. Luo, Z. Dang, and L. Wei, "Tube-based robust output feedback model predictive control for autonomous rendezvous and docking with a tumbling target,"

Advances in Space Research, vol. 65, no. 4, pp. 1158–1181, feb 2020. [Online]. Available: https://doi.org/10.1016%2Fj.asr.2019.11.014

[92] O. Araar, N. Aouf, and I. Vitanov, "Vision based autonomous landing of multirotor UAV on moving platform," Journal of Intelligent & Robotic Systems, vol. 85, no. 2, pp. 369–384, aug 2016. [Online]. Available: https://doi.org/10.1007%2Fs10846-016-0399-z

[93] K. E. Wenzel, A. Masselli, and A. Zell, "Automatic take off, tracking and landing of a miniature UAV on a moving carrier vehicle," Journal of Intelligent & Robotic Systems, vol. 61, no. 1-4, pp. 221–238, oct 2010. [Online]. Available: https://doi.org/10.1007%2Fs10846-010-9473-0

[94] T. Muskardin, G. Balmer, L. Persson, S. Wlach, M. Laiacker, A. Ollero, and K. Kondak, "A novel landing system to increase payload capacity and operational availability of high altitude long endurance UAV," in 2016 International Conference on Unmanned Aircraft Systems (ICUAS). IEEE, jun 2016. [Online]. Available: https://doi.org/10.1109%2Ficuas.2016.7502668

[95] L. Persson, T. Muskardin, and B. Wahlberg, "Cooperative rendezvous of ground vehicle and aerial vehicle using model predictive control," in 2017 IEEE 56th Annual Conference on Decision and Control (CDC). IEEE, dec 2017. [Online]. Available: https://doi.org/10.1109%2Fcdc.2017.8264069

[96] L. Persson and B. Wahlberg, "Model predictive control for autonomous ship landing in a search and rescue scenario," in AIAA Scitech 2019 Forum. American Institute of Aeronautics and Astronautics, jan 2019. [Online]. Available: https://doi.org/10.2514%2F6.2019-1169

[97] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," ACM SIGGRAPH Computer Graphics, vol. 21, no. 4, pp. 25–34, aug 1987. [Online]. Available: https://doi.org/10.1145%2F37402.37406

[98] S. Zhang, H. Zhang, B. Di, and L. Song, "Cellular cooperative unmanned aerial vehicle networks with sense-and-send protocol," IEEE Internet of Things Journal, vol. 6, no. 2, pp. 1754–1767, apr 2019. [Online]. Available: https://doi.org/10.1109%2Fjiot.2018.2875140

[99] M. Mistry, J. Nakanishi, G. Cheng, and S. Schaal, "Inverse kinematics with floating base and constraints for full body humanoid robot control," in Humanoids 2008 - 8th IEEE-RAS International Conference on Humanoid Robots, 2008, pp. 22–27.

[100] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization," IEEE Robotics and Automation Letters, vol. 3, no. 3, pp. 1560–1567, 2018.

[101] Z. Zhou and Y. Zhao, "Accelerated admm based trajectory optimization for legged locomotion with coupled rigid body dynamics," 2020 American Control Conference (ACC), pp. 5082–5089, 2020.

[102] K. Yamamoto, T. Kamioka, and T. Sugihara, "Survey on model-based biped motion control for humanoid robots," Advanced Robotics, vol. 34, no. 21-22, pp. 1353–1369, 2020. [Online]. Available: https://doi.org/10.1080/01691864.2020.1837670

[103] D. Pardo, L. Möller, M. Neunert, A. W. Winkler, and J. Buchli, "Evaluating direct transcription and nonlinear optimization methods for robot motion planning," IEEE Robotics and Automation Letters, vol. 1, no. 2, pp. 946–953, 2016.

[104] C. Mastalli, R. Budhiraja, W. Merkt, G. Saurel, B. Hammoud, M. Naveau, J. Carpentier, S. Vijayakumar, and N. Mansard, "Crocoddyl: An efficient and versatile framework for multi-contact optimal control," 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 2536–2542, 2020.

[105] R. Budhiraja, J. Carpentier, C. Mastalli, and N. Mansard, "Differential dynamic programming for multi-phase rigid contact dynamics," in 2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids), 2018, pp. 1–9.

[106] A. Herzog, S. Schaal, and L. Righetti, "Structured contact force optimization for kino-dynamic motion generation," 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2703–2710, 2016.

[107] A. W. Winkler, C. Mastalli, I. Havoutis, M. Focchi, D. G. Caldwell, and C. Semini, "Planning and execution of dynamic whole-body locomotion for a hydraulic quadruped on challenging terrain," in 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 5148–5154.

[108] R. Budhiraja, J. Carpentier, and N. Mansard, "Dynamics consensus between centroidal and whole-body models for locomotion of legged robots," in 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 6727–6733.

[109] F. L. Lewis, H. Zhang, K. Hengster-Movric, and A. Das, Algebraic Graph Theory and Cooperative Control Consensus. Springer, London, 2013 doi: , ch. 2, pp. 23–70.

[110] A. Stanoev, S. Filiposka, V. In, and L. Kocarev, "Cooperative method for wireless sensor network localization," Ad Hoc Networks, vol. 40, pp. 61–72, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570870516300129

[111] Z. Yan, G. Gu, K. Zhao, Q. Wang, G. Li, X. Nie, H. Yang, and S. Du, "Integer linear programming based topology design for gnsss with inter-satellite links," IEEE Wireless Communications Letters, vol. 10, no. 2, pp. 286–290, 2021.

[112] E. P. van Horssen and S. Weiland, "Synthesis of distributed robust h-infinity controllers for interconnected discrete time systems," IEEE Transactions on Control of Network Systems, vol. 3, no. 3, pp. 286–295, 2016.

[113] P. Baerlocher and R. Boulic, "Task-priority formulations for the kinematic control of highly redundant articulated structures," in Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No.98CH36190). IEEE. [Online]. Available: https://doi.org/10.1109%2Firos.1998.724639

[114] R. Lind, "Linear parameter-varying modeling and control of structural dynamics with aerothermoelastic effects," AIAA Journal of Guidance, Control, and Dynamics, vol. 25, no. 4, pp. 733–739, 2002.

[115] G. J. Balas, "Linear, parameter-varying control and its application to aerospace systems," in 23rd Internatinoal Congress of Aeronautical Sciences, September 2002.

[116] F. Wu, X. H. Yang, A. Packard, and G. Becker, "Induced L2-norm control for LPV systems with bounded parameter variation rates," International Journal of Robust and Nonlinear Control, vol. 6, no. 9-10, pp. 983–998, 1996.

[117] M. S. Andersen, A. Hansson, S. K. Pakazad, and A. Rantzer, "Distributed robust stability analysis of interconnected uncertain systems," in 51st IEEE Conference on Decision and Control (CDC). IEEE, 2012, pp. 1548–1553.

[118] R. A. Horn and C. R. Johnson, Matrix analysis. Cambridge university press, 2012.

[119] K. Zhou, J. C. Doyle, and K. Glover, Robust and optimal control. Prentice Hall New Jersey, 1996.

[120] J. Ackermann, Robust control: the parameter space approach. Springer Science & Business Media, 2012.

[121] A. Shapiro, "First and second order analysis of nonlinear semidefinite programs," Mathematical Programming, vol. 77, no. 1, pp. 301–320, 1997.

[122] Q. Tran Dinh, S. Gumussoy, W. Michiels, and M. Diehl, "Combining convex–concave decompositions and linearization approaches for solving bmis, with

application to static output feedback," IEEE Transactions on Automatic Control, vol. 57, no. 6, pp. 1377–1390, 2012.

[123] Y. Wang, R. Rajamani, and A. Zemouche, "Sequential lmi approach for the design of a bmi-based robust observer state feedback controller with nonlinear uncertainties," International Journal of Robust and Nonlinear Control, vol. 28, no. 4, pp. 1246–1260, 2018.

[124] H. Pfifer and P. Seiler, "Robustness analysis of linear parameter varying systems using integral quadratic constraints," International Journal of Robust and Nonlinear Control, vol. 25, no. 15, pp. 2843–2864, 2015. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rnc.3240

[125] G.-R. Duan and H.-H. Yu, LMIs in control systems: analysis, design and applications, 1st ed. Boca Raton: CRC Press, 2013, ch. 2.

[126] J. Ackermann, P. Blue, T. Bunte, L. Guvenc, D. Kaesbauer, M. Kordt, M. Muhler, and D. Odenthal, Robust control: the parameter space approach. Springer, 2002, vol. 2.

[127] B. Taner and K. Subbarao, "Distributed h_∞ edge weight synthesis for cooperative systems," in 2021 60th IEEE Conference on Decision and Control (CDC). IEEE, dec 2021. [Online]. Available: https://doi.org/10.1109%2Fcdc45484.2021.9682966

[128] T. Anderson, C.-Y. Chang, and S. Martínez, "Distributed approximate newton algorithms and weight design for constrained optimization," Automatica, vol. 109, p. 108538, 2019, doi: https://doi.org/10.1016/j.automatica.2019.108538.

[129] J. W. Simpson-Porco, "Equilibrium-independent dissipativity with quadratic supply rates," IEEE Transactions on Automatic Control, vol. 64, no. 4, pp. 1440–1455, 2019, doi: http://doi.org/10.1109/TAC.2018.2838664.

[130] C.-e. Park, N. K. Kwon, and P. Park, "New bounded real lemma for singular markovian jump systems: Application to h_∞ control," International

Journal of Robust and Nonlinear Control, vol. 31, no. 3, pp. 907–919, 2021, doi: https://doi.org/10.1002/rnc.5340.

[131] M. Marcus and H. Minc, A survey of matrix theory and matrix inequalities. Courier Corporation, 1992, vol. 14.

[132] C. Kanzow, C. Nagel, H. Kato, and M. Fukushima, "Successive linearization methods for nonlinear semidefinite programs," Computational Optimization and Applications, vol. 31, no. 3, pp. 251–273, 2005.

[133] D. Bhattacharjee and K. Subbarao, "Robust control strategy for quadcopters using sliding mode control and model predictive control," in AIAA Scitech 2020 Forum. American Institute of Aeronautics and Astronautics, jan 2020. [Online]. Available: https://doi.org/10.2514%2F6.2020-2071

[134] T. Anderson, C.-Y. Chang, and S. Martínez, "Distributed approximate newton algorithms and weight design for constrained optimization," Automatica, vol. 109, p. 108538, nov 2019. [Online]. Available: https://doi.org/10.1016%2Fj.automatica.2019.108538

[135] J. T. Betts, Practical Methods for Optimal Control and Estimation Using Nonlinear Programming. Society for Industrial and Applied Mathematics, jan 2010. [Online]. Available: https://doi.org/10.1137%2F1.9780898718577

[136] E. Uzunoğlu, E. Tatlicioğlu, and M. İ. C. Dede, "A multi-priority controller for industrial macro-micro manipulation," Robotica, vol. 39, no. 2, pp. 217–232, may 2020. [Online]. Available: https://doi.org/10.1017%2Fs0263574720000338

[137] P. Baerlocher and R. Boulic, "Task-priority formulations for the kinematic control of highly redundant articulated structures," in Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No.98CH36190). IEEE. [Online]. Available: https://doi.org/10.1109%2Firos.1998.724639

[138] K. M. Lynch and F. C. Park, Modern Robotics: Mechanics, Planning, and Control, 1st ed. USA: Cambridge University Press, 2017.

[139] M. Khadiv, A. Herzog, S. A. A. Moosavian, and L. Righetti, "Walking control based on step timing adaptation," IEEE Transactions on Robotics, vol. 36, no. 3, pp. 629–643, Jun. 2020. [Online]. Available: https://doi.org/10.1109/tro.2020.2982584

[140] E. Daneshmand, M. Khadiv, F. Grimminger, and L. Righetti, "Variable horizon MPC with swing foot dynamics for bipedal walking control," IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 2349–2356, Apr. 2021. [Online]. Available: https://doi.org/10.1109/lra.2021.3061381

Biographical Statement

Baris Taner was born in Izmir, Turkey, in 1990. He received his Bachelor of Science in mechanical engineering from Izmir Institute of Technology (IZTECH), Turkey, in 2013. He started working in UKCAST in his senior year and continued working there until October 2013. He then attended M.Sc. program in Robotics in his alma mater, where he defended his thesis in February 2016. During his studies in IZTECH he worked as a research assistant in IZTECH Robotics Laboratory. While continuing his Ph.D. in IZTECH on Robotics, he latches onto aerospace and piloting, which leaded him to United States in Fall 2018 to pursue Ph.D. in Aerospace Engineering at Unversity of Texas at Arlington (UTA). During his graduate studies, Baris worked as a graduate research assistant at the Aerospace Systems Laboratory in UTA. He mainly works in the areas of dynamical systems, control theory, robust control, and cooperative control as applied to aerial unmanned systems and floating base robots (bipedal robots).