Practical Coordination of Multi-Vehicle Systems in Formation

by

İsmail Bayezit

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Mechanical Engineering

Waterloo, Ontario, Canada, 2014

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

This thesis considers the cooperation and coordination of multi vehicle systems cohesively in order to keep the formation geometry and provide the string stability. We first present the modeling of aerial and road vehicles representing different motion characteristics suitable for cooperative operations. Then, a set of three dimensional cohesive motion coordination and formation control schemes for teams of autonomous vehicles is proposed. The two main components of these schemes are $i)$ platform free high level online trajectory generation algorithms and $ii)$ individual trajectory tracking controllers. High level algorithms generate the desired trajectories for three dimensional leader-follower structured tight formations, and then distributed controllers provide the individual control of each agent for tracking the desired trajectories. The generic goal of the control scheme is to move the agents while maintaining the formation geometry. We propose a distributed control scheme to solve this problem utilizing the notions of graph rigidity and persistence as well as techniques of virtual target tracking and smooth switching. The distributed control scheme is developed by modeling the agent kinematics as a single-velocity integrator; nevertheless, extension to the cases with simplified kinematic and dynamic models of fixed-wing autonomous aerial vehicles and quadrotors is discussed. The cohesive cooperation in three dimensions is so beneficial for surveillance and reconnaissance activities with optimal geometries, operation security in military activities, more viable with autonomous flying, and future aeronautics aspects, such as fractionated spacecraft and tethered formation flying. We then focus on motion control task modeling for three dimensional agent kinematics and considering parametric uncertainties originated from inertial measurement noise. We design an adaptive controller to perform the three dimensional motion control task, paying attention to the parametric uncertainties, and employing a recently developed immersion and invariance based scheme. Next, the cooperative driving of road vehicles in a platoon and string stability concepts in one-dimensional traffic are discussed. Collaborative driving of commercial vehicles has significant advantages while platooning on highways, including increased road-capacity and reduced traffic congestion in daily traffic. Several companies in the automotive sector have started implementing driver assistance systems and adaptive cruise control (ACC) support, which enables implementation of high level cooperative algorithms with additional softwares and simple electronic modifications. In this context, the cooperative adaptive cruise control approach are discussed for specific urban and highway platooning missions. In addition, we provide details of vehicle parameters, mathematical models of control structures, and experimental tests for the validation of our models. Moreover, the impact of vehicle to vehicle communication in the existence of static road-side units are given. Finally, we propose a set of stability guaranteed controllers for highway platooning missions. Formal problem definition of highway platooning considering constant and velocity dependent spacing strategies, and formal string stability analysis are included. Additionally, we provide the design of novel intervehicle distance based priority coefficient of feed-forward

filter for robust platooning. In conclusion, the importance of increasing level of autonomy of single agents and platoon topology is discussed in performing cohesive coordination and collaborative driving missions and in mitigating sensory errors. Simulation and experimental results demonstrate the performance of our cohesive motion and string stable controllers, in addition we discuss application in formation control of autonomous multi-agent systems.

Acknowledgements

First, I am indebted to Prof. Barış Fidan and Prof. Jan P. Huissoon for guiding and contributing my research facility since the beginning of my study at University of Waterloo. I also thankful both of my advisors due to the prominent support and understanding during developing professional aspect in the field of control of intelligent ground and aerial systems. In addition, I thank to Prof. Steven Waslander for his support and encouragement during the period of our collaborative driving research. Besides, I am much obliged to Prof. Gökhan İnalhan for promoting me in understanding intelligent aerial platforms in the first period of my graduate study and Prof. Fevzi Ünal from Istanbul Technical University for his notable support and guidance in order to start my research at University of Waterloo.

Here, I also want to recall the great hospitality and understanding of all the members of Fontys Hogescholen Automotive Control Center during my collaborative driving research and electronic modification of Smart Fortwo vehicle in Helmond, Netherlands. Especially, I am grateful to my fellow engineer Tjalling Veldhuizen and Prof Henk Lupker from Fontys Automotive Control Center and engineer Gertjan Tillema and Prof. Marieke Martens from Center for Transport Studies at University of Twente for their pivotal assistance and guidance in designing collaborative driving strategies on real platforms in between January and May, 2011.

Finally, I am so thankful to senior research scientist Jeroen Ploeg and my labmates Arun Das and Siddhant Ahuja for their cooperation during my research at University of Waterloo.

Dedication

For my lovely wife Zeynep Bayezit,

and my family,

Dad, Mum, Tuğçe, Tuğba, and İbrahim

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Collaborative Control in Aerial and Ground Platforms

The *division of labour* concept is originally introduced in the first chapter of *The Wealth of Nations* by a Scottish moral philosopher Adam Smith, who is considered today as the father of modern economics and capitalism. In his doctrine, Adam Smith claimed dividing the whole work into a number of branches in order to increase daily productivity much more than manufacturing assigned the complete work to employees separately. In this manifest, the trade of a pin-maker is taken as an example of dividing the whole work into the branches, such as drawing out the wire, straightening, cutting, pointing, and grinding it at the top, etc. [78]. Therefore, pin-making is divided into about eighteen distinct operations and Adam Smith estimates rate of increase in productivity somewhere between 240 and 4800 times in his pin factory. This concept in economics or sociology also inspires cooperation of intelligent systems for increasing the efficiency in maintaining the large-scale complex missions, for instance, monitoring disaster areas and environment, situational awareness in special military operations, and homeland security tasks [19, 69]. Since the developments in sensor and wireless technology, we are more convenient performing collaborative missions in a short period of time. *Consensus dynamics* or *agreement dynamics* is an area of research focusing on agreement issues in multi-agent systems using system and graph theory.[1]

Research on multi-vehicle systems initially started in the mobile robotics field [55]. The development of inexpensive and reliable wireless communication modules accelerated the studies concerned with cooperative systems in the 1990s. First initiatives mostly focus on autonomous systems with the help of advanced vehicle sensors and communication systems, advanced GPS

[1] Wikipedia

assistance with digital mapping services, and advanced processing technology, high level control, and navigation algorithms. Then, researchers work on designing cooperative systems using intelligent vehicles, which are able to negotiate between each other and communicate with infrastructures, including sensors, softwares, and positioning technologies [2,55]. Instead of designing complex and large systems, it is much better to employ simpler designs with faster built times. Cooperative control and coordination of unmanned aerial vehicles (UAVs) have become a popular research field in the late 1990s and the early 2000s [48, 55, 87]. These systems are employed in many different application areas ranging from traffic density detection, controlling traffic congestion, payload transportation, searching agricultural lands, and many different civilian, and military cooperative missions [20, 28, 44, 47, 53, 67, 74]. In the corresponding studies, the system of interest is a collection of autonomous vehicles, cooperating with each other to perform a collective task. A particular problem of interest for aerial systems is to keep the shape of the agent collection (swarm) constant in the course of moving this swarm [37, 61, 79].
A particular goal of cooperative mission with land platforms is to keep the length of a vehicle platoon constant and ensure the string stability, while traveling at a desired speed. In this study, we consider the control of longitudinal motion of vehicle platoons at highways. In this context, we work on one-dimensional longitudinal motion with detailed constraints in order to mitigate the level of traffic congestion at highways. The first large-scale program in this field was California Partners for Advanced Transit and Highways (PATH) [60]. This program demonstrated the potential for multiple cars driving cooperatively in a platoon and consequently many other projects were initiated in Intelligent Transportation Systems (ITS), including SAfe Road TRains for the Environment (SARTRE)[2], 2007 Darpa Urban Challange, Grand Cooperative Driving Challenge (GCDC 2011) [47], Green ITS project at University of Waterloo, and European Lateral Safe in PReVENT and SafeMAP projects. Many other projects have also been promoted by governments and private sector entrepreneurs in Europe, Japan, and North America for developing driver assistance and driving safety systems contributing collaborative driving technology on the highways as the solution of daily traffic problems in modern countries [2]. In the final chapter of this study, we also discuss the integration of collaborative driving algorithms on road-vehicles incorporating static and dynamic road-side units.

1.2 Research Definition and Objectives

The cooperation of vehicles is widely studied topic with different scientific aspects by many engineering institutes. Here, we focus on practical implementation of distributed cohesive control missions in aerial and ground levels using different vehicle models. In this research, we first

[2] http://www.sartre-project.eu/en/Sidor/default.aspx

focus on modeling of single aerial and road vehicles due to cooperative missions. The simplified experimental vehicle models enable implementing one and multi-dimensional distributed cohesive motion control algorithms for unmanned formations. For the ease of understanding, cohesive control missions can be described in three different layers as demonstrated in Fig. 1.1. In the *path generation layer* of the implementation, the algorithm deal with generating three dimensional trajectories due to the sphere intersection theory. In addition, we also determine the type of cooperation as leader-follower based formation here. Second, *control layer* is responsible for defining control method of individual vehicle up to the complexity of the agent model and quality of sensor measurement. Finally, *system model layer* gives the dynamics and kinematics of the aerial platform.

Next, platooning of road vehicles with detailed objectives is considered regarding to string sta-

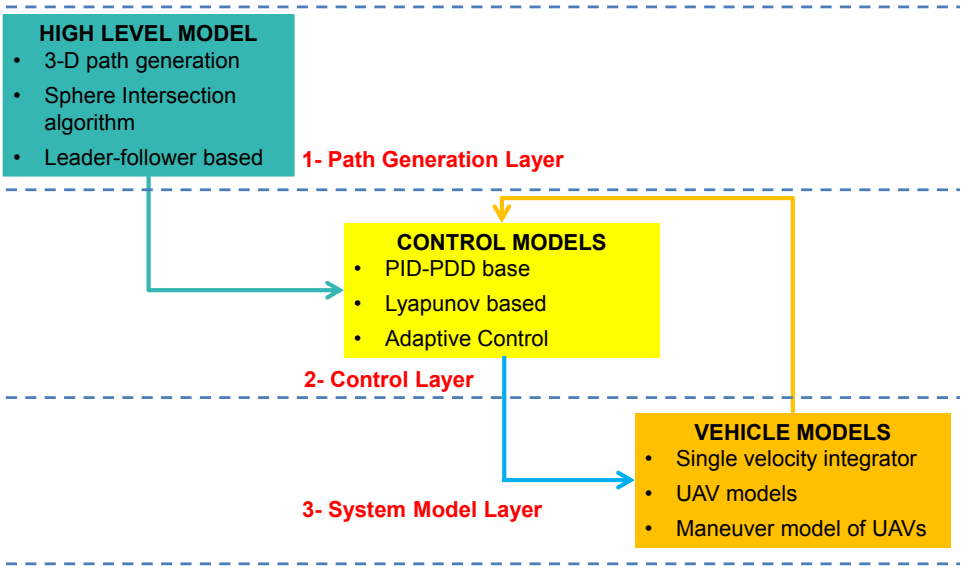


Figure 1.1: Implementation scheme of cohesive motion control to flight vehicle formations

bility aspect. For developing feasible Cooperative Adaptive Cruise Control (CACC) algorithms, first we design a simulation model using mathematical representation of the vehicle dynamics, actuator dynamics, and vehicle kinematics. Then, we test our controllers in realistic scenarios with simulations. Afterwards, the validated controllers are tested on Smart Fortwo cars. Finally, we give the formal analysis of our proposed control models considering one-vehicle ahead or multiple preceding vehicles for constant and variable spacing models. The impact of different platoon formations on traffic congestion and advantages of using static or dynamic roadside units are also discussed.

1.3 Contributions

The main contributions claimed within this thesis are as follows:

- Extension from 2-D, single integrator model agent with global frame setting to 3-D, detailed model agent with local (relative sensing) setting [4, 6, 7],
- A novel design of platform free 3-D trajectory generation algorithm according to sphere intersection theory [7],
- Implementation of immersion and invariance based adaptive control on UAV maneuvers with measurement noise [34],
- Designing specific electronic architecture for real time implementation of longitudinal highway platooning algorithms [10, 11],
- Cooperative cruise controller design during different data-sharing topologies [10, 11],
- Stability analysis for Constant vs. Velocity dependent intervehicle spacing,
- A novel design of intervehicle spacing based priority filter for wireless state information from preceding vehicles in order to mitigate higher time delay factors [9].

1.4 Thesis Organization

The remainder of this thesis is organized as follows:

Chapter 2 details three vehicle models representing different motion characteristics and constraints. Modeling and experimental design stages of small-size fixed wing aerial and quadrotor vehicles are given implementing cooperative tasks. We provide all steps of linearization and model decomposition in these aerial platforms. Next, the longitudinal modeling of realistic road vehicles is described and electronic modification of Smart Fortwo car is detailed for implementing platooning missions. The CarSim and Matlab/Simulink softwares are used to identify the parameters of different road vehicle models with asymmetric acceleration and deceleration characteristics.

Chapter 3 gives the mathematical definition of three-dimensional cohesive formations and design a leader-follower based online trajectory generation algorithm due to sphere intersection algorithm. Then, the cohesive motion control task is performed with quadrotor and fixed-wing aerial vehicle swarms. In the implementation of distributed control missions, we tested cohesive motion of leader-follower based formations with the help of linear PID, PDD, and Lyapunov based controllers [4, 6, 8].

Chapter 4 provides immersion and invariance based adaptive autonomous maneuver control of UAVs with measurement error. Additionally, the application of immersion and invariance based adaptive control to formation control and its advantageous are discussed.

Chapter 5 presents developing of high level adaptive cooperative cruise control algorithms improving one-dimensional flow of daily traffic and reducing its adverse effects. Generic high level control algorithms are also described for different highway and urban missions considering data from preceding vehicle only.

Chapter 6 provides formal problem definition of highway platooning and formal analysis of string stability with different spacing policy and formation control structures. For implementing collaborative driving algorithms and generating desired control signals successfully, we provide Vehicle to Vehicle (V2V), and Vehicle to Road Side Unit (V2I) communication. We assume that all vehicles can derive their state information with the help of an integrated onboard sensor suite. We also describe the design of novel intervehicle distance based priority filtering of the data transmitted through different preceding vehicles in order to improve string stability during highway platooning.

Chapter 7 presents conclusions of the thesis and demonstrates the future research directions in this area.

Chapter 2

Design and Implementation of Autonomy on Single Aerial and Ground Platforms

2.1 Introduction

This chapter presents the experimental designs and modifications on aerial and ground platforms implementing one-dimensional and multi-dimensional cohesive tasks detailed in subsequent chapters. The models of three different vehicles representing different motion behaviors are provided. First, we obtain the small-fixed wing UAV model. The motion characteristics can be simplified and decomposed as lateral and longitudinal dynamics in three-dimensional space. The lateral motion includes rolling and yawing maneuvers, which can be considered as nonholonomic constraint. Therefore, roll and yaw turns are accepted as joint actions and this nonholonomic effect can be seen in the planar motion characteristics of piccolo UAV model given in Chapter 3. Additionally, the longitudinal motion consists of pitch maneuvers and airspeed. As a result, we simplify the motion of fixed-wing UAV with two separate coupled dynamics in lateral and longitudinal axes. Then, we provide the simplified motion dynamics of quadrotor vehicles in three-dimensional axis. The motion of quadrotor vehicle is obviously different than the fixed-wing UAV. All rotations are independent of each other and generated by arranging thrust forces of propellers correspondingly. Furthermore, we are able to make a hover flight balancing the weight of quadrotor with total thrust. Navigation of quadrotor vehicle is achieved with controlling the rotation angles. To this end, the design of reference angle generator is given for defining the desired angle input corresponding to the desired position for quadrotor vehicle. Finally, we give the longitudinal dynamics of road vehicles with first order actuator delays. This motion can be described in one-dimensional space considering internal first order constraints for implement-

ing platooning tasks to increase string stability of the cooperative driving. These different motion models of mostly used testbeds are taken for developing platooning and cohesive motion control algorithms and testing the efficiency of cohesive coordination missions with various practical platforms.

2.2 Modeling Small-Size Fixed Wing UAV

Autonomous systems are a popular and fast-growing technology, with numerous applications including unmanned aerial vehicles, intelligent vehicle cruise control, and cooperative agents. Small-scale unmanned aerial vehicles are widely used in research institutes to test diverse control algorithms and sensor suites for different levels of autonomy with specific metrics [24]. Small-scale conventional UAVs are also actively used in many military and non-military operations, such as search and track applications, and cooperative flights. Small scale fixed-wing UAVs are therefore an emerging technology, and have been widely used in many manned/unmanned and cooperative operations as research testbeds. This section describes the realistic modeling of a small-scale UAV for autonomous control that serves as an agent in three dimensional cohesive formations and a dynamic roadside unit to provide critical roadside assistance of highway traffic. Unmanned systems are primarily targeted at reducing the required level of operator work load, and also enable operation under difficult conditions.

It is important to precisely specify the desired level of autonomy and the requirements of the autonomous system before starting flight operations, since UAV operation may include many complexities and nonlinearities due to internal dynamics, weather conditions, as well as interaction with other agents [14]. In this work, we consider autonomous UAV acting as a single-agent in three-dimensional cohesive missions or as an expert-node providing assistance for road platoons. The avionics hardware provides a great deal of sensory information for determining the state parameters of the UAV model under the operating conditions, and this hardware enable to generate adaptive control parameters for flying over and directing the highway traffic. The UAV also needs to communicate with other nodes, such as ground stations and other autonomous agents, to implement higher level tasks as summarized in [22].

After selecting the avionics hardware for implementing autonomous waypoint tracking in level flight, the small-scale UAV can be conveniently used for executing the role of a dynamic agent and help ease traffic congestion by transmitting online optimal cruise accelerations to the platoons on the highway. The flight of the UAV starts with manual take-off, and the pilot then switches the UAV to autonomous mode. As soon as the adaptive autopilot takes control, the UAV starts to implement the traffic control mission and tracks its desired trajectory, which is defined by corner waypoints and a path planning algorithm. On completing its mission, the pilot

switches it back to manual mode and lands the UAV. During the low level flight, the UAV can interact with highway traffic via camera or wireless transmission.
In the following, Section 2.2.1 describes a generic implementation of autonomous flight and waypoint navigation algorithms; Section 2.2.2 gives the kinematics, nonlinear dynamics, and decomposed longitudinal and lateral behaviors of a small-scale UAV platform; Section 2.2.3 provides experimental methods for calculating the moment of inertia (pendulum tests), servo tests and mathematical modeling of UAV.

2.2.1 Implementation of Autonomous Flight Mission

The primary objective of the autopilot flight system includes pitch attitude hold, altitude hold, air-speed hold, automatic take-off and landing, roll-angle hold, turn coordination, and heading hold [21]. Linear successive feedback control loops are typically designed to meet the performance requirements of each operating point and maneuvering set of the aerial vehicle. After designing feedback controllers covering N operating points, the sets of predesigned constant controller gains are switched to ensure stable flight of the UAV under different flight conditions [72].
Since more intelligence is required in aerial vehicles, the level of uncertainty is increased [65]. The main reason for selecting trimmed linear models over nonlinear models with small-scale UAVs can be explained by the simplicity of feedback control implementation on the trimmed model, availability of model decomposition of the full flight envelope, and external uncertainties and instabilities of internal dynamics due to natural disturbances and nonlinearities in the full flight model [14].
In this subsection, we focus on autonomous adaptive waypoint tracking of the UAV at highways for detecting and transmitting the optimum required acceleration for traffic stability. Instead of using pre-generated controller gains, the selection of adaptive controller gains enables online estimation of flight parameters and overcomes unpredictable variations with retuning control parameters during slight changes in the same flight regime [72]. Fig. 2.1 illustrate the details of autonomous waypoint tracking implementing adaptive control. Autonomous waypoint navigation controller generates the signal for UAV actuators, such as throttle pedal, elevator, rudder, and ailerons. Sensor suite is used for measuring actual positions, velocity, and rotation rates. Then, all these measurements are converted using rotation matrices in order to find the exact 3-D location and rotations with respect to the global frame. In respect to autonomous waypoint control implementation, we first convert the nonlinear UAV dynamics to decompose lateral and longitudinal modes for trimmed operating conditions. A Tornado T135 model aircraft was used to obtain the stability derivatives as described in [5]. The physical dimensions of the aircraft were measured, as were the mass and centre of gravity (including the avionics), and reference

conditions such as altitude and air speed were defined. The servo saturation and delays, as well as the moments of inertia about the principle axes, were determined experimentally. We were then able to calculate the longitudinal and lateral coefficients in the state and input matrices of the UAV [13, 51, 71]. The next step was to determine the operating modes of the system and decide on the performance and the efficiency of the control depending on the pure dynamics of the lateral and longitudinal model [64, 81].

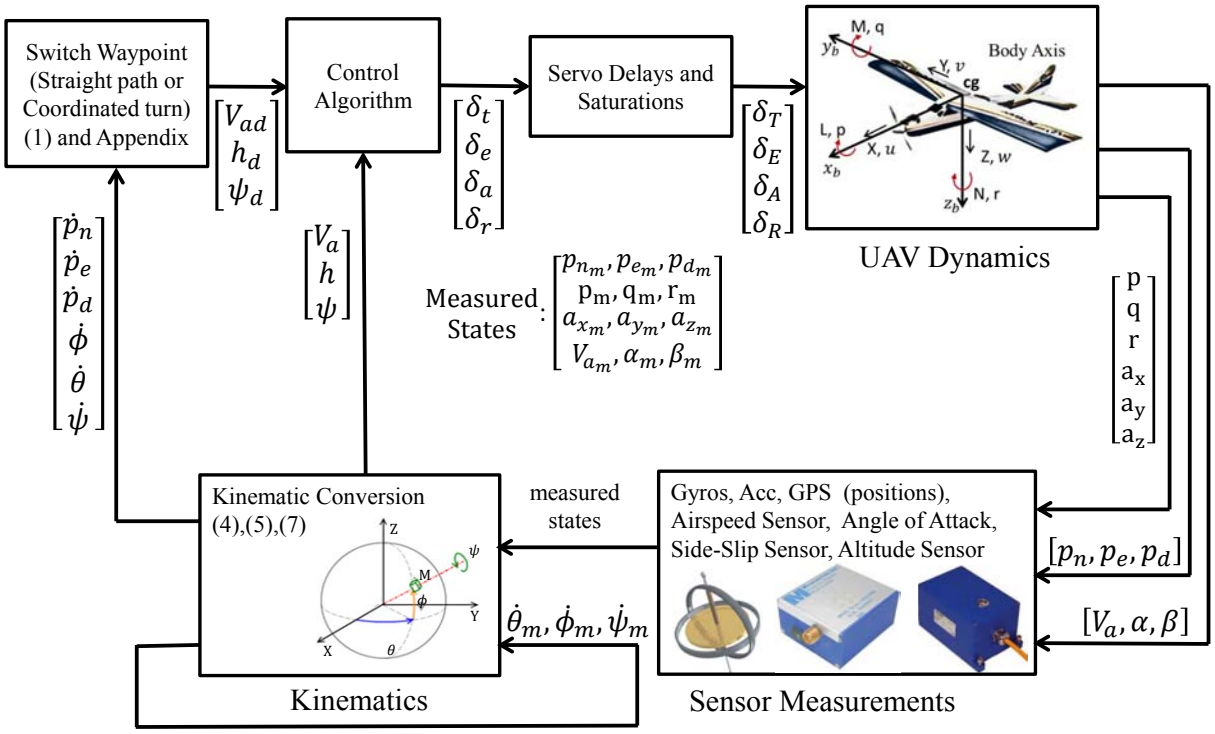


Figure 2.1: Conceptual Logic of Autonomous Control on a Small Scale UAV

Waypoint Switching Strategy

The flight path of our small-scale UAV was designed with an ordered sequence of waypoints $\mathrm{w}_i = (p_{n,i}, p_{e,i}, p_{d,i}) \in \mathbb{R}^3$ in the inertial frame. As illustrated in Fig. 2.2, our UAV first tracked the path $\overline{\mathrm{w}_{i-1}\mathrm{w}_i}$ between waypoints w_{i-1} and w_i. When the UAV enters the sphere $S(\mathrm{w}_i, \varepsilon) = \{p | \|p - \mathrm{w}_i\| \leq \varepsilon\}$ centered on waypoint w_i, it switches to the next waypoint w_{i+1} to avoid chattering due to disturbances and sensor errors [14].

In the course of the flight, we apply a two level strategy for tracking the path and switching to the next waypoint. We start by generating the next waypoint, and once we are able to define a straight path between the actual position $p(t)$ of the UAV and the desired waypoint w_i , we calculate the desired heading angle $\psi_d(t)$ of the UAV. In the second level, the path planning algorithm is used to define the turning maneuver to reach the next waypoint w_{i+1} as illustrated in Fig 2.2. As soon as the UAV enters the sphere $S(\mathrm{w}_i, \varepsilon)$, the pre-defined waypoint w_{i+1} is generated. The details of autonomous switching and turning are given in Appendix A.

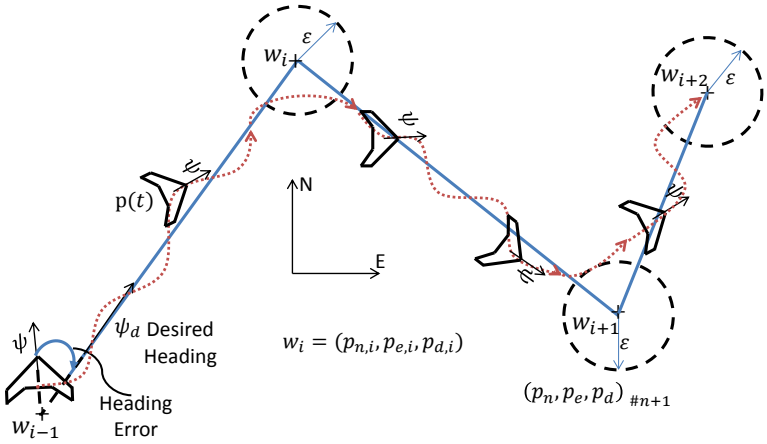


Figure 2.2: Waypoint switching and path tracking strategy during traffic flow surveillance at highways

2.2.2 Kinematic and Dynamic Modeling

Kinematic Modeling

The vector $p(t) = [p_n, p_e, p_d]^T$ is used to describe the position of the UAV in three-dimensional earth fixed frame. In the same inertial frame, the attitude of the UAV is described with the vector $\bar{p}(t) = [\phi, \theta, \psi]^T$, where the angles ϕ, θ and ψ are the vehicle's roll, pitch, and yaw angles and represent the rotations with respect to x, y, and z axes of inertial frame, respectively [13].
The linear velocities and angular rates of the UAV are introduced with the following notation in a body fixed frame:

$$V^b = [u, v, w]^T \tag{2.1}$$

$$\Omega^b = [p, q, r]^T \tag{2.2}$$

where u, v, and w are linear velocities expressed in body frame x, y, and z-axis, respectively. Additionally, p_i, q_i, and r_i are angular rates in roll, pitch and yaw axis, respectively.

Remark 1 *The inertial forces in the dynamic equations of the UAV are due to accelerations and velocities, which are relative to inertial frame. However, the aerodynamic forces are dependent on airspeed V_a relative to the wind frame [13].*

As mentioned in Remark 1, the effect of the wind experienced by the UAV leads to angle-of-attack α and side-slip angle β. The airspeed V_a relative to the the wind frame and conversion of the speed from wind frame to the body frame can be given as:

$$V_a = V - V_w \tag{2.3}$$

$$V_a^b = V^b - V_w^b = \begin{pmatrix} \cos\alpha\cos\beta \\ \sin\beta \\ \sin\alpha\cos\beta \end{pmatrix} V_a \tag{2.4}$$

where V_w and V_a^b denote, respectively, the velocity vector of the wind relative to the inertial frame and the velocity of the UAV relative to the body frame [14]. Since we avoid gimbal locks or aggressive actions in the maneuvers of our UAV, three dimensional rotation matrices and (2.4) are adequate to make the conversions during autonomous flight.

Nonlinear UAV Dynamics and Model Decomposition

The reliability of the mathematical modeling and experimental testing process is important in defining the true control parameters for specific flight regimes. As a first approximation, the small scale model aircraft can be simply considered as a rigid body, and we can therefore start modeling the UAV with Newton's force and moment equations. Newton's rigid body equations are transformed with the vectorial representation into the resulting forces considering the gravitational forces, as follows:

$$\begin{aligned} \mathrm{F_x} &= m(\dot{\mathrm{u}} + \mathrm{qw} - \mathrm{rv}) + \mathrm{g}\sin\theta \\ \mathrm{F_y} &= m(\dot{\mathrm{v}} + \mathrm{ru} - \mathrm{pw}) - \mathrm{g}\cos\theta\sin\phi \\ \mathrm{F_z} &= m(\dot{\mathrm{w}} + \mathrm{pv} - \mathrm{qu}) - \mathrm{g}\cos\theta\cos\phi \end{aligned} \tag{2.5}$$

Additionally, the moment equations are obtained as in below:

$$\begin{aligned} \mathrm{M_x} &= \dot{\mathrm{p}}I_x - \dot{\mathrm{r}}I_{xz} + \mathrm{qr}(\mathrm{I_z} - \mathrm{I_y}) - \mathrm{qpI_{xz}} \\ \mathrm{M_y} &= \dot{\mathrm{q}}I_y + \mathrm{pr}(\mathrm{I_x} - \mathrm{I_z}) + (\mathrm{p}^2 - \mathrm{r}^2)\mathrm{I_{xz}} \\ \mathrm{M_z} &= \dot{\mathrm{r}}I_z - \dot{\mathrm{p}}I_{xz} + \mathrm{pq}(\mathrm{I_y} - \mathrm{I_x}) + \mathrm{qrI_{xz}} \end{aligned} \tag{2.6}$$

The equations (2.5) and (2.6) are obtained as nonlinear equations of UAV motion [71]. However, most of maneuvers of interest are controlled assuming decoupled lateral (roll and yaw actions) and longitudinal (airspeed, pitch, and climb/descent actions) motion dynamics [14]. Small perturbation theory is appropriate for decomposing lateral and longitudinal dynamics for a specific operation condition, such as desired altitude, desired airspeed, and angles. Most of the current commercial autopilot systems (Procerus Kestrel, MicroPilot MP series, Cloup Cap Piccolo, and UNAV 3500) use model decomposition and PID based linear control strategies with gain tuning, which yields to better performance in different phases of flight [13, 21].
The nonlinear force and moment equations of UAV flight dynamics can be formulated as:

$$\dot{x} = f(x, u) \tag{2.7}$$

where x is the state of the system, and u is the input. For the linearization of dynamic force and moment equations, our UAV platform have a flight at desired trim conditions (pitch angle, angle of attack, airspeed), which can be represented with a state x^*, and input u^* [82]. In our mission scenario, the UAV is doing surveillance, collecting traffic data, and managing traffic flow commands with interacting roadside units in proximity.
At first, our UAV is making wings-level steady flight with a constant-altitude for our traffic surveillance mission. Additionally, we need to fly with a constant yaw angle (straight path following) or constant yaw-rate (coordinated-turn) with growing yaw-angle in order to analyze traffic flow characteristics. As a result, the objective is to calculate trim parameters when UAV satisfies the required trim conditions, such as traveling at a constant speed V_a^*, climbing at a constant flight path angle γ^*, and constant orbit of radius R^* [14]. The aerodynamic force and moment equations are divided into the mean and the perturbed parts in consequence of trim operation [25, 51].
The Taylor expansion is applied for calculation of the linearized equations of the flight considering aerodynamic, gravitational, and thrust terms. Then, we fully decouple the longitudinal and lateral dynamics. As a result, elevator deflection is only the control input for longitudinal motion dynamics. Because of the constant velocity condition to simplify nonlinear dynamics, the throttle position is fixed and it is neglected in longitudinal equations. After the cancelation of negligible stability derivatives $X_{\dot{u}}$, $X_q, X_{\dot{w}}$, X_{δ_E}, $Z_{\dot{u}}$, $Z_{\dot{w}}$, $M_{\dot{u}}$, $Z_{\dot{\delta}_E}$, and $M_{\dot{\delta}_E}$ [14, 25, 51], the longitudinal model equations are obtained as:

$$\begin{aligned}
\dot{\bar{\mathrm{u}}} &= X_u\bar{\mathrm{u}} + X_w V_a^* c\alpha^*\bar{\alpha} - X_q\bar{q} - gc\theta^*\bar{\theta} + X_{\delta_e}\delta_e \\
\dot{\bar{\alpha}} &= Z_u\bar{\mathrm{u}} + Z_w\bar{\alpha} + Z_q\bar{q} - gs\theta^*\bar{\theta} + Z_{\delta_e}\delta_e \\
\dot{\bar{q}} &= M_u\bar{\mathrm{u}} + M_w V_a^* c\alpha^*\bar{\alpha} + M_q\bar{q} + M_{\delta_e}\delta_e \\
\dot{\bar{\theta}} &= \bar{q}
\end{aligned} \tag{2.8}$$

Then, the decoupled lateral dynamics includes yaw and roll rotations, which is obtained with the series expansion for the rudder and aileron actions [14, 25, 51]. After neglecting insignificant stability derivatives in lateral model, such as $Y_{\dot{v}}$, $Y_p, Y_{\dot{p}}$, Y_r, $Y_{\dot{r}}$, $Y_{\delta A}$, $L_{\dot{v}}$, $L_{\dot{r}}$, $N_{\dot{v}}$, $N_{\dot{r}}$, and I_{xz}[1], lateral equations are obtained as:

$$\begin{aligned}
\dot{\bar{\beta}} &= Y_v\bar{\beta} + Y_p\bar{p} + Y_r\bar{r} + gc\theta^* c\phi^*\bar{\phi} + Y_{\delta_a}\delta_a + Y_{\delta_r}\delta_r \\
\dot{\bar{p}} &= L_v V_a^* c\beta^*\bar{\beta} + L_p\bar{p} + L_r\bar{r} + L_{\delta_a}\delta_a + L_{\delta_r}\delta_r \\
\dot{r} &= N_v V_a^* c\beta^*\bar{\beta} + N_p\bar{p} + N_r\bar{r} + N_{\delta_a}\delta_a + N_{\delta_r}\delta_r \\
\dot{\bar{\phi}} &= \bar{p} + c\phi^* t\theta^*(\bar{r} + q^*\bar{\phi}) - s\phi^* t\theta^* r^*\bar{\phi} \\
\dot{\bar{\psi}} &= c\phi^* \sec\theta^*(\bar{r} + p^*\bar{\phi}) - s\phi^* \sec\theta^* r^*\bar{\phi}
\end{aligned} \tag{2.9}$$

where $s\cdot$, $c\cdot$, and $t\cdot$ denote sine, cosine, and tangent of the Euler angles, respectively. As it is seen from (2.8) and (2.9), the nonlinear equations of UAV flight motion are split into two decoupled rotational and translational equation sets, which are lateral and longitudinal dynamics. The longitudinal characteristics of the derived flight model are called short-period mode (faster poles with low damping ratio) and phugoid mode (slower poles with high damping ratio) depending on the locations in s-plane. Then, the lateral characteristics are called spiral mode (slower pole with no damping), roll subsidence mode (faster pole with no damping), and dutch-roll mode (complex conjugate poles with the lowest damping ratio). Short period mode is the reduced longitudinal dynamics for constant altitude and constant thrust deflection, and phugoid mode when we assume a constant angle of attack α. Reduced lateral modes are classified as roll subsidence mode when we ignore heading dynamics and assume a constant pitch angle θ, spiral mode when we neglect rudder command and assume $\dot{\bar{p}}$, $\bar{p}$ zero, and dutch-roll mode when we ignore rolling motions and focus yawing actions [14, 25].

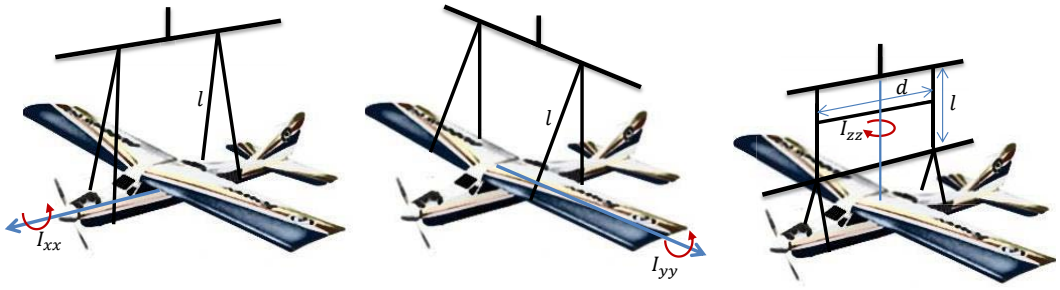
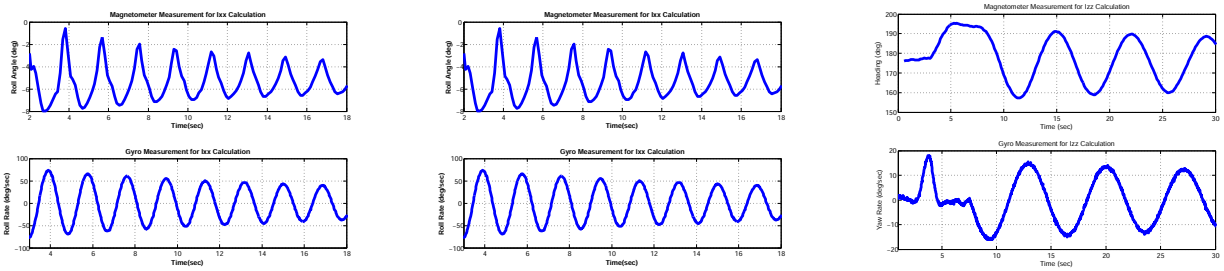


Figure 2.3: Pendulum setup for moment of inertia calculation

[1]The I_{xz} inertia term is assumed to be zero in most of the small-scale UAV applications because of the difficulty in experimental calculation and insignificance with respect to inertia of main axes.



(a) I_{xx} Measurement (b) I_{yy} Measurement (c) I_{zz} Measurement

Figure 2.4: Experimental measurements of the inertias of small scale UAV

2.2.3 Experimental Methods for Parameter Identification

In this part, the calculation of stability derivatives of fixed-wing UAV are achieved using our geometrical measurements, pendulum test results, and required trim conditions. At first, geometric lengths are measured, the trimmed airspeed is determined as 20 m/s and the alpha (α) and beta (β) angles are determined as 0 degree. The aerodynamic coefficients and stability derivatives of our assigned UAV were obtained using Tornado software. These coefficients are used for computing the parameters of lateral and longitudinal equation. The generated lift is 53.27 N, which is equal to the weight of Trainer60 with its microavionics hardware [5].

Pendulum Tests

Experimental pendulum methods are used to estimate the diagonal elements of inertia tensor as it is seen in Fig. 2.3. The calculation of I_x, I_y, and I_z inertia values are performed with pendulum techniques as depicted in Fig. 2.3 and Fig 2.4. The calculations of inertias about x_b, y_b, and z_b axis are given as:

$$I_\varsigma = \frac{mgT_\varsigma^2 l}{4\pi^2} - ml^2, \quad \varsigma =: x, y \tag{2.10}$$

$$I_z = \frac{mgd^2T_z^2}{16\pi^2 l} \tag{2.11}$$

where m, g, l, T, and d are mass of the UAV with the avionics, gravitational constant, length of the rope, period of oscillation, and distance between two ropes as illustrated in Fig 2.3, respectively. The parameters m, l, d, T_x, T_y, and T_z for the calculation of inertia values obtained are 5.36 kg, 0.93 m, 0.2 m, 2 sec, 1.8 sec, and 7 sec, respectively. Equations (2.10), and (2.11) are used for calculating I_x, I_y and I_z inertia values [63]. These equations were tested and validated

many times for small fixed-wing aircrafts with the help of gyroscope and magnetometer. Coupled inertia parameters $I_{xy} = I_{yx}$ and $I_{yz} = I_{zy}$ are taken to be zero because of complex experimental calculation procedures and low importance in comparison with I_x, I_y and I_z [64, 81].
An IMU is used as the main measurement sensor in order to calculate the oscillation period of the UAV as in [63]. Crista IMU and 3-axis magnetometer are used to calculate the rotations and rotation rates about the related axis as in Fig. 2.3. The gyro and magnetometer measurements demonstrate same characteristics as it is predicted due to the oscillation about the related axis and identify the angle and angular rate of the same oscillation. The pure characteristic and the derivatives of the cycles are sinusoidal. All trials were repeated and compared many times for filtering noise affects in the final calculations.

Servo Tests

Small scale fixed wing UAV needs servo actuators for the autonomous deflection of its aileron, elevator, rudder surfaces, and throttle lever. In this experiment, we measured the maximum, minimum, and middle deflection angles of control surfaces as shown in Fig. 2.5. This step is one of the important steps for control design in order to derive saturation limits and actuator delays of our UAV in real flight.

The limits of elevator deflection (pitch attitude and altitude), aileron deflection (roll attitude and course angle), and rudder deflection (sideslip angle) were measured as $\pm 20^\circ$, $\pm 20^\circ$, and $\pm 15^\circ$, respectively. As a result, we were able to calculate the required level of input deflection and regarding Pulse Width Modulation (PWM) cycle during real flight tests in order to give autonomous commands. Additionally, we derived the velocity response of servo actuators $\bar{\delta}_{des} = \{K_\delta/\tau_\delta s + 1\}\delta_{des}$ for our command surfaces as a first order delay with a constant gain.

System Model of an UAV

Using the procedure described in Section 2.2.2, we derived a state space model of lateral and longitudinal dynamics for a fixed condition. The system model state space coefficients are calculated considering our experimental pendulum and servo testing results, geometrical calculations, aerodynamic parameters derived with Tornado and Matlab, and mathematical approaches presented in [14, 25, 51]. The simplified lateral model for waypoint tracking is derived as follows:

$$\begin{bmatrix} \dot{\phi} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_{\phi_1} & -a_{\phi_2} \end{bmatrix} \begin{bmatrix} \phi \\ p \end{bmatrix} + \begin{bmatrix} 0 \\ b_{\phi_1} \end{bmatrix} \phi_c \tag{2.12}$$

$$\begin{bmatrix} \dot{\chi} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_{\chi_1} & -a_{\chi_2} \end{bmatrix} \begin{bmatrix} \chi \\ r \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ b_{\chi_1} & b_{\chi_2} \end{bmatrix} \begin{bmatrix} \chi_c \\ \dot{\chi}_c \end{bmatrix} \tag{2.13}$$

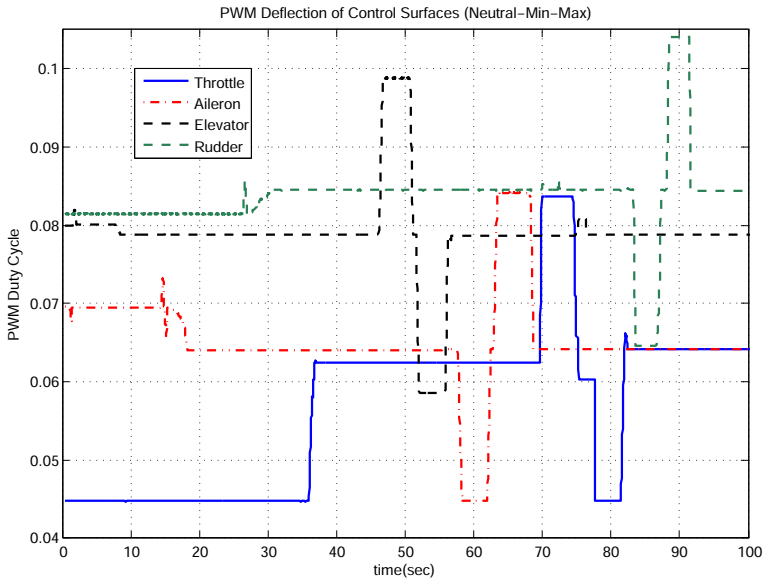


Figure 2.5: PWM duty cycle values of throttle, aileron, elevator, and rudder servos of Trainer 60 UAV

Next, we defined longitudinal model for autopilot design [14]. The final simplified longitudinal model obtained is:

$$\begin{bmatrix} \dot{\theta} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_{\theta_1} & -a_{\theta_2} \end{bmatrix} \begin{bmatrix} \theta \\ q \end{bmatrix} + \begin{bmatrix} 0 \\ b_{\theta_1} \end{bmatrix} \theta_c \tag{2.14}$$

$$\begin{bmatrix} \dot{h} \\ \dot{\mathrm{w}} \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ -a_{h_1} & -a_{h_2} \end{bmatrix} \begin{bmatrix} h \\ \mathrm{w} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ b_{h_1} & b_{h_2} \end{bmatrix} \begin{bmatrix} h_c \\ \dot{h}_c \end{bmatrix} \tag{2.15}$$

In this final control model, we assumed rudder and throttle are fixed and ignored the wind force in the course of flight due to the fixed airspeed and zero sideslip conditions. The disturbances and other minor external affects are considered as parameter variations of linearized model calculated in (2.8), (2.9), and (2.12)−(2.15). The order of the disturbances are similar to the system model and the affect of these variations are compensated with the help of controllers. In the following section, we provide the simplified modeling of quadrotor vehicles and the generic control control implementation with avionics.

2.3 Modeling of Quadrotor Agents

Quadrotor vehicles are controlled with the actions of four propellers as illustrated in Fig. 2.6. The vehicle positions are arranged with the help of rotations. Pitch, yaw, and roll orientations are independent of each other. These angular actions enable quadrotor moving in three-dimensional space.

We provide in Fig. 2.6 four different well-known quadrotor testbeds for control and coordination

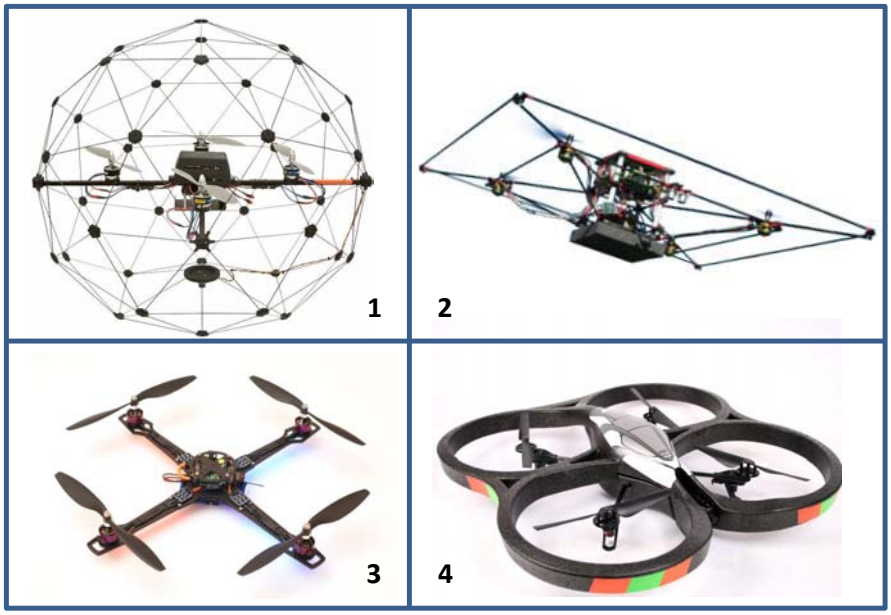


Figure 2.6: Different quadrotor vehicles for autonomous control implementation: 1) Quanser Qball-X4; 2) The Stanford/Berkeley Testbed of Autonomous Rotorcraft for Multi-Agent Control (STARMAC); 3) Microcopter Quadrotor Vehicle; 4) Parrot AR Drone 2.0

applications. The generic control diagram for Parrot AR Drone or other quadrotor vehicles is illustrated by Fig. 2.7. The quadrotor control mechanism can be split into four parts: main control unit, electronic speed control of propeller motors; brushless motors and propeller; the position and orientation of propellers; quadrotor dynamics [73]. The microprocessor in main control unit generates the desired PWM signals for changing the orientation of quadotors, then the electronic speed controller converts these signals in accordance with the characteristics of propeller rotors for generating desired thrusts. Next, the applied lift and torques are obtained based on the geometric distances and orientations of propellers according to center of mass. Finally, the quadrotor vehicle is moving to the desired location with a desired speed based on its dynamics.

In this section, we use a dynamic model of autonomous quadrotor vehicles, specifically STARMAC presented in [8, 41] and Aerial Robot in [50]. With this model, the nonlinear dynamics of quadrotor vehicle are described in terms of the applied force $F_i \in \Re^3$ and moment $M_i \in \Re^3$ as follows:

$$F_i = m_i \ddot{p}_i \tag{2.16}$$

$$M_i = I_{b_i} \dot{\omega}_{B_i} + \omega_{B_i} \times I_{b_i} \omega_{B_i} \tag{2.17}$$

where m_i is the mass of the quadrotor, $I_{b_i} \in \Re^{3\times3}$ is the body fixed inertia matrix and $\omega_{B_i} \in \Re^3$ is angular velocity in the body frame. In (2.17), the superposition of the angular momentums of the rotors is assumed to be (near) zero, since the momentums from the counter rotating propellers cancel each other when yaw is held steady [41]. The free body diagram in Figure 2.8 shows the main forces and moments of the quadrotor. The motion and the rotation of quadrotors are provided by thrust vectors of its propellers. Every rotor is located at a distance l from the center of gravity. Each of $j = 1, 2, 3, 4$ rotor contributes thrust force T_{ij}, which is perpendicular to the relevant rotor plane; and produces a moment M_{ij} in the body fixed frame, which is a function of motor torque, thrust, and additional aerodynamic effects as shown in Figure Fig 2.8.

Roll and pitch are controlled via the torques generated by differential speeds of opposing pairs

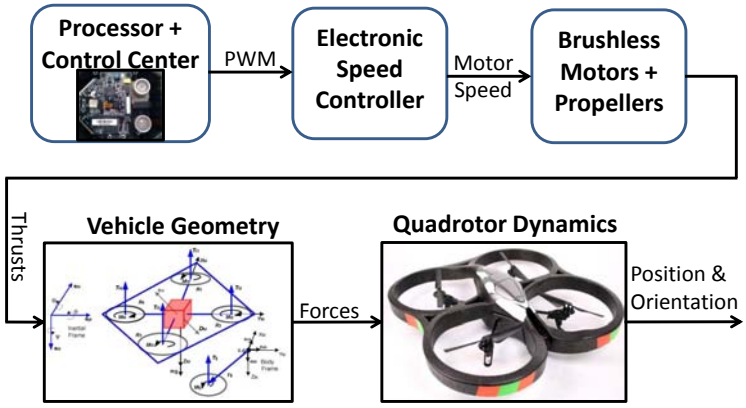


Figure 2.7: Experimental control diagram of Parrot AR Drone

of rotors. Differential speed between rotors 2 and 4 is used for roll action and differential speed between rotors 1 and 3 is used for pitch action. The yaw angle is controlled via the lateral torque generated by the difference between the superposition of the speeds of rotors 1 and 3, and the superposition of the speeds of rotors 2 and 4. D_{b_i} is the quadrotor body drag, and is a function of vehicle and wind speeds. Acceleration due to gravity is denoted g. The total force F_i is given by

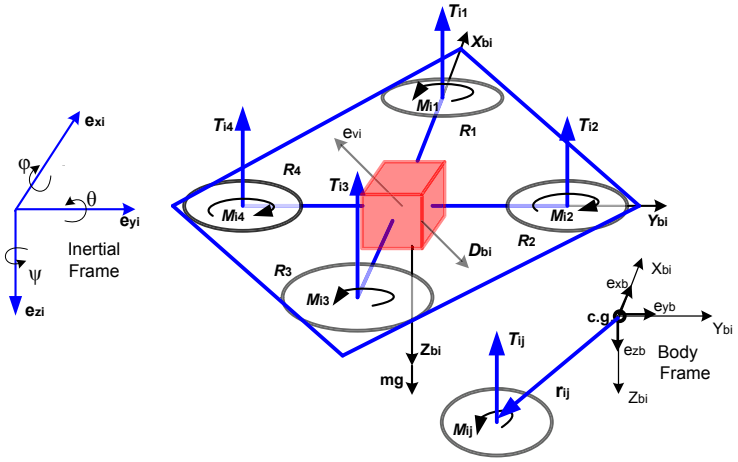


Figure 2.8: Free body diagram of quadrotor vehicle

$$F_i = -D_{b_i} e_{v_i} + mge_Z + \sum_{j=1}^{4}(-T_{ij} R^I_{r_{ij}} e_{z_b}) \tag{2.18}$$

where $R^I_{r_{ij}}$ is the rotation matrix from the plane of rotor j to inertial coordinates, e_{v_i} is the instant velocity direction in the earth fixed frame, e_Z denotes unit vector along the down direction in North-East-Down (NED) inertial coordinates, and e_{z_b} represents the unit vector along the z-axis of body fixed coordinates of quadrotor as it can be seen in Fig. 2.8. Similarly, the total moment, M_i, is given by:

$$M_i = \sum_{j=1}^{4}(M_{ij} + r_{ij} \times (-T_{ij} R^b_{r_{ij}} e_{z_b})) \tag{2.19}$$

where $R^b_{r_{ij}}$ is the rotation matrix from the plane of rotor j to body coordinates and r_{ij} is position vector of rotors with respect to center of gravity. Equations (2.16) and (2.19) form the nonlinear dynamics of quadrotor vehicles. An alternative simplified dynamic model is:

$$M_i \ddot{z}_i = u_{iz} - M_i g \tag{2.20}$$

$$M_i \ddot{y}_i = u_{iz} \sin\phi_i \ = u_{iz}\phi_i \tag{2.21}$$

$$M_i \ddot{x}_i = -u_{iz} \sin\theta \cong -u_{iz}\theta_i \tag{2.22}$$

$$u_{iz} = \frac{T_{i1} + T_{i2} + T_{i3} + T_{i4}}{4} \tag{2.23}$$

where ϕ_i and θ_i, respectively, are roll angle and pitch angle. u_{iz} represents a total thrust on the body along z-axis. $T_{i1}, \ldots, T_{i4}$ are thrust commands of each propeller. Next, we present the longitudinal modeling of parameter estimation of road vehicles for highway platooning.

2.4 Autonomous Road Vehicles

In this section, the details of electronic modification and modeling of road vehicle are provided for implementing longitudinal autonomous driving at highways or urban places.

2.4.1 Longitudinal Vehicle Dynamics

The vehicle and powertrain dynamics are the two main parts of longitudinal model of road vehicles. We first consider a vehicle moving on an inclined road to better analyze all affecting longitudinal forces. The external forces on Smart Fortwo vehicle illustrated in Fig. 2.9 include gravity, longitudinal aerodynamic drag force F_a, longitudinal tire forces F_{xf} and F_{xr} at the front and rear tires, rolling resistance forces R_{xr} and R_{xf} [68].

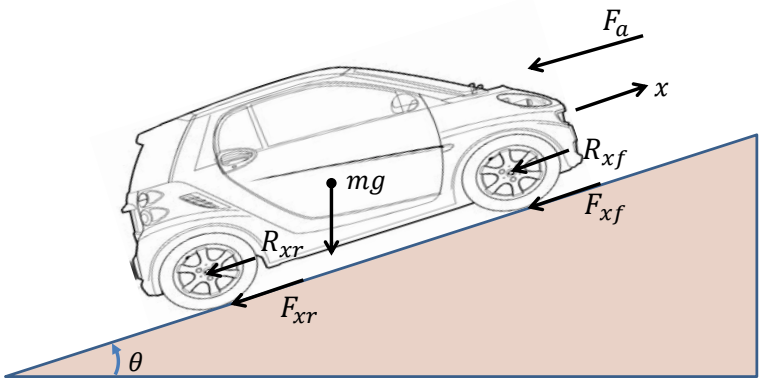


Figure 2.9: Longitudinal forces acting on a vehicle at inclined road

These forces are balanced on the vehicle as illustrated in 2.9, and it is represented as

$$F_{xf} + F_{xr} = m\ddot{x} + mg\sin\theta + F_a + R_{xr} + R_{xf}, \tag{2.24}$$

where m, g, θ are the vehicle mass, the gravitational acceleration, and the angle of road inclination, respectively. The aerodynamic force F_a depends on mass density of air, aerodynamic

drag coefficient, velocity of the vehicle and wind, and the projection of the vehicle's frontal area. The longitudinal tire force is affected by slip ratio of the tires, normal load on the tire, and friction coefficient. Since the road material is much stiffer than tires, the tires are deformed in the contact patch. This tire deformation results in energy loss during driving, which is called rolling resistance. Finally, the rotational velocity of the wheels are highly affected by the driveline dynamics of the vehicle. The major elements of driveline dynamics are torque converter, transmission dynamics, engine dynamics, and wheel dynamics [68]. The complex and nonlinear dynamics of road-vehicle are simplified for our string stability guaranteed control designs for platooning [46, 56–58].

2.4.2 Road Vehicle Longitudinal Autonomy

To control longitudinal driving is one of the challenging factor in mitigating traffic congestion at highways. So, we first start with two vehicles platoon scenario improving acceleration and deceleration profile at a single lane. Then, we extend our design employing multiple vehicles equipped with wi-fi and lidar units. Initially, the Fontys Automotive Research center provided two Smart Fortwo vehicles for testing non-cooperative and cooperative driving strategies. Electronic modification of these Smart cars were conducted by Fontys for smooth longitudinal driving under rapid acceleration/deceleration actions of the front vehicles. There are mainly four parts of our autonomous vehicle architecture for implementing advanced CACC algorithms: $i)$ front control panel, $ii)$ sensor suite, $iii)$ communication box, and $iv)$ embedded computer[2], as depicted in Fig. 2.10.

The front control panel is for switching between human pilot and autonomous control, and emergency cases. The autonomous mode takes the control of throttle, brake pedal, and gear shifting with the help of embedded computer and Controller Area Network Bus (CANBus). The components of the sensor suite are Ibeo Lidar for relative distance and relative velocity measurements, accelerometer and gyro for longitudinal acceleration and yaw rate, speedometer for vehicle speed, GPS receiver, and wireless transceiver. Moving average filter is used to enhance the accuracy of Lidar information. The stream of ego vehicle's sensory information and states is provided by CANBus interface, and the data transmission by other preceding vehicles is received by wireless communication unit (ComBox).

The vital tasks of the xPC Target computer are transmitting current vehicle states and characteristics to the ComBox via CANBus, controlling vehicle actuators, and controlling emergency unit considering safety requirements. Finally, a human driver is always responsible for steering

[2]We have only one xPC Target PC104 stack as embedded computer implementing autonomous throttle and braking.

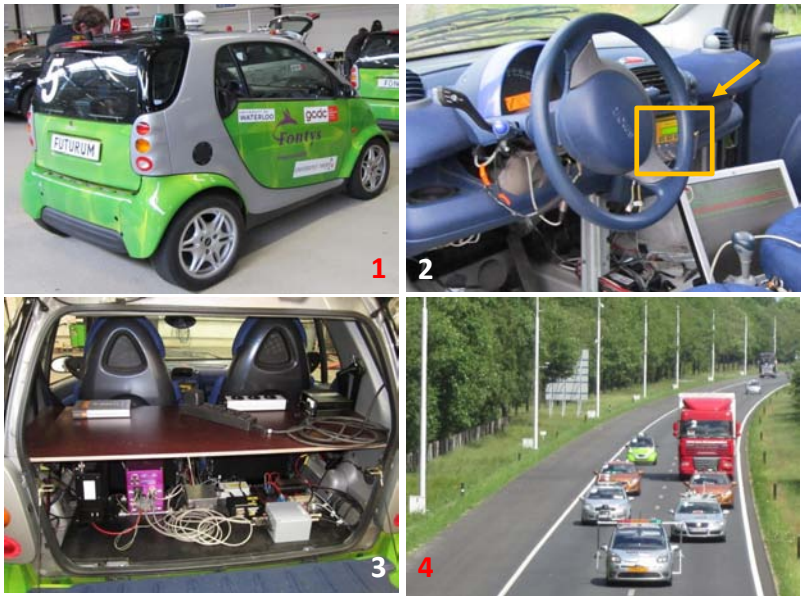


Figure 2.10: Smart vehicle for platooning purposes: 1) Smart Vehicle with wireless transceiver, GPS, and flashing lights 2) Front Control Panel 3) xPC Target, inertial sensors, communication box, and CanBus 4) GCDC heats and road side units

actions. Flashing lights at the top of the Smart vehicle are for warning other platoon drivers whether our current mode is autonomous or human driving.

2.4.3 Electronic Modification of Smart Fortwo Vehicles

For the autonomous control mode, a switch is used to enable the emergency unit and embedded computer controlled braking, throttle, and gearbox. We can also switch back with the same trigger switch so that the human driver take the longitudinal control again. Autonomous braking is controlled using the CANBus, but throttle and gearbox units are activated with analog voltage signals. Switching between manual and autonomous braking is controlled by the emergency unit [26].

Embedded Computer

The vehicle was equipped with a single xPC Target PC104 stack as a real-time embedded computer. The inputs to the PC104 are all sensory information and state (vehicle dimensions, po-

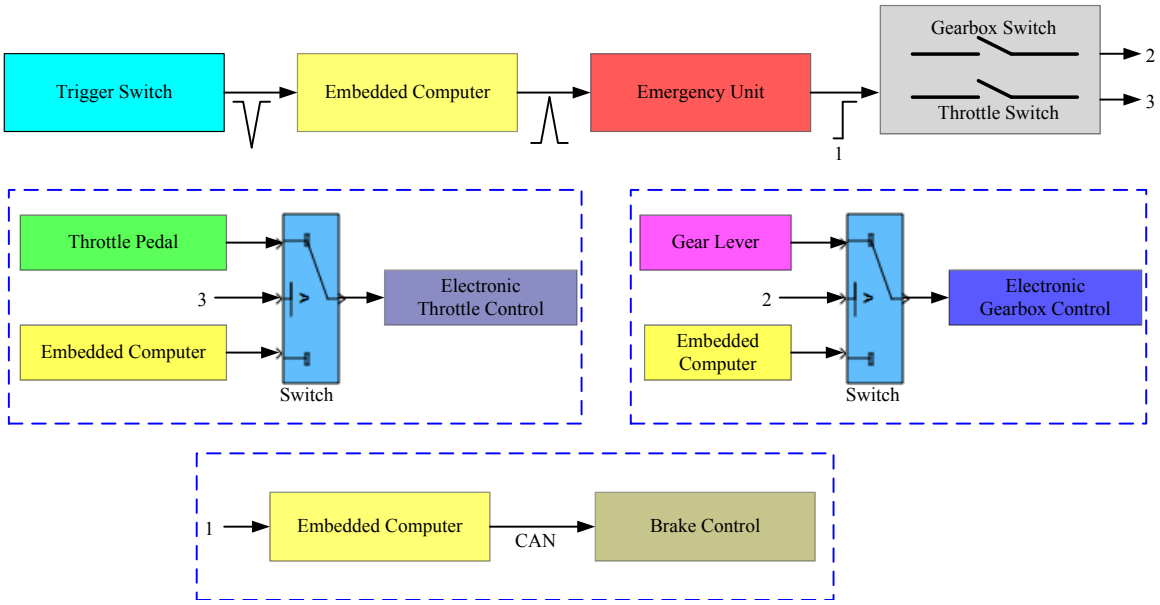


Figure 2.11: The design for autonomous control of throttle, brake pedal, and gear-shifting, and mechanism of switching to manual driving during platooning

sition, velocity, and acceleration) of the vehicle via CANBus interface, other vehicles' ID and state information from a wireless ComBox, autonomous or manual switching information from the dashboard switch, and output of the Emergency Unit. As a result, xPC Target Box is provided control signals for the vehicle actuators, ego vehicle's state data for transmission purposes, and control of the Emergency Unit. For designing executable control algorithms for the xPC Target computer, Matlab/Simulink was used.

Sensor Suite

An Ibeo LIDAR sensor was mounted in front of the vehicle for ACC implementation. This sensor scans a specified angle using laser beams with a certain frequency and detect objects with their relative position and relative velocity in its scanning range. While the lidar detects all objects within a certain angle in front of the vehicle, the control algorithm was enabled to concentrate on only one main object. In GCDC-based controller designs, the vehicle in front was considered as the most important. An accelerometer and a gyroscope were mounted in the vehicle for measuring longitudinal acceleration and yaw rate respectively. These additional signals were necessary for developing the object interception algorithm and transmitting current

acceleration to the other vehicles in the platoon. The vehicle's longitudinal acceleration was used as a reference for the actuators in the internal feedback control loop. A GPS Receiver receives the vehicle's GPS coordinates, which were broadcast to road side unit (RSU) and other vehicles in the platoon. These additional sensors and inboard vehicle sensors could be received and transmitted through the CANBus.

Autonomous Control of Actuators

For autonomous longitudinal controlling of the Smart Fortwo cars, we needed to have full control of the throttle, the brake and the gearbox of the vehicle. Braking of the car is carried out by an electro hydraulic brake (EHB) mechanism. The control algorithm calculated the brake pressure, which was transmitted to the EHB via the CAN interface. The EHB system generated the prescribed brake pressure for four wheels separately. The throttle position has been already controlled electronically in this road vehicle. We added additional electronic switches between the electronically controlled throttle pedal and throttle signal generator for autonomously controlling the throttle mechanism with the PC104 stack. Similar modification was used for the shifting mechanism of the gearbox. The vehicle was equipped with an automated manual transmission (AMT). Team Futurum developed a gearshift algorithm, which decided the timing of gear shifting and generated the desired gearshift signal for accelerating or decelerating the car.

Emergency Unit

The emergency unit was designed for the safety of the vehicle in the course of autonomous heats during GCDC and this was one of the requirements in order to participate. The embedded computer sent a watchdog signal to the emergency unit while the autonomous mode was activated. If this watchdog signal was active and the switch on the dashboard was triggered, the output of the emergency unit was turned high (ON). This signal and the position of sensors for braking, throttle, and gearbox control assigned the full control of the Smart to the embedded computer. If the driver pushed the emergency button on the dashboard, the output of the emergency unit switched to low, and full control of the Smart was returned to the driver. If the xPC Target Box failed, the watchdog signal would not be transmitted, and the driver had to take full control of the Smart. Any abrupt pressure on brake or throttle pedal whit the autonomous mode active caused full control of the vehicle to be returned to the human driver.

2.4.4 System Identification of Road Vehicles

In this section, we present analysis of acceleration/deceleration profile of selected group of vehicles, and longitudinal identification of three different vehicle parameters for platooning by means of Matlab and CarSim.

Longitudinal Parameter Identification

The realistic compact car, middle size passenger car, and sport utility vehicle (SUV) are selected for longitudinal modeling of vehicle dynamics and parameter identification. A-class hatchback compact model (*Audi A4*, *Mazda Millennia*, *Mazda Protege*, *BMW E 323i*, *Infinity G20*), D-class mid-size sedan (*Acura 3.2TL*, *Mercury Stable*, *Audi A6*, *A8*, *Mazda 626*), and E-class SUV (*BMW X5*, *Lexus LX 740*, *Mercedes Benz ML320*, *Subaru Forester*) are selected from CarSim library. Matlab/Simulink are used to analyze the vehicle acceleration/deceleration dynamics, and test actual throttle, gear shifting, and braking actions.

Analyzing the realistic response of the desired velocity profile in Fig. 2.12, we can deduce

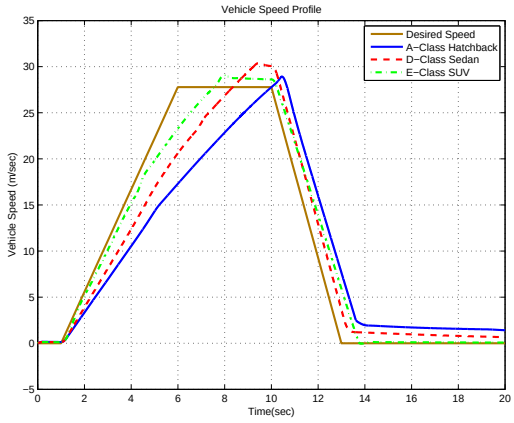


Figure 2.12: Longitudinal speed of compact, mid-size, and SUV vehicles

that the acceleration profile is not symmetric with the deceleration profile of these vehicles. But, we cannot directly define deceleration profile using acceleration coefficients. That is why, we obtain an acceleration/deceleration profile as depicted in Fig. 2.13, which can be also used to analyze the saturation of actuators during acceleration and deceleration period. To accommodate true model parameters, we specify a rapid acceleration/deceleration actions and a constant speed time span. Thus, Table 2.1 identifies relevant model parameters using transient and steady state

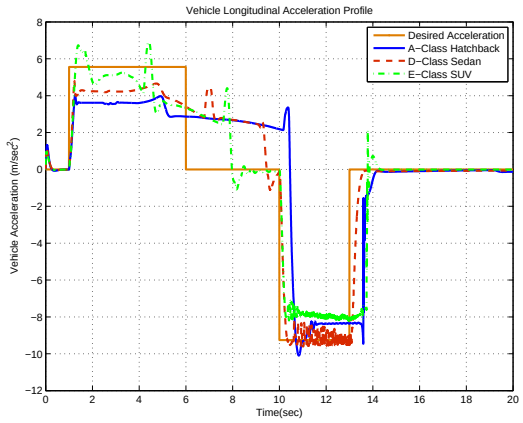


Figure 2.13: Longitudinal acceleration of compact, mid-size, and SUV car models

responses of Fig. 2.12 and Fig. 2.13. Additionally, saturation limits of acceleration/deceleration limits are calculated with the help of vehicle response to the rapid speed cycle. The maximum and minimum saturation limits are the reason of asymmetric longitudinal speed profile or fast braking, but slow accelerating of all vehicles. The K_i, τ_i, t_{di}, and f_{di} parameters in Table 2.1 represent the steady state gain, time constant of the actuator dynamics, the actuator delay, and desired force, respectively.

Finally, as it can be easily seen from Fig. 2.12, the longitudinal speed performance of different

Table 2.1: Linear Parameter Identification of Road Vehicles

System Parameters for Vehicle Models		
Compact Car (A-Class Hatchback)	Sat. Acc.: 3.6 m/s2, Sat. Dec.: -8.4 m/s2	
Acceleration	$t_{di} = 0.02$ sec, $\tau_i = 0.15$ sec, $K_i = 1$	Mazda Protege
Deceleration	$t_{di} = 0.43$ sec,$\tau_i = 0.17$ sec, $K_i = 1$	m=1151 kg
Mid-Size Car (D-Class Sedan)	Sat. Acc.: 4.2 m/s2, Sat. Dec.: -9.4 m/s2	
Acceleration	$t_{di} = 0.02$ sec, $\tau_i = 0.15$ sec, $K_i = 1$	Chevrolet Lumina
Deceleration	$t_{di} = 0.1$ sec, $\tau_i = 0.17$ sec, $K_i = 1$	m=1583 kg
SUV Vehicle (E-Class SUV)	Sat. Acc.: 6.4 m/s2, Sat. Dec.: -8.0 m/s2	
Acceleration	$t_{di} = 0.02$ sec, $\tau_i = 0.15$ sec, $K_i = 1$	Dodge Durango
Deceleration	$t_{di} = 0.06$ sec, $\tau_i = 0.17$ sec, $K_i = 1$	m=2201 kg

vehicle models are differ for the same reference speed profile and the acceleration and deceleration behaviors are not symmetric as it is shown in Fig. 2.14. The asymmetric behavior is caused

by fast braking, but slow accelerating of all type of road vehicles. After calculating the model parameters as in Table 2.1, we are able to test various cooperative highway platooning scenarios. In the following chapter, the distributed cohesive coordination algorithm and cooperative control strategies are given using various UAV models.

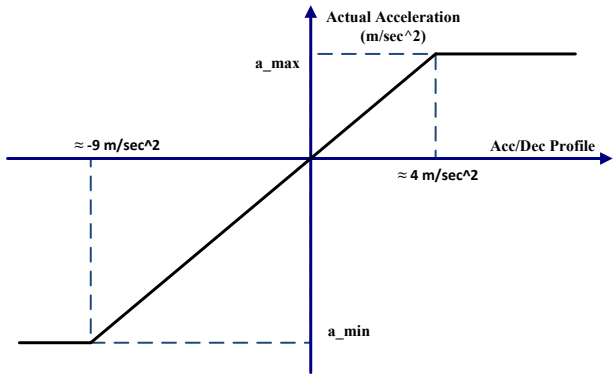


Figure 2.14: Saturation limits for acceleration/deceleration profile of road vehicles

2.5 Conclusion

In this chapter, the modeling of small scale fixed-wing UAVs, quadrotor vehicles, and road vehicles is described with some fundamental experimental design principles considering cohesive control and platooning missions. We start with the fixed-wing model UAV with a simplified motion characteristics. The lateral motion includes nonholonomic constraint due to roll and yaw actions. Additionally, the pitch actions and airspeed are included in longitudinal motion of the UAV. Therefore, the motion characteristics of fixed-wing UAV are reflected within lateral and longitudinal axes in three-dimensional space. Second, we provide the simplified motion characteristics of quadrotor vehicles in three-dimensional axis with independent rotations. The quadrotor rotations and relevant translational motion are generated by arranging thrust forces of propellers correspondingly. Moreover, quadrotors are capable of making hover flight but UAVs are not. Finally, the longitudinal motion model of road vehicles in one-dimensional space is given with second order dynamics and first order actuator delays in order to implement platooning tasks to increase string stability of the cooperative driving missions at urban and highway roads. Next chapters build high level control strategies and coordination algorithms using the models of aforementioned aerial and road platforms.

Chapter 3

Distributed Cohesive Motion Control of Flight Vehicle Formations

3.1 Introduction

Cooperation and coordination of dynamical systems has become a popular research area in recent years [28, 37, 38, 44, 61, 74, 75, 79]. In this respect, especially micro aerial vehicles are one of the most desired testbeds, which are used for traffic density detection, payload transportation, and different civilian and military cooperative missions [3, 44]. In the corresponding studies, the system of interest is a collection of autonomous vehicles, cooperating with each other to perform a collective task. A particular problem of interest for such systems is to keep the shape of the agent collection (swarm) constant in the course of moving the swarm. In other words, the goal is to move the swarm in a formation and make the agents in the swarm act as one unit while there is no physical connection between these agents.

In this chapter, we focus on distributed cohesive motion control of swarms of autonomous agents moving in three dimensions, where the main task is to maintain the formation geometry of the swarm defined by a set of inter-agent distances during arbitrary maneuvers. We design our base distributed control scheme assuming a single-velocity-integrator kinematics model for the agents and then extend this design to be applicable for more practical three-dimensional kinematics and dynamics of fixed-wing aerial and quadrotor vehicles [17, 18, 35, 54].

For better accommodation of the mentioned cohesive motion control problem, we simplify the swarm system using a point-agent system model and represent each multi-agent formation of interest by a directed graph where each agent is represented by a vertex. The control architecture defining the individual responsibilities of the agents is represented either explicitly or implicitly

by the edges, each of which corresponds to an agent pair, the distance between which is required to be kept constant. For robust maintenance of the formation geometry, we use a rigid graph theoretical framework; more specifically we require the graphs representing the control architecture to be rigid and constraint consistent, i.e. persistent [33, 89]. Section 3.2 presents a brief explanation of this framework as well as detailed modeling of the swarm formation and the corresponding control architecture. Furthermore, the three dimensional on-line trajectory algorithm is given in Section 3.3.

In addition to the notions of graph rigidity and persistence, we utilize techniques of virtual target tracking and smooth switching to develop our decentralized control scheme similar to [33,74,90], where the same cohesive motion control problem has been tackled for two-dimensional formations. In this work, the main contribution is revision and extension of the corresponding two-dimensional designs [33, 74] for three dimensional formations, considering various sensory disturbances and actuator time delays as well. As a result, the particular control designs presented in Sections 3.4 and 3.5 for quadrotor and fixed wing unmanned aerial vehicle formations are challenging for implementing cohesive coordination missions.

As the first case study, we consider formations of quadrotor vehicles, which have received grown interest related with control implementation, coordination and cooperation experiments in recent years because of the design simplicity, indoor and outdoor operation efficiency and low weight of these vehicles [16, 41, 54, 87]. The main idea of the control implementation is to maintain the formation geometry in the course of arbitrary maneuvers. As a second case study, fixed-wing UAVs are considered as the agents in our three-dimensional formations. The design of the low-level controllers in each case study provides the trajectory tracking of each fixed-wing UAV or quadrotor vehicle, and distributed coordination algorithm calculates and generates the desired positions (trajectory) of each fixed-wing UAV or quadrotor. The rest of this chapter includes simulations for the designed control schemes in Section 3.6 and conclusions, final remarks and some future research directions in Section 3.7.

3.2 Background on Cohesive Motion Control

Consider a swarm S with m agents (mobile robots or autonomous vehicles) $A_1, ..., A_m$ moving in $\Re^3$ (where $m \geq 4$). In *cohesive motion control*, we deal with the problem of moving S from a given original setting (location and orientation) to a desired final setting and maintaining a certain formation during the motion. Before giving the formal definition of the *cohesive motion control problem* in Section 3.2.3, we first present the general modeling of the autonomous formations in Section 3.2.1 and a brief introduction to the notions of rigidity and persistence in Section 3.2.2.

3.2.1 Modeling the Formation

We consider asymmetric formation control structures [33, 89]. In the asymmetric control structure, only one agent (in the corresponding agent pair) is responsible for each inter-agent distance keeping task. We represent the multi-agent swarm, which is required to maintain a certain formation using such an asymmetric control architecture, by a directed graph $G_F = (V_F, E_F)$, called the directed underlying graph of the swarm or the formation. A leader-follower structured example of such representation can be seen in Fig. 3.1. The directed underlying graph $G_F = (V_F, E_F)$ has a vertex set V_F and an edge set E_F where each vertex $i \in V_F$ corresponds to an agent A_i in S and each directed edge $\overrightarrow{(i,j)}$ from i to j represents the control and information link between neighboring agents A_i and A_j [33]; more specifically a directed edge implies that A_i can sense its distance from A_j and is responsible to keep a desired distance d_{ij} from A_j. In this case, we also say that A_i follows A_j, or A_i is a follower of A_j. In the sequel, we call the swarm of agents $S = \{A_1, A_2, ..., A_m\}$ together with the underlying graph $G_F = (V_F, E_F)$ and the desired distance set $D_F = \{d_{ij} \mid \overrightarrow{(i,j)} \in E_F\}$ a *formation* represented by $F = (S, G_F, D_F)$.

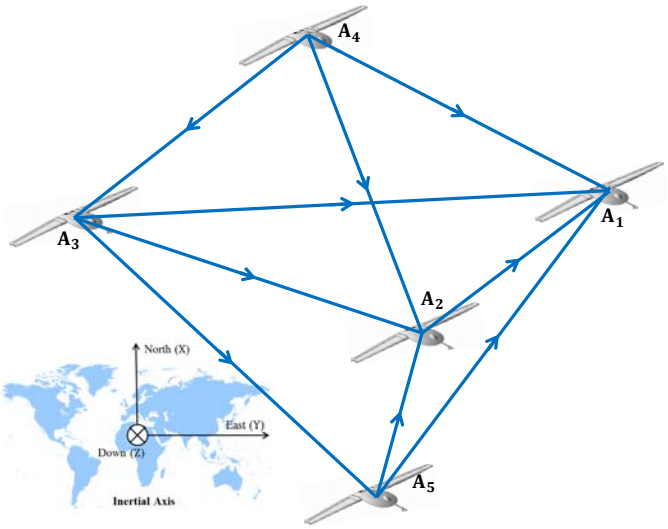


Figure 3.1: Directed underlying graph of a 3-dimensional persistent formation in leader-follower structure.

3.2.2 Rigid and Persistent Formations

A *rigid* formation is a formation in which the distance d_{ij} between each agent pair (A_i, A_j) remains constant, which implicitly means that the formation shape is maintained during any continuous motion of the formation.
In a formation, if each agent in the formation is always able to satisfy the distance constraints it is responsible for, then this formation is called *constraint-consistent*. A formation that is both rigid and constraint-consistent is called *persistent*. Furthermore, a persistent formation maintaining its persistence with the minimum possible number of links, which is $|E_f| = 3|V_f| - 6$ in three dimensions, is called *minimally persistent*. More formal definitions of *rigidity, constraint-consistence*, and *persistence* in three dimensions can be found in [89].
By the above convention, the number of directed edges that originate from a vertex in a directed underlying graph equals the number of distance constraints the corresponding agent must keep with other agents. In addition, an agent that has no directed edge originating from has 3 degrees of freedom (DOF) and can move freely in $\Re^3$. An agent without degree one has 2-DOF and can rotate around any axis passing through the agent it follows to fulfill its distance constraint. An agent without degree two has 1-DOF and can rotate around the axis joining the two agents it follows. Finally, an agent without degree 3 or more has 0-DOF and is called an *ordinary follower* which moves completely depending on the agents it follows. In a *3-dimensional minimally persistent formation*, the sum of $DOFs$ of individual agents is always 6 [89]. Based on the distribution of these 6 $DOFs$ among non-0-DOF agents, the formation may have various graphical structures. In this section, we consider only the leader-follower structure, which is the most convenient structure for to well-define distributed control schemes and the corresponding leading-following hierarchies. In this structure there is one 3-DOF agent which is called the *leader* (A_1), one 2-DOF agent which is called the *first-follower* (A_2), one 1-DOF agent which is called the *second-follower* (A_3) and some 0-DOF agents which are called *ordinary followers* (A_4, A_5), as illustrated in Fig. 3.1.

3.2.3 The Cohesive Motion Control Problem

The following assumptions are made in the definition of the cohesive motion control problem and implementation of the control task:

A1: Each agent A_i in the formation can perfectly measure its own (center of mass) position $p_i(t) = [x_i(t), y_i(t), z_i(t)]^T$ as well as the (relative) positions of the agents it follows (is required to maintain a distance with) at any time t.

A2: The distance-sensing range for a neighbor agent pair (A_i, A_j) is sufficiently larger than the desired distance d_{ij} to be maintained.

A3: The leader, first-follower and second-follower agents know their final desired positions p_{if}.

The cohesive motion control problem of interest can be formally stated as follows:
Problem 1: Consider a swarm S with m agents $A_1, ..., A_m$ ($m \geq 4$) that are initial at positions, respectively, $p_{10}, ..., p_{m0} \in \Re^3$. Let $F = (S, G_F, D_F)$ be a minimally persistent formation in leader follower structure, where D_F is consistent with the initial positions p_{i0}. The control task is, under Assumptions A1$-$A2, to move S to a final desired setting defined by a set of desired final positions, $p_{1f}, ..., p_{mf}$, which is consistent with D_F, without deforming the shape of the formation, i.e. forcing the inter-agent distances track the values defined in D_F, during motion.
Problem 1 is approached by a two-level control scheme: At the high level, a trajectory generator is designed in Section 3.3. At the low level, individual motion controllers are designed in Sections 3.4 and 3.5 for the aerial vehicle agents to track the trajectories generated at high level.

3.3 On-Line Trajectory Generation

In the scenario of *Problem 1*, there are more than four agents and that have to maintain the inter-agent distances while generating trajectories. For this purpose, first three agents are assigned as main agents: leader, first-follower, and second-follower. The trajectory for each agent A_i is defined as the desired relative position:

$$p_{id}^{(i)}(t) = [x_{id}^{(i)}(t),\ y_{id}^{(i)}(t),\ z_{id}^{(i)}(t)]^T = p_{id}(t) - p_i(t) \tag{3.1}$$

where the desired global position $p_{id}(t)$ is defined based on the order of A_i in the leader-follower formation structure as follows:

$$\begin{aligned}
p_{1d}(t) &= p_{1f} && (3.2)\\
p_{2d}(t) &= p_1(t) + p_{21},\ p_{21} = p_{2f} - p_{1f} && (3.3)\\
p_{3d}(t) &= p_1(t) + p_{31},\ p_{31} = p_{3f} - p_{1f} && (3.4)\\
p_{id}(t) &= \bar{p}_{jkl}(t, p_i(t)), \forall i \in 4, ..., m && (3.5)
\end{aligned}$$

where j, k, l are the indices of agents that i follows. $\bar{p}_{jkl}(t)$ denotes the intersection point of the spheres $S(p_j, d_{ij})$, $S(p_k, d_{ik})$ and $S(p_l, d_{il})$ that is closer to p_0. In the notion $S(.,.)$, the first argument indicates the center of sphere and the second argument indicates the radius of sphere. Detailed analytical derivation of $\bar{p}_{jkl}(t)$ is given in the Appendix B [4, 8].

3.4 Distributed Control Design for Quadrotor Formations

In this section, we design low level controllers for the case study with quadrotor agents. The simplified modeling of quadrotor dynamics is described in Section 2.3. Here, we derive suitable control laws for trajectory tracking, which are used to track the trajectories generated by (3.1)−(3.5). The vehicle kinematics are also applicable to quadrotor vehicles.

3.4.1 Trajectory Tracking Control

In this section, we present a set of attitude and altitude control laws for single quadrotor vehicle adapted from [41]. Attitude control loop provides the rapid adaptation to the varying commands and stabilize the attitude of the quadrotor. Altitude and attitude control loops use the vehicle's thrust to produce the desired vertical and lateral accelerations.

Attitude and Altitude Control

At small attitude angles (roughly $\pm 30^{\circ}$) and low angular speeds, the equations of motion can be decoupled about each attitude axis. Hence the effective control inputs $u_{i\phi}$, $u_{i\theta}$ and $u_{i\psi}$ about each axis (roll, pitch, yaw) can be considered to be applied independently [41, 50]. The effective control inputs for each axis can then be combined to generate the corresponding thrust commands $T_{i1}, \ldots, T_{i4}$ that need to be applied to, respectively, rotors $1, \ldots, 4$:

$$\begin{aligned} T_{i1} &= -u_{i\theta} + u_{i\psi} + u_{iz} \\ T_{i2} &= u_{i\phi} - u_{i\psi} + u_{iz} \\ T_{i3} &= u_{i\theta} + u_{i\psi} + u_{iz} \\ T_{i4} &= -u_{i\phi} - u_{i\psi} + u_{iz} \end{aligned} \tag{3.6}$$

These thrust commands are the control inputs of the quadrotor vehicle's attitude and trajectory tracking control system. The time delay in thrust can be approximated as a first order delay and has to be considered in controlling each attitude angle. The linearized transfer function for the roll axis is [41]:

$$\frac{\Phi_i(s)}{U_{i\phi}(s)} = \frac{I_{i\phi}/l}{s^2(\tau s + 1)} \tag{3.7}$$

where $I_{i\phi}$ is the component of the inertia matrix I_{b_i} for the roll axis, and l is the distance of each rotor to the center of gravity, and τ is the thrust time delay. The transfer functions for pitch and yaw axes can be obtained in a similar way.

We design the following tracking controller based on proportional-derivative-double derivative (PDD) and pole placement designs with using root-locus techniques:

$$U_{i\phi}(s) = C_i(s)(\Phi_i^d(s) - \Phi_i(s)) \tag{3.8}$$

$$C_i(s) = (\frac{k_{dd_i}s^2}{(1+\frac{s}{P_c})^2} + \frac{k_{d_i}s}{(1+\frac{s}{P_c})} + k_{p_i})\bar{C}_i(s) \tag{3.9}$$

where $\Phi_i^d(s)$, k_{dd_i}, and P_c are desired roll angle, double derivative, and filtering term, respectively. P_c determines the pole locations of the filter in derivative and double-derivative actions. The coefficients of $C_i(s)$ are tuned applying root-locus graphics tools on the simplified model presented in Section 2.3 and the transfer function (3.7) to provide faster and more accurate performance. The corresponding control laws for all angles ϕ, θ and ψ (angles in body axes) are obtained as follows:

$$u_{i\phi} = C_i(s)(\phi_i^d - \phi_i) \tag{3.10}$$

$$u_{i\theta} = C_i(s)(\theta_i^d - \theta_i) \tag{3.11}$$

$$u_{i\psi} = C_i(s)(\psi_i^d - \psi_i) \tag{3.12}$$

For the implementation of the effective controllers, rate gyros are used as measurement units, and then the measured data are filtered with the help of standard filtering techniques. At each time t, $T_{i1}(t), \ldots, T_{i4}(t)$ are obtained processing $u_{i\phi}(t), u_{i\theta}(t), u_{i\psi}(t)$.
In a similar way, the altitude controller is derived for the stabilization of the quadrotor altitude during tracking actions. The accelerations are obtained with accelerometers and feedback linearization is used to compensate for the gravity offset and thrust deflection. The tolerance of the controller is limited with 30° according to plant model. The following proportional-integral-derivative (PID) control law is proposed for altitude tracking:

$$u_{iz} = \bar{k}_{d_i}(\dot{z}_i^d - \dot{z}_i) + \bar{k}_{p_i}(z_i^d - z_i) + \bar{k}_{I_i}\int_0^t (z_i^d - z_i)dt + T_{0_i} \tag{3.13}$$

where z_i is the altitude, z_i^d is reference altitude and T_{0_i} is nominal thrust. The quadrotor altitude dynamics is modeled by a second order transfer function that is obtained by linearization of (2.20) together with (2.19), (3.6).

Reference Angle Generator

In this subsection, we present a scheme to generate reference angles $\phi_i^d,\ \theta_i^d,\ \psi_i^d$ (for attitude and altitude controllers) from desired x_{id} and y_{id} positions which is generated by distributed coordination and sphere intersection algorithms as described in Section 3.3 and Appendix B.

The general process for reference angle generation can be easily seen from the left hand-side of block diagram in Fig. 3.2. The generation of ϕ_i^d, θ_i^d and ψ_i^d is implemented via a PID control design based on simplified dynamic equations (2.20)-(2.22).
The corresponding PID trajectory generators for the attitude angles are as follows:

$$U_{r_i}(s) = k_{p_i}^r + \frac{k_{I_i}^r}{s} + k_{d_i}^r s \tag{3.14}$$

$$\phi_i^d = U_{r_i}(s)(y_{d_i} - y_i) = U_{r_i}(s) y_{id}^{(i)} \tag{3.15}$$

$$\theta_i^d = U_{r_i}(s)(x_{d_i} - x_i) = U_{r_i}(s) x_{id}^{(i)} \tag{3.16}$$

$$\psi_i^d = 0 \tag{3.17}$$

Distributed algorithm in Section 3.3 provides the desired positions x_{d_i}, y_{d_i} and z_{d_i} for the quadrotor vehicles in the formation. Then, the equations (2.20)−(2.22), (3.13) and (3.14) ensure the generation of reference attitude angles ϕ_i^d, θ_i^d, ψ_i^d for the quadrotor vehicles as it can be seen in equations (3.15)−(3.17) [29].

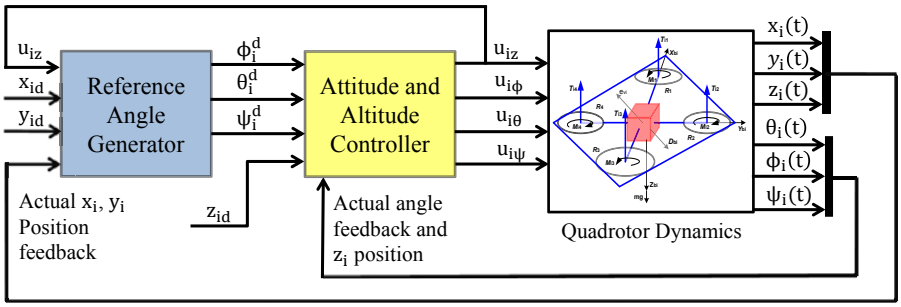


Figure 3.2: General control scheme of single quadrotor vehicle

3.4.2 Summary of Individual Controllers

As stated in Assumption A1, each quadrotor agent A_i knows its current position $p_i(t)$; and its desired position $p_{id}(t)$ is generated online using (3.2)−(3.5). At each time t, A_i takes its current position $p_i(t)$ as the last visited waypoint p_{ik}^d and the desired position $p_{id}(t)$ as the next waypoint $p_{i(k+1)}^d$. Based on this setting, the individual controller of A_i is composed of (3.6), (3.10), (3.11), (3.13) and (3.14), where z_i^d is taken as the z-component of $p_{id}(t)$ at each time instant t, and $u_{i\psi}$ is set to zero.
The final distributed control structure for single quadrotor takes into account (3.9). When we

consider the linearized model equation of quadrotor as in (3.7) and PDD based control model in (3.9), it can be easily seen that the closed system is always stabilized with a zero steady state error. A generic proof of the asymptotic stability of PID type controllers utilizing Lyapunov based event triggering scheme can be found in [86]. Tuning of quadrotor control parameters is achieved using root locus for finding a stable and desired step response similar to [27]. A summary of theoretical tracking and error performance analysis of PID based quadrotor controllers is provided in [27, 86].

3.5 Distributed Control Design for Fixed-Wing UAV Formations

As the second case study after quadrotor vehicles, we deal with the formations of unmanned aerial vehicles (UAVs). These small fixed wing UAVs employed in lots of different military and civilian operations. After the implementation of realistic quadrotor models in distributed cohesive motion control of autonomous formations in Section 3.4, UAV models are tested with the help of 5-agent three dimensional formation scenario as it is depicted in simulations. The simplified single UAV model is first described and then control laws for path tracking are derived for the trajectory tracking of the UAV. These control loops can be used in cohesive motion control design for UAV formations as described in Section 3.2.3 [4, 6]. In Section 3.5.2, we revise our control design considering transformation processes between body-frame-fixed commands and earth-frame-fixed system outputs.

3.5.1 Base Distributed Control Design

In this subsection, we model each agent A_i as a point agent, ignoring its geometric properties for the sake of simplicity as described in Section 3.2. We present our base distributed control design considering the following single-velocity-integrator kinematic models for the agents:

$$\dot{p_i}(t) = \mathrm{v}_i(t) \tag{3.18}$$

$$\|v_i(t)\| = \bar{\mathrm{v}}_i \tag{3.19}$$

where $p_i(t) = [x_i(t), y_i(t), z_i(t)]^T$ is the position and the vector $\mathrm{v}_i(t) = [\mathrm{v}_{x_i}(t), \mathrm{v}_{y_i}(t), \mathrm{v}_{z_i}(t)]^T \in \Re^3$ denote the velocity of the center of mass of the agent A_i at time t.
Considering Problem 1 and the control architecture of minimally persistent formations in leader-follower structure, the individual controller of each agent needs to be designed separately according to the DOF of that agent. As a result, we need four different types of individual controllers.

Next, based on the schemes developed in [33,74] for 2-dimensional formations, we propose a distributed control scheme for 3-dimensional cohesive motion. Noting that the schemes in [33, 74] led to a complicated control law for an agent with 1-DOF in 2 dimensions to switch between translational and rotational actions. In our design, we use another approach which uses the relative positions between two agents.
Let the agents be labeled according to leading-following hierarchy: A_1 is the leader (and therefore, A_1 does not have any constraint to satisfy.), A_2 is the first follower that follows the leader, A_3 is the second follower that follows A_1 and A_2; each A_i for $i = 4, ..., m$ is an ordinary follower following three agents A_j, A_k, A_l where $j, k, l < i$. Now, we propose the following control law for each agent A_i :

$$\mathrm{v}_i(t) = \bar{\mathrm{v}}\sigma_i(t)p_{id}^{(i)}(t)/\|p_{id}^{(i)}(t)\| \tag{3.20}$$

$$\sigma_i = \begin{cases} 0, & \|p_{id}(t)\| < \varepsilon_k \\ \frac{\|p_{id}(t)\|-\varepsilon_k}{\varepsilon_k}, & \varepsilon_k \leq \|p_{id}(t)\| < 2\varepsilon_k \\ 1, & \|p_{id}(t)\| \geq 2\varepsilon_k \end{cases} \tag{3.21}$$

where $\varepsilon_k > 0$ is small design constant; $\sigma_i(t)$ is to avoid chattering due to small but acceptable errors in the final position of A_i; $\bar{\mathrm{v}}_i$ is the constant maximum speed.
Since the velocity vector $\mathrm{v}_i(t)$ cannot be directly applied as a control signal, we consider (3.20) as a control law to generate the desired value v_{id} of v_i, i.e.,

$$\mathrm{v}_{id} = \bar{\mathrm{v}}\sigma_i(t)p_{id}^{(i)}(t)/\|p_{id}^{(i)}(t)\| \tag{3.22}$$

In the following subsections, the kinematical and dynamical modeling of practical autonomous systems, and distributed control scheme with considering kinematical modeling and low-level motion control of UAVs, are given for the maintenance of geometric formation of the swarm.

3.5.2 Low Level Control Design for Generic Fixed-Wing UAVs

This subsection is on the design of low level control laws for generic fixed-wing UAV agents. Since the flight dynamics is vehicle and environment specific and typically involves an extensive set of equations and database, we focus on kinematic models and develop a control law that generates the translational speed, angle-of-attack, and side-slip angles of the agent.

Kinematic Modeling

The vector $p_i = [x_i, y_i, z_i]^T$ is used to describe the position of the unmanned aerial agent A_i in three-dimensional earth fixed frame, where x-axis points north direction, y-axis points east

direction and z-axis indicates downward. The unit x, y and z coordinate vectors in this frame are denoted as e_x, e_y and e_z respectively. In the same inertial frame, the attitude of the single unmanned aerial vehicle is described with the vector $\bar{p}_i = [\phi_i, \theta_i, \psi_i]^T$, where the angles ϕ_i, θ_i and ψ_i angles are the vehicle's roll, pitch, and yaw angles, which represent the rotations with respect to x,y and z axes of inertial frame, respectively [49, 54].
The linear velocities and angular rates of quadrotor vehicle are introduced with the following notation in a body fixed frame:

$$\begin{aligned} V_i^b &= [u_i, v_i, w_i]^T \\ Q_i^b &= [\mathrm{p}_i, \mathrm{q}_i, \mathrm{r}_i]^T \end{aligned} \tag{3.23}$$

where u, v and w are linear velocities expressed in body frame x,y and z-axis, respectively. Additionally, $\mathrm{p}_i, \mathrm{q}_i$ and r_i are angular rates in roll, pitch and yaw axis, respectively. Kinematics of A_i which is widely used in aircraft, spacecraft, and submarine motion control, involving (commands of) angle-of-attack, side-slip angle and speed directly instead of 3-dimensional velocity vector v_i which cannot be directly applied:

$$\dot{p}_i = \mathrm{v}_i = R_{b_1}^I(\bar{p}_i) V_i^b \tag{3.24}$$

$$\dot{\bar{p}}_i = R_{b_2}^I(\bar{p}_i) Q_i^b \tag{3.25}$$

$$u_i(t) = \bar{V}_i^b(t)\cos(\alpha(t))\cos(\beta(t)) \tag{3.26}$$

$$v_i(t) = \bar{V}_i^b(t)\cos(\alpha(t))\sin(\beta(t)) \tag{3.27}$$

$$w_i(t) = \bar{V}_i^b(t)\sin(\alpha(t)) \tag{3.28}$$

$$R_{b_1}^I(\bar{p}_i) = \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi c\phi s\theta \\ s\psi c\theta & c\psi c\phi + s\theta s\phi s\psi & -c\psi s\phi + s\theta s\psi c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \tag{3.29}$$

$$R_{b_2}^I(\bar{p}_i) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix} \tag{3.30}$$

where $\bar{V}_i^b = \|V_i^b\|$, α, β denote, respectively, the velocity vector of the agent A_i in body axes, the translational speed of A_i, angle-of-attack and side-slip angle, as depicted partially in Fig. 3.3. $s\cdot$, $c\cdot$ and $t\cdot$ denote the sine, cosine and tangent of the Euler angles. Moreover, $R_{b_1}^I$ fulfills the transformation of the position of the aerial vehicle from body axis to inertial axis and $R_{b_2}^I$ is used for the angular conversion of the vehicle as in (3.25).

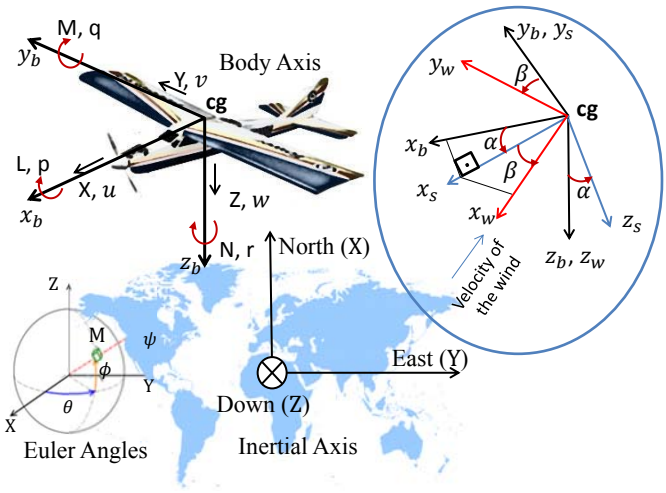


Figure 3.3: Modeling of the three dimensional motion of an UAV and conversion between body, stability, and wind-axis.

Remark 2 *All the kinematic equations and transformations in this subsection are applicable for quadrotor vehicles and unmanned underwater vehicles as well [17, 30, 35].*

The complete rotation matrix $R_{b_1}^I$ for three-dimensional space is obtained using the *zyx-convention*:

$$R_{b_1}^I := R_{z,\psi} R_{y,\theta} R_{x,\phi} \tag{3.31}$$

where the angles $\phi,\ \theta,\ \psi$ represent the vehicle's roll, pitch, and yaw angles, respectively. (3.31) can be written in explicit form as in (3.29).
In general, researchers benefit from strapdown inertial navigation system (INS) for attitude determination in multi-vehicle coordination scenarios. In such scenarios, gyroscopes are used for measuring body angular rates and accelerometers for measuring linear accelerations. In a distributed control design based on (3.24) and (3.26)−(3.28), $u_i(t), v_i(t)$ and $w_i(t)$ can be considered as control inputs of the individual kinematic system of agent A_i. These inputs depend on $\bar{V}_i^b(t), \alpha(t)$ and $\beta(t)$, which can be measured easily with the help of a sensor suite over the agent. In addition, these variables can be adjusted with the help of throttle and other control surfaces of the agents and the transition dynamics between the control surfaces and $\bar{V}_i^b(t), \alpha(t), \beta(t)$ values are stable and rapid for conventional type of unmanned and underwater vehicles [59]. Depending on this fact, we ignore these transition dynamics, i.e., we consider them as unity dynamics.
The Euler angles with respect to inertial frame and the elements of the coordinate transformation matrix can be derived using the algorithm as described in [39]. In the first step of this procedure,

the output of strapdown inertial measurement (IMU) unit is converted to rotation information with calibration and compensation. Then, this rotation vector is converted to a quaternion vector. Quaternion vector notation is used for preventing singularities in matrix operations during movement of realistic unmanned or underwater swarms. After these calculations, quaternions are converted to Euler rotation angles ψ, θ and ϕ, which are necessary for the derivation of coordinate transformation matrix $R^I_{b_1}(\bar{p}_i)$. Thus, we can obtain the transformation matrix from vehicle roll-pitch-yaw (x_b, y_b, z_b) coordinates to the global north-east-down (X, Y, Z) coordinates.

Control Design

The agent model (3.24), (3.26)−(3.28) can be rewritten as

$$\dot{p}_i = \mathrm{v_i} = \mathrm{R^I_{b_1}(\bar{p}_i)V^b_i} = \mathrm{R^I_{b_1}(\bar{p}_i)} \begin{bmatrix} c\alpha(t) \cdot c\beta(t) \\ c\alpha(t) \cdot s\beta(t) \\ s\alpha(t) \end{bmatrix} \mathrm{\bar{V}^b_i(t)} \tag{3.32}$$

Based on (3.32), the control law (3.20) can be implemented as follows:

$$V^b_i = \begin{bmatrix} u_i \; v_i \; w_i \end{bmatrix}^T = (R^I_{b_1}(\bar{p}_i)^{-1} \mathrm{v_{id}}) \tag{3.33}$$

$$\mathrm{v_{id}(t)} = \bar{\mathrm{v}}\sigma_i(t) p^{(i)}_{id}(t) / \| p^{(i)}_{id}(t) \| \tag{3.34}$$

The velocity vector V^b_i in body-axis system is derived by multiplying the inverse of transformation matrix $R^I_{b_1}(\bar{p}_i)$ with V_{id} as in (3.33).
The velocity vector V^b_i is the control input of our distributed controllers. For the optimal calculation of our control inputs, we need the instant measurements of Euler angles, which are derived with the help of inertial sensors (gyros and accelerometers). In addition, quaternion conversion is used for preventing singularity problems during matrix operations due to values of tangential terms in angular transformation matrices during agile maneuvers. Gyro and accelerometer outputs and some mathematical manipulations are used for calculation of global frame rotation angles momentarily with noise effects. Furthermore, the implementation of calculated velocity vector to the system is transmitted with a time delay due to dynamics of servo actuators. These noise and time-delay effects are modeled in our simulated scenario with multiplicative random signals and transport-delay blocks. The actual positions of agents are calculated continuously in (3.32) under the consideration of accuracy errors, mechanical effects, and with the contribution of agent kinematics as illustrated in simulation outputs of Section 3.4 and 3.5.
In order to mitigate the effect of sensory disturbances and time delays, (3.33) is remodified using a PI controller form, as follows:

$$V^b_i = (K_p + \frac{K_i}{s})(R^I_{b_1}(\bar{p}_i))^{-1} \mathrm{v_{id}} \tag{3.35}$$

The use of the PI controller (3.35) is demonstrated in Section 3.5.3.

3.5.3 Low Level Control Design for Piccolo-Controlled Fixed-Wing UAV Agents

Dynamic Modeling

As it is indicated in Remark 2, kinematic equations and transformations are applicable to both Quadrotor vehicles and fixed-wing UAVs. Therefore, the main focus of this part is to simplify the lateral-longitudinal model of the UAV. For the fixed-wing UAV dynamics model, we adopt the one for the Piccolo-controlled UAVs given in [6, 12]:

$$\begin{aligned} \dot{x}_i &= s_{lat,i} cos\theta_i \\ \dot{y}_i &= s_{lat,i} sin\theta_i \\ \dot{s}_{lat,i} &= \frac{1}{\tau_s}\left(-s_{lat,i} + s_{lat,i_{cmd}}\right), \underline{s} \leq \dot{s}_{lat,i} \leq \overline{s} \end{aligned} \tag{3.36}$$

$$\begin{aligned} \dot{\theta}_i &= \omega_i \\ \dot{\omega}_i &= \frac{1}{\tau_\omega}\left(-\omega_i + \omega_{i_{cmd}}\right) \end{aligned} \tag{3.37}$$

$$\dot{z}_i = \frac{1}{\tau_z}\left(-z_i + z_{i_{cmd}}\right), \underline{z} \leq \dot{z}_i \leq \overline{z} \tag{3.38}$$

where $p_i(t) = [x_i(t), y_i(t), z_i(t)]^T \in R^3$, $s_{lat,i}$, θ_i and ω_i denote the position in three-dimensional earth fixed frame, lateral speed, lateral heading and lateral turn rate of the UAV agent A_i, respectively. The angle θ_i is defined within $-\pi < \theta_i \leq \pi$. $s_{lat,i_{cmd}}$, $\omega_{i_{cmd}}$ and $z_{i_{cmd}}$ are the control signals. These three external inputs are generated with the help of inner control loop applied on mechanical actuators such as elevators, ailerons and rudders.

Control Design

Having established the high level control scheme generating the desired positions for the UAV agents on-line in Section 3.3 and 3.5.1, we present the low level individual UAV motion controller design to generate the control signals to track the desired positions generated by the high level formation control scheme in this section.
Let us consider the following partitioning of the desired velocity $\mathrm{v_{id}(t)}$ in (3.20):

$$\mathrm{v_{id}} = \left[\ \mathrm{v_{lat,id}}\ \mathrm{v_{z,id}}\ \right]^T \in \Re^3 \tag{3.39}$$

$$\mathrm{v_{lat,id}} = \left[\ \mathrm{v_{x,id}}\ \mathrm{v_{y,id}}\ \right]^T \in \Re^2 \tag{3.40}$$

Consider the Lyapunov function

$$P_{\mathrm{v_i}}(t) = \frac{1}{2} e_{\mathrm{v_i}}^T(t) e_{\mathrm{v_i}}(t) \tag{3.41}$$

$$e_{\mathrm{v_i}}(t) = \dot{p}_i(t) - \mathrm{v_{id}(t)} \tag{3.42}$$

We approach design of the low level motion controllers to generate the command signals $s_{lat,i_{cmd}}$, $\omega_{i_{cmd}}, z_{i_{cmd}}$ based on forcing $P_{\mathrm{v_i}}$ in (3.41) to decay to zero. From (3.36)−(3.38), (3.42) we have

$$\begin{aligned} e_{\mathrm{v_i}} &= \begin{bmatrix} s_{lat,i}\cos\theta_i - \mathrm{v_{x,id}} \\ s_{lat,i}\sin\theta_i - \mathrm{v_{y,id}} \\ \frac{1}{\tau_z}(-z_i + z_{i_{cmd}}) - \mathrm{v_{z,id}} \end{bmatrix} \\ &= \begin{bmatrix} s_{lat,i} e_{\theta_i} - \mathrm{v_{lat,id}} \\ \frac{1}{\tau_z}(-z_i + z_{i_{cmd}}) - \mathrm{v_{z,id}} \end{bmatrix} \end{aligned} \tag{3.43}$$

where for any angle α, e_α is defined as $e_\alpha = [\cos\alpha, \sin\alpha]^T$. Selecting

$$z_{i_{cmd}} = z_i + \tau_z \mathrm{v_{z,id}} \tag{3.44}$$

(3.41) reduces to

$$P_{\mathrm{v_i}}(t) = \frac{1}{2}\| s_{lat,i} e_{\theta_i} - \mathrm{v_{lat,id}} \|^2 \tag{3.45}$$

(3.36) and (3.45), together with the fact

$$\dot{e}_{\theta_i} = e_{\theta_i}^{\perp}\omega_i, \quad e_{\theta_i}^{\perp} \triangleq [-\sin\theta_i, \cos\theta_i]^T$$

leads to

$$\begin{aligned} \dot{P}_{\mathrm{v_i}} &= (s_{lat,i} e_{\theta_i} - \mathrm{v_{lat,id}})^{\mathrm{T}} (\dot{\mathrm{s}}_{\mathrm{lat,i}} \mathrm{e}_{\theta_i} + \mathrm{s_{lat,i}} \dot{\mathrm{e}}_{\theta_i} - \dot{\mathrm{v}}_{\mathrm{lat,id}}) \\ &= (s_{lat,i} e_{\theta_i} - \mathrm{v_{lat,id}})^{\mathrm{T}} \left(-\frac{1}{\tau_s}(\mathrm{s_{lat,i}} \mathrm{e}_{\theta_i} - \mathrm{v_{lat,id}}) \right. \\ &\quad \left. -\frac{\mathrm{v_{lat,id}}}{\tau_s} + \frac{s_{lat,i_{cmd}} e_{\theta_i}}{\tau_s} + \omega_i s_{lat,i} e_{\theta_i}^{\perp} - \dot{\mathrm{v}}_{lat,id} \right) \end{aligned}$$

which can be written in the form

$$\begin{aligned} \dot{P}_{\mathrm{v_i}} &= -\frac{2}{\tau_s} P_{\mathrm{v_i}} + (s_{lat,i} e_{\theta_i} - \mathrm{v_{lat,id}})^{\mathrm{T}} \left(\left[\frac{\mathrm{e}_{\theta_i}}{\tau_s}, \mathrm{s_{lat,i}} \mathrm{e}_{\theta_i}^{\perp} \right] \right. \\ &\quad \left. \begin{bmatrix} s_{lat,i_{cmd}} \\ \omega_i \end{bmatrix} - \left(\frac{\mathrm{v_{lat,id}}}{\tau_s} + \dot{\mathrm{v}}_{lat,id} \right) \right) \end{aligned} \tag{3.46}$$

If ω_i was applicable as a control signal, choosing:

$$\begin{bmatrix} s_{lat,i_{cmd}} \\ \omega_i \end{bmatrix} = \begin{bmatrix} \tau_s e_{\theta_i} \\ s_{lat,i}^{-1}(e_{\theta_i}^{\perp})^T \end{bmatrix} \left(\frac{\mathrm{v}_{\mathrm{lat,id}}}{\tau_s} + \dot{\mathrm{v}}_{lat,id} \right) \tag{3.47}$$

(3.46) would be simplified to $\dot{P}_{\mathrm{v_i}} = -\frac{2}{\tau_s} P_{\mathrm{v_i}}$, from which, using standard Lyapunov analysis arguments, $P_{\mathrm{v_i}}$ can be shown to converge to zero. Since ω_i is not a control signal, we replace (3.47) with

$$: \begin{bmatrix} s_{lat,i_{cmd}} \\ \omega_{id} \end{bmatrix} = \begin{bmatrix} \tau_s e_{\theta_i} \\ s_{lat,i}^{-1}(e_{\theta_i}^{\perp})^T \end{bmatrix} \left(\frac{\mathrm{v}_{\mathrm{lat,id}}}{\tau_s} + \dot{\mathrm{v}}_{lat,id} \right) \tag{3.48}$$

and generate the command signal corresponding to ω_i using the compensator:

$$\omega_{i_{cmd}} = \frac{k_\omega \tau_\omega (s + 1/\tau_\omega)}{s + k_\omega} [\omega_{id}],\ k_\omega > 0 \tag{3.49}$$

so that (3.37) and (3.49) lead to:

$$\omega_i = \frac{k_\omega}{s + k_\omega} [\omega_{id}] \tag{3.50}$$

Using standard Lyapunov analysis and backstepping arguments it can be shown that (3.48), (3.49) force $P_{\mathrm{v_i}}$ and $\omega_i - \omega_{id}$ to converge to zero exponentially for sufficiently large k_ω. Summarizing our Lyapunov analysis based design, our low level motion controller for agent A_i is given by (3.44), (3.48), (3.49).

3.6 Simulations

In this section, we present the results of a set of simulations for a five agent formation using the control laws proposed in Sections 3.4 and 3.5.

3.6.1 Generic Fixed-Wing UAVs

The formation shape that is used for simulation purposes is illustrated in Fig. 3.1. As mentioned to keep the persistence of a formation with 5 agents in three dimensions, the $3 \cdot 5 - 6 = 9$ inter-agent distances remain constant. In the simulations, we have used the following parameters: $\varepsilon_k = 0.01$ and $d_{ij} = 1\ m$ for $(i, j) \in \{(1,2), (1,3), (2,3), (1,4), (2,4), (3,4), (1,5), (2,5), (3,5)\}$. The initial position of the agents are $p_{10} = (1, 0, 0)$, $p_{20} = (0.5, 0.866, 0)$, $p_{30} = (0, 0, 0)$, $p_{40} = (0.5, 0.2887, 0.8165)$ and $p_{50} = (0.5, 0.2887, -0.8165)$. The desired final positions of the p_1, p_2

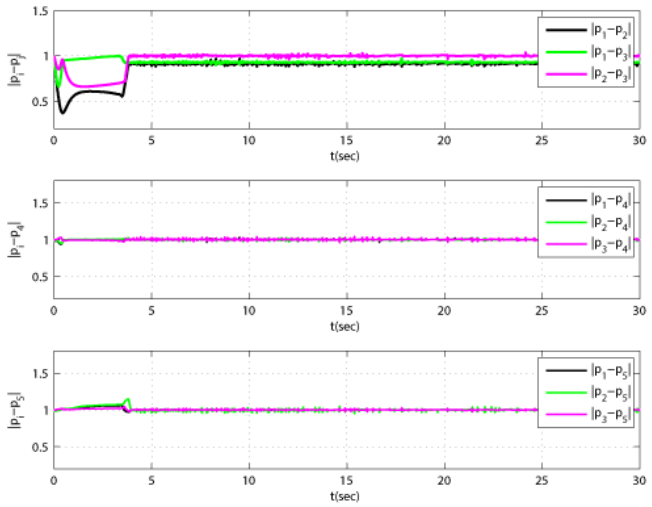


Figure 3.4: Inter-agent distances for the generic fixed-wing UAV formation case study

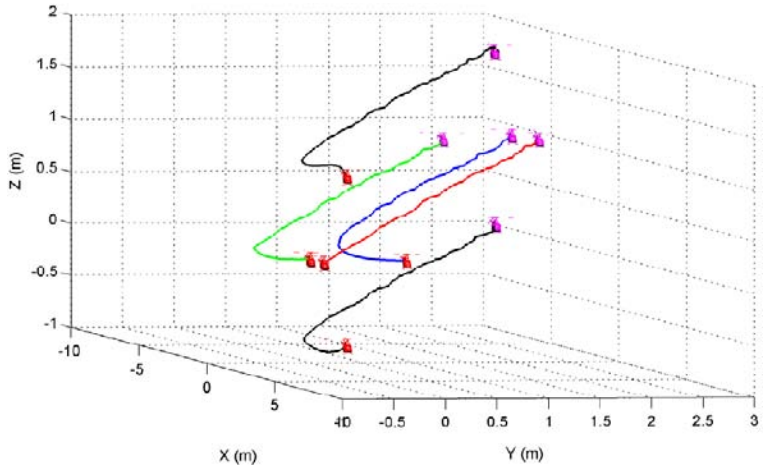


Figure 3.5: Pathway of autonomous agents for the generic fixed-wing UAV formation case study

and p_3 are $p_{1f} = (-3.9697, 2.7428, 1.0)$, $p_{2f} = (-4.8696, 2.5995, 1.0)$ and $p_{3f} = (-4.2134, 1.8462, 1.0)$. We simulate the effects of sensor noises using bounded random signals in the range $[-10^\circ, +10^\circ]$ added to the angles ϕ, θ, ψ in (3.29). The measured angular noise and disturbance are obtained a random signal, which is in between $\pm 10^\circ$. Furthermore, we use a time-delay block to simulate the measurement and processing delays in the

actuators. Transport delay time is taken as 0.02 sec in our simulations. Under these assumptions and for $\bar{\mathrm{v}} = 1$ m/s, as it can be seen in Figures 3.4 and 3.5, our five agents can move in formation to their final positions cohesively. It is clear that the agents do not collide with each other and successfully reach the final position. In this figures, we consider the kinematic modeling, and low level control design presented in Section 3.5.1 and 3.5.2. K_p and K_i are selected as $K_p = 1$, $K_i = 0.05$. Figures 3.4 and 3.5 demonstrate the effectiveness of the proposed distributed control scheme (3.33), (3.34).

3.6.2 Quadrotor Formations

For the better understanding of the implementation in the case study, we present the results of a set of simulations for a five quadrotor agents formation using the control laws proposed in Section 3.4. In the simulations, we have used the same inter-agent distances and initial positions as in Section 3.6.1. In addition, the final positions of leader, first follower and second follower are same with the final positions in Section 3.6.1. The final positions of ordinary followers are calculated with the help of sphere intersection algorithm which is described in Appendix.
The coefficients of $u_{i\phi}$ and $u_{i\theta}$ are $k_{dd} = 0.99,\ k_d = 1.48,\ k_p = 0.83,\ P_c = 100$ and $\bar{C}_i(s) = 2.7872(s + 3.57)/(s + 21.3)$, for $u_{i\psi}$ are $k_{dd} = 0.99,\ k_d = 0.67,\ k_p = 1.11,\ P_c = 100$ and $\bar{C}_i(s) = 0.0187(s + 1)/(s + 0.27)$,and the coefficients of u_{iz} are $\bar{k}_d = 7,\ \bar{k}_p = 7.5,\ \bar{k}_I = 6$. Additionally, the coefficients for the generation reference pitch angle θ^d are $k^r_{d_i} = 0.72,\ k^r_{p_i} = 0.72,\ k^r_{I_i} = 0.3$ and for reference roll angle ϕ^d are $k^r_{d_i} = 0.6,\ k^r_{p_i} = 0.6,\ k^r_{I_i} = 0.42$. Under these assumptions, as it can be seen in Fig. 3.6 and 3.7, our 5 quadrotor agents can move in formation to their final positions cohesively. It is clear that the agents do not collide with each other and successfully reach the final positions. In this application as we can see from Figure 3.7, the quadrotor vehicles try to meet the altitude requirement and reach the desired x and y coordinates at the same time.

3.6.3 Piccolo-Controlled Fixed-Wing UAV Formations

For the better understanding of the implementation on fixed-wing UAV formations, we present the results of a set of simulations for a five UAV agents formation using the dynamics in Section 3.5.2 and control laws proposed in Section 3.5.3. In the simulations, inter-agent distances are assumed to be $d_{ij} = 1$ m. In addition, the desired positions of agents in the leader-follower structured UAV formation are generated with the help of formulations as described in Section 3.3. The time constants τ_s, τ_ω and τ_z in (3.36)$-$(3.38) are assumed to be 0.01 sec and initial lateral speed is $s_{lat_0} = 1$ m/sec. Under these assumptions, as it can be seen in Fig. 3.8 and

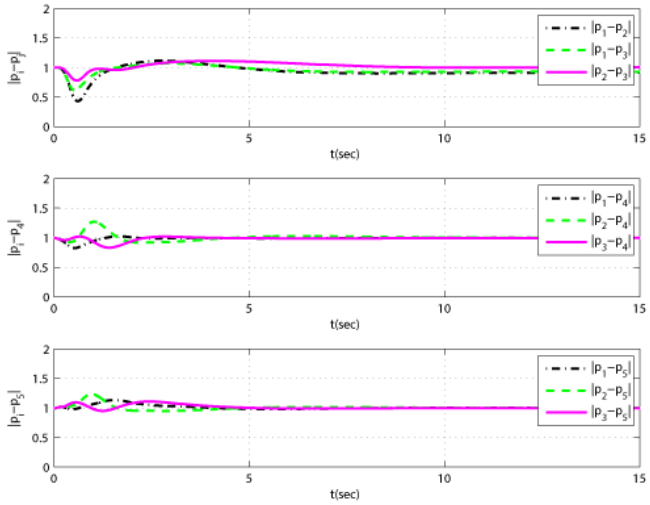


Figure 3.6: The inter-agent distances for the quadrotor formation case study

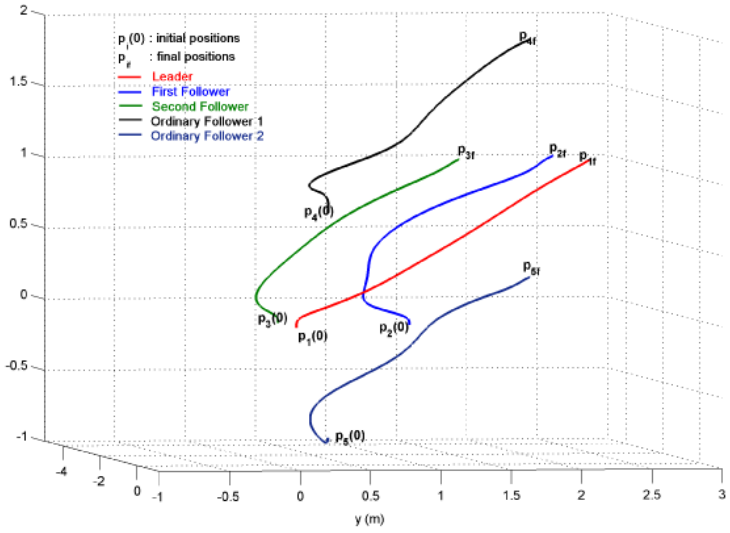


Figure 3.7: Pathways of agents for the quadrotor formation case study

Fig. 3.9, our five UAV agents can move in formation to their final positions cohesively. It is clear that the agents do not collide with each other and successfully reach their final positions. In this

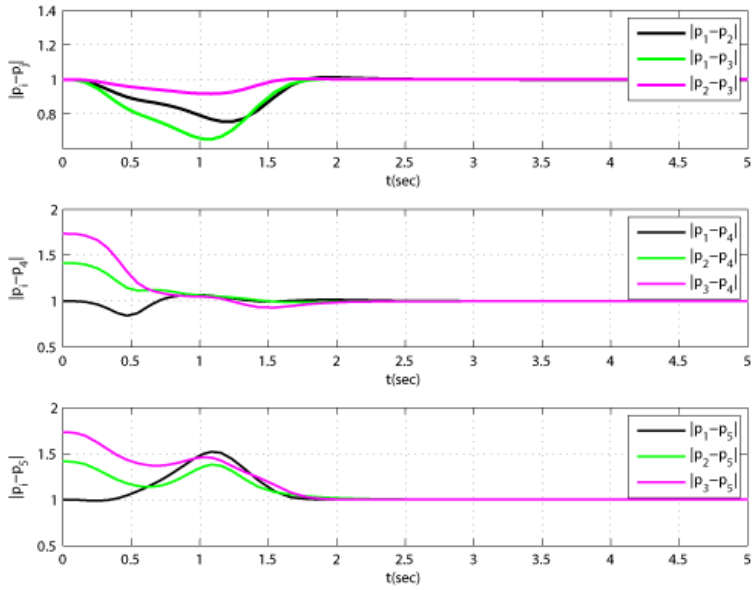


Figure 3.8: The inter-agent distances for the Piccolo-controlled fixed-wing UAV formation case study

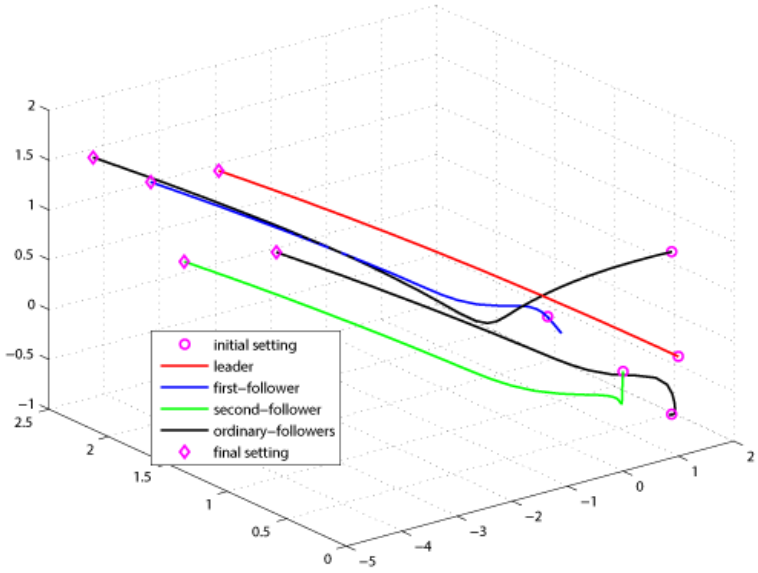


Figure 3.9: Pathways of agents for the Piccolo-controlled fixed-wing UAV formation case study

application as we can see from Fig. 3.9, the UAVs try to meet the altitude requirement and reach the desired x and y coordinates at the same time.

3.7 Conclusion

In this chapter, we have proposed a distributed cohesive motion control scheme for autonomous formations in three dimensions. The proposed control scheme is composed of two levels. The high level distributed trajectory generation scheme is based on geometric and graph theoretical analysis of the formation behavior during maneuvers. The lower level of the distributed control scheme is composed of the individual controllers of the agents that generate the applicable control signals to track the trajectories produced by the high level trajectory generator. For the low level control design, three case studies are considered; one with the modeling of quadrotors, one with single-velocity-integrator kinematics, and one with modeling of piccolo-controlled fixed-wing UAVs, have been considered in design and simulation testing of the distributed control algorithms. In addition, the sensory disturbances, measurement and processing delays of generic fixed-wing UAV testbeds are taken into account and the compensation of these terms are achieved using PID control action. The performance of the designed control schemes is verified via numerical simulations.
One theoretical research direction is enhancement of the dynamic controllers via constructive stability and convergence analysis. Other directions, considering real-time implementation, include consideration of detailed actuator dynamics and more realistic models of the units taking the dynamic uncertainties, external disturbances, and measurement noise effects into account [31]. After the verification of the reliability of our control algorithms, another future task is to test these algorithm in real-time experimental testbeds for different operational scenarios.

Chapter 4

Immersion and Invariance Based Adaptive Spatial Autonomous Maneuver Control

4.1 Introduction

This chapter proposes an adaptive control scheme for three dimensional maneuvering of autonomous vehicles that are required to track specified desired trajectories. Since the control signals to be generated are body frame fixed and the desired trajectories are often specified in the inertial frame, the motion control task involves measurements of the Euler angles defining the transformation between body and inertial frames via the navigation equipment mounted on the vehicles. The angular measurements carry noises, which bring parametric uncertainty to the corresponding motion kinematics. Our adaptive control design focuses on the parametric uncertainties caused by such measurement noises, treats the rotation matrix defined by the actual values of the Euler angles at each time instant as an unknown system matrix, and employs a recently developed immersion and invariance based adaptive control approach. After establishing applicability of this adaptive control approach to the motion control problem above, simulation results are provided demonstrating the performance of the adaptive control design for maneuvering scenarios involving single vehicles as well as autonomous vehicle formations.

In recent decades, autonomous vehicles have found applications in many areas, in the forms of unmanned ground vehicles (UGVs), unmanned aerial vehicles (UAVs), autonomous underwater vehicles (AUVs) and autonomous spacecraft [15, 23, 40, 42, 52, 70, 77, 80], or their robotic versions with robotic manipulators attached to them. Among these four types of autonomous vehicles, leaving UGVs aside, the motion space of the other three types, i.e. UAVs, AUVs and autonomous spacecraft are all three dimensional. Hence, consideration of three dimensional ma-

neuvers is essential in motion control of such autonomous vehicles, whether flying alone or in a multi-vehicle formation.

This chapter proposes an adaptive three dimensional motion control scheme for autonomous vehicles that are required to track specified desired trajectories. In this typical motion control problem for UAVs and AUVs, the applicable control signals are body frame fixed whereas the desired trajectories are specified in the inertial frame [17, 35]. Therefore the adaptive motion control scheme involves measurements of the Euler angles defining the transformation between body fixed and inertial frames via the navigation equipment mounted on the vehicles [39]. These angular carry noises, which bring parametric uncertainty to the corresponding motion kinematics.

Our adaptive control design focuses on the parametric uncertainties caused by the measurement noises and treats the rotation matrix defined by the actual values of the Euler angles at each time instant as an unknown system matrix, and employs a recently developed immersion and invariance based adaptive control approach for a certain class of linear multivariable systems [62]. After establishing applicability of the adaptive control scheme of [62] to the three dimensional maneuvering problem theoretically, we provide simulation results demonstrating the performance of the adaptive motion controller. The simulation studies and the discussions are provided for both single vehicle trajectory tracking tasks as well as cohesive motion control (simultaneous formation and motion control) of teams of vehicles.

The chapter is organized as follows. Section 4.2 provides background on kinematic modeling of three dimensional UAV and AUV maneuvers. Section 4.3 summarizes the immersion and invariance based approach of [62]. Section 4.4 first establishes applicability of this approach to the three dimensional autonomous vehicle maneuver control problem, and then presents the corresponding adaptive motion controller design. Section 4.5 discusses applications of the control scheme designed in Section 4.4 to the formation control of autonomous vehicle swarms. Section 4.6 presents simulation results demonstrating the performance of the proposed adaptive controller. Final remarks and summary are presented in Section 4.7.

4.2 Three Dimensional Autonomous Vehicle Maneuvers

In this section, we describe the three-dimensional maneuvering of a UAV and related potential practical sources of uncertainties before focusing on immersion and invariance based adaptive controller design, noting that similar attributes apply to AUVs. For this purpose Section 4.2.1 describes kinematic model of a UAV and Section 4.2.2 presents the navigation system and sensory suite to obtain the euler angles and acceleration of the UAV. The motion control problem for the UAV is described in Section 4.2.3.

4.2.1 Modeling Agent Kinematics

In this chapter, we model autonomous vehicles as point agents and ignore the dynamics issues originating from geometric properties of these vehicles.
We use the following kinematic model, which applies to autonomous aircraft, as well as spacecraft and submarines [17, 35]:

$$\dot{p} = V = \mathbf{R}_b^n V^b = \mathbf{R}_b^n \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{4.1}$$

$$\mathbf{R}_b^n = \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi c\phi s\theta \\ s\psi c\theta & c\psi c\phi + s\theta s\phi s\psi & -c\psi s\phi + s\theta s\psi c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \tag{4.2}$$

where $p(t) = [x(t), y(t), z(t)]^T$ denote the position and $V(t) = [v_x(t), v_y(t), v_z(t)]^T \in \Re^3$ denote the velocity of the center of mass of the autonomous agent at time t, $V^b = [u, v, w]^T$, $\bar{V}^b = \|V^b\|$ denote, respectively, the velocity vector of the autonomous agent in body axes and the translational speed of agent as depicted partially in Fig. 3.3, and three-dimensional kinematical transformation matrix for velocity vector from the body-axis to inertial frame is represented by $\mathbf{R}_b^n$. The Euler angles ϕ, θ, ψ represent the rotations with respect to x,y and z axes of the inertial frame, respectively. $s\cdot$ and $c\cdot$ are used to denote $\sin(\cdot)$ and $\cos(\cdot)$, respectively.

4.2.2 Navigation System and Measurement Issues

Navigation system is an essential part of kinematical modeling and adaptive control implementation. Body angular rates and linear accelerations of the autonomous vehicle are measured with the help of inertial navigation sensors, such as gyroscopes and accelerometers. Besides, angle-of-attack (α) and side-slip angle (β) are measured with the help of analog sensors. All these real-time measurements are used in generation of the control input u, v, w of the kinematic model in (4.1).
The angular rates and accelerations are used to obtain Euler angles, which are used to measure the transformation matrix R_b^n in (4.2) as R_{bm}^n as described in [39]. Since the measurement process consists of various error factors such as sensor location problems, high frequency disturbances due to the structure of micro-electromechanical sensors and cruise condition. Our measured and calculated Euler angles need to be considered with error factors, i.e., the actual rotation matrix can be written as

$$R_b^n = R_{bu}^n R_{bm}^n, \tag{4.3}$$

where

$$R_{bm}^{n} = \begin{bmatrix} c\psi_m c\theta_m & -s\psi_m c\phi_m + c\psi_m s\theta_m s\phi_m & s\psi_m s\phi_m + c\psi_m c\phi_m s\theta_m \\ s\psi_m c\theta_m & c\psi_m c\phi_m + s\theta_m s\phi_m s\psi_m & -c\psi_m s\phi_m + s\theta_m s\psi_m c\phi_m \\ -s\theta_m & c\theta_m s\phi_m & c\theta_m c\phi_m \end{bmatrix}$$

and

$$R_{bu}^{n} = \begin{bmatrix} c\psi_u c\theta_u & -s\psi_u c\phi_u + c\psi_u s\theta_u s\phi_u & s\psi_u s\phi_u + c\psi_u c\phi_u s\theta_u \\ s\psi_u c\theta_u & c\psi_u c\phi_u + s\theta_u s\phi_u s\psi_u & -c\psi_u s\phi_u + s\theta_u s\psi_u c\phi_u \\ -s\theta_u & c\theta_u s\phi_u & c\theta_u c\phi_u \end{bmatrix} \tag{4.4}$$

are, respectively, the measured and unknown parts of the rotation matrix R_b^n. Above, $\phi_m(t), \theta_m(t), \psi_m(t)$ are the measurements of the Euler angles $\phi(t), \theta(t), \psi(t)$, respectively; and ψ_u, ϕ_u, θ_u are the corresponding measurement errors.

4.2.3 Motion Control Problem

The motion control problem of interest is as follows:

Problem 1 *Consider the vehicle kinematic model in (4.1)–(4.3). Given a differentiable trajectory $p_d(t)$, design a control law for generating V_b so that $\|p(t) - p_d(t)\|$ converges to zero in time.*

Noting that R_{bm}^n is avaliable for measurement, and nonsingular for all time. We can rewrite (4.1)–(4.3) as:

$$\dot{p} = R_{bu}^n V_u^b, \tag{4.5}$$

$$V_u^b = R_{bm}^n V^b. \tag{4.6}$$

Hence Problem 1 simplifies to generation of V_u^b in (4.5) with the unknown matrix R_{bu}^n for mentioned tracking problem.Because of the unknown part of the model, a particular adaptive control design approach is taken for the path tracking of the autonomous agent as described in Section 4.3.

4.3 Immersion and Invariance Adaptive Control of Linear Multivariable Systems

In [62], an adaptive control scheme is proposed for multi-variable systems of the form

$$\dot{y} = K_p u \tag{4.7}$$

where $u, y \in \Re^m$ and K_p is an unknown nonsingular $m \times m$ matrix. Both regulation and tracking problems are considered for the system (4.7). In the tracking problem, y is required to track a known differentiable trajectory y^*, whose derivative $\dot{y}^*$ is also available. The proposed adaptive control scheme requires existence and knowledge of a nonsingular matrix $\Gamma \in \Re^{mxm}$ such that

$$K_p\Gamma^T + \Gamma K_p^T > 0. \tag{4.8}$$

Once the condition (4.8) is satisfied, Γ is used as an adaptive gain matrix in the adaptive control scheme. The benefits of this adaptive control scheme, compared to other adaptive control designs in the literature for similar plants, are mainly requirement of (4.8) in place of the (typical) more restrictive condition $K_p\Gamma^T = \Gamma K_p^T > 0$ and simplicity of the resultant adaptive control law, without requiring any projections or a priori information other than the matrix Γ. In general requirement of knowing Γ is restrictive in practice; however, there exist practical cases where this requirement is satisfied, as it is demonstrated in Section 4.4. The aforementioned immersion and invariance based adaptive control law for tracking the reference trajectory y^* is as follows:

$$u = \Psi(\widehat{\theta} - \Psi_f^T\Gamma^{-1}e_f) - \Psi_f\Psi_f^T\Gamma^{-1}e, \tag{4.9}$$

$$\dot{\widehat{\theta}} = -(2\Psi_f - \Psi)^T\Gamma^{-1}e_f \tag{4.10}$$

where $e = y - y^*$, $\dot{e}_f = -e_f + e$, $\dot{y}_f^* = -y_f^* + y^*$ and

$$\Psi := \begin{bmatrix} (e-\dot{y}^*)^T & 0\cdots 0 & \dots & 0\cdots 0 \\ 0\cdots 0 & (e-\dot{y}^*)^T & \dots & 0\cdots 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0\cdots 0 & 0\cdots 0 & \dots & (e-\dot{y}^*)^T \end{bmatrix} \in \Re^{m\times m^2} \tag{4.11}$$

$$\Psi_f := \begin{bmatrix} (e_f-\dot{y}_f^*)^T & 0\cdots 0 & \dots & 0\cdots 0 \\ 0\cdots 0 & (e_f-\dot{y}_f^*)^T & \dots & 0\cdots 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0\cdots 0 & 0\cdots 0 & \dots & (e_f-\dot{y}_f^*)^T \end{bmatrix} \in \Re^{m\times m^2} \tag{4.12}$$

It is established in [62] that the control law (4.9)–(4.12) guarantees $\lim_{t\to\infty} e(t) = 0$ with all the signals in (4.7),(4.9)–(4.12) being bounded for all initial conditions $y(0), e_f(0), y_f^*(0) \in \Re^m, \hat{\theta} \in \Re^{m^2}$ and all bounded trajectories y^* with bounded derivative $\dot{y}^*$. Section 4.4 develops an adaptive control law of the form (4.9)–(4.12) to solve Problem 1.

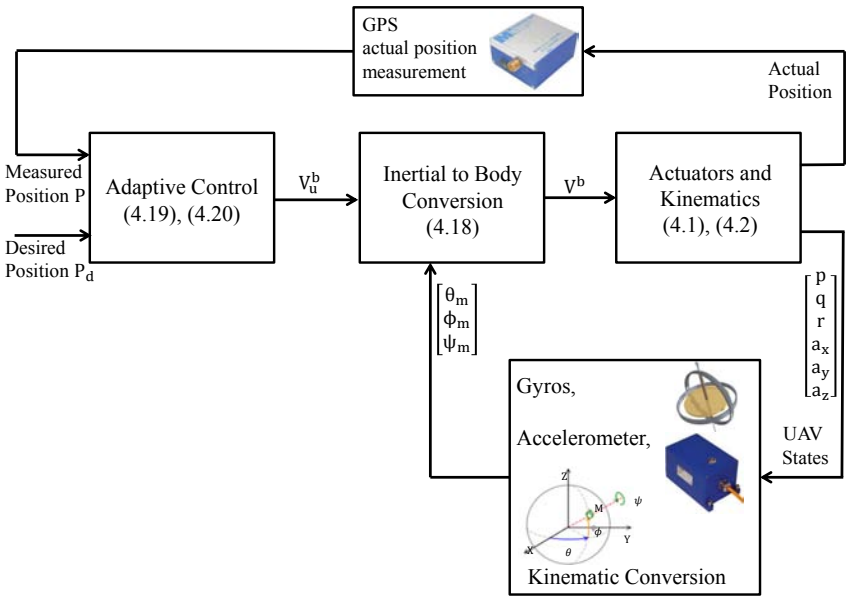


Figure 4.1: Derivation procedure of Euler angles with respect to inertial axes with the help of gyros and accelerometers.

4.4 Adaptive Motion Control Design

In order to apply the adaptive control law (4.9)−(4.12), we establish that our system can be modeled in the form of (4.7) and fix an adaptive gain matrix Γ satisfying (4.8). We follow a pointwise (in time) design approach [91] and treat R_{bu}^n as constant with value $R_{bu}^n(t)$ in assignment of the control signal $V_u^b(t)$ at time instant t. Below we use the fact that any rotation that is described by three Euler angles can also be described by a rotation axis−rotation angle pair (see, e.g., Chap. 5 of [43]), and for the same representation $r_\alpha - \alpha$ of R_{bu}^n, which is the rotation matrix corresponding to small angular measurement errors, we assume that $|\alpha| < \frac{\pi}{2}$.

Lemma 1 *The system (4.5) is in the form (4.7), where $K_p = R_{bu}^n$ is nonsingular. Furthermore, assuming that the rotation angle α in 0 the rotation axis−rotation angle representation of R_{bu}^n satisfies $|\alpha| < \frac{\pi}{2}$, for the same system, (4.8) is satisfied for $\Gamma = \gamma I$, where $\gamma > 0$ is an arbitrary scalar.*

Proof 1 *Since R_{bu}^n is a rotation matrix, it is nonsingular. Hence, the system (4.5) is in the form (4.7).*

To establish that (4.8) is satisfied, i.e. to find a matrix Γ satisfying (4.8), we use the properties of rotation matrices. Any rotation that is described by three Euler angles can also be described by a rotation axis - rotation angle pair (see, e.g., Chap. 5 of [43]). Hence R_{bu}^n corresponds to a rotation around a certain axis vector $r_\alpha = [r_x, r_y, r_z]^T$ by a certain angle α. Furthermore applying the procedure in Section 2.2.3 of [36], R_{bu}^n can be written in the form

$$R_{bu}^n = W^T \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} W \tag{4.13}$$

where W is a (orthogonal) rotation matrix that is defined by the entries of r_α. Therefore, for any $\nu \in \Re^3$,

$$\nu^T R_{bu}^n \nu = \nu^T W^T \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} W\nu \tag{4.14}$$

Let

$$\omega = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = W\nu \tag{4.15}$$

Then

$$\begin{aligned} \nu^T R_{bu}^n \nu &= \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 \end{bmatrix} \begin{bmatrix} \omega_1 \cos(\alpha) - \omega_2 \sin(\alpha) \\ \omega_1 \sin(\alpha) + \omega_2 \cos(\alpha) \\ \omega_3 \end{bmatrix} \\ &= \left(\omega_1^2 + \omega_2^2\right) \cos(\alpha) + \omega_3^2 \end{aligned} \tag{4.16}$$

Hence, for $|\alpha| < \frac{\pi}{2}$, $\nu^T R_{bu}^n \nu = \nu^T (R_{bu}^n)^T \nu > 0$ for any $\nu \neq 0$. Therefore,

$$R_{bu}^n + (R_{bu}^n)^T > 0 \tag{4.17}$$

and hence (4.8) is satisfied for $\Gamma = \gamma.I$ with any scalar $\gamma > 0$.

Next, we rewrite the adaptive control scheme (4.9)−(4.12) for our specific case (4.5),(4.6):

$$V_b = (R_{bm}^n)^{-1} V_u^b, \tag{4.18}$$

$$V_u^b = \Psi(\widehat{\theta} - \Psi_f^T \Gamma^{-1} e_f) - \Psi_f \Psi_f^T \Gamma^{-1} e, \tag{4.19}$$

$$\dot{\widehat{\theta}} = -(2\Psi_f - \Psi)^T \Gamma^{-1} e_f \tag{4.20}$$

where $e = p - p_d$, $\dot{e}_f = -e_f + e$, $\dot{p}_d^f = -p_d^f + p_d$ and

$$\Psi := \begin{bmatrix} (e - \dot{p}_d)^T & 0_3^T & 0_3^T \\ 0_3^T & (e - \dot{p}_d)^T & 0_3^T \\ 0_3^T & 0_3^T & (e - \dot{p}_d)^T \end{bmatrix} \in \Re^{3\times 9} \tag{4.21}$$

$$\Psi_f := \begin{bmatrix} (e_f - \dot{p}_d^f)^T & 0_3^T & 0_3^T \\ 0_3^T & (e_f - \dot{p}_d^f)^T & 0_3^T \\ 0_3^T & 0_3^T & (e_f - \dot{p}_d^f)^T \end{bmatrix} \in \Re^{3\times 9} \tag{4.22}$$

4.5 Application to Formation Control

The adaptive control law proposed in Section 4.4 for trajectory tracking of a single UAV can be utilized in formation control of teams of such UAVs as well. In multi-layer distributed control schemes composed of a high-level task distribution and desired trajectory generation scheme and low level trajectory tracking individual controllers, such as the one in [6], (4.18)–(4.20) can be used as an adaptive control law within the low level controllers to generate the body fixed velocity commands. As a case study, we focus on the rigid formation control problem of [6, 7], where the distance between pairs of UAV agents are required to be maintained to move the team of UAVs cohesively as a rigid body.

The *cohesive motion control* task defined in [6, 7] is carrying a group $S = \{A_1, \cdots, A_m\}$ of vehicles from a given position-orientation setting to a desired final one, preserving the initial geometry (inter-vehicle distances) to a desired final location. The task allocation for the multi-agent swarm S is represented by a directed graph $G_F = (V_F, E_F)$, called the directed underlying graph of S, where each vertex i in the vertex set V_F corresponds to an agent A_i in S, and each directed edge $\overrightarrow{(i,j)}$ in the edge set E_F from i to j denotes that A_i can sense its distance from A_j and is responsible to keep a desired distance d_{ij} from A_j [7]. In this case, we also say that A_i follows A_j. In the sequel, we call the swarm together with the underlying graph $G_F = (V_F, E_F)$ and the desired distance set $D_F = \{d_{ij} \mid \overrightarrow{(i,j)} \in E_F\}$ a *formation* and represent by $F = (S, G_F, D_F)$.

A necessary condition for achievement of the above task is *persistence* of G_F [6, 7, 89]:

The main purpose in *rigid* formation is keeping the distance d_{ij} between each agent pair (A_i, A_j) constant, which means maintaining the formation shape during the motion of the swarm. In a formation, if each agent in the formation is always able to satisfy the distance constraints it is responsible for, then this formation is called *constraint-consistent*. A formation that is both rigid and constraint-consistent is called *persistent*. Furthermore, a persistent formation maintaining its persistence with the minimum possible number of links, which is $|E_f| = 3|V_f| - 6$ in

three dimensions, is called *minimally persistent*. More formal definitions of *rigidity, constraint-consistence*, and *persistence* in three dimensions can be found in [6, 7].
We assume the following for the cohesive motion control problem of interest, to be formally defined next:

A1: Each agent A_i in S can perfectly measure its own (center of mass) position $p_i(t) = [x_i(t), y_i(t), z_i(t)]^T$ and the (relative) positions of the agents it follows at any time.

A2: The distance-sensing range for a neighbor agent pair (A_i, A_j) is sufficiently larger than the desired distance d_{ij}.

A3: F is minimally persistent. Denoting the number of edges originating from vertex i by $d^+(i)$, $d^+(1) = 0$, $d^+(2) = 1$, $d^+(3) = 2$; $(\overrightarrow{2,1})$, $(\overrightarrow{3,1})$, $(\overrightarrow{3,2}) \in E_F$.

A4: Each A_i ($i \in \{1, 2, 3\}$) knows its final desired positions p_{if}.

The cohesive motion control problem for UAV can be stated as:
Problem 2: Consider a swarm S with m agents $\{A_1, ..., A_m\}$ ($m \geq 4$) that are initially at positions, respectively, $p_{10}, ..., p_{m0} \in \Re^3$, consistent with D_F. The control task is, under Assumptions A1$-$A4, to move S to a final desired setting defined by a set of desired final positions, $p_{1f}, ..., p_{mf}$, which is consistent with D_F, without deforming the shape of the formation, i.e. forcing the inter-agent distances track the constant values defined in D_F, during motion.

As a sample scenario, consider five autonomous vehicles $A_1, ..., A_5$ for our cohesive formation F moving in $\Re^3$, as depicted in Fig. 3.1. Problem 2 is approached in [6, 7] by a two-level control scheme: A high level trajectory generator and low level individual adaptive motion controllers. Here, we consider the same scheme where the low level motion controllers of the agents are taken as those presented in Section 4.4, used to track the trajectories generated at high level. Additionally, the high level trajectory generator for the formation flight is the same with Section 3.4.1 [6, 7].

4.6 Simulations

In this section, the performance of immersion and invariance based adaptive controllers is tested for both single agent trajectory tracking and formation control.

4.6.1 Single Agent Flight

Various single agent scenarios were used the test the adaptive control design of Section 4.4, and compare with some non-adaptive controllers. Here, we present a case where the desired position vector $p_d = [p_{nd}, p_{ed}, p_{hd}]$ is the helical trajectory $p_{nd} = 20\sin(\frac{\pi}{9}t), p_{ed} = 20\sin(\frac{\pi}{9}t), p_{hd} = 5t$, where n, e, h stand for north, east and height. In these simulations, adaptive control gain Γ is taken as I. A realistic angular noise, which includes white noise, angle bias, and low frequency sinusoidal affect, is considered in our measurements and directly involved in our control signal during both adaptive and non-adaptive cases. The components of sensor noises added to the angles ϕ, θ, ψ are bias error in the range $[0^\circ, 15^\circ]$, band-limited white noise with 0.3 noise power and 0.1s sample time, and sinusoidal error with 30° amplitute and 1 rad/s frequency.

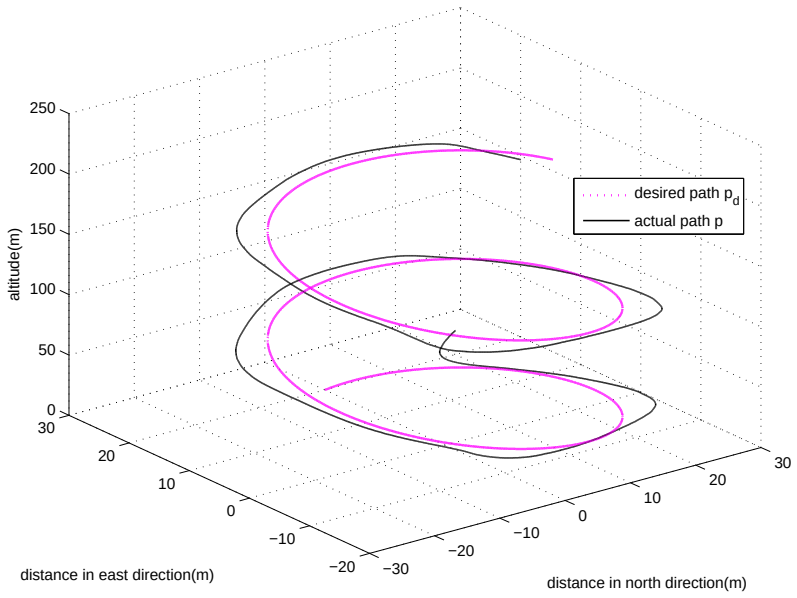


Figure 4.2: Non-Adaptive based helical trajectory tracking

As can be seen in Fig. 4.2 and Fig. 4.3, UAV with non-adaptive controller is not settled well, and distance error and root mean square error (RMSE) cannot be cancelled in spite of reinforcing P based non-adaptive control in (4.18) with PI control. However, the immersion and invariance based adaptive control provides perfect tracking of the helical trajectory and settlement in less than 0.1 sec, for the same amount of measured noise, as shown in Fig. 4.4 and Fig. 4.5.

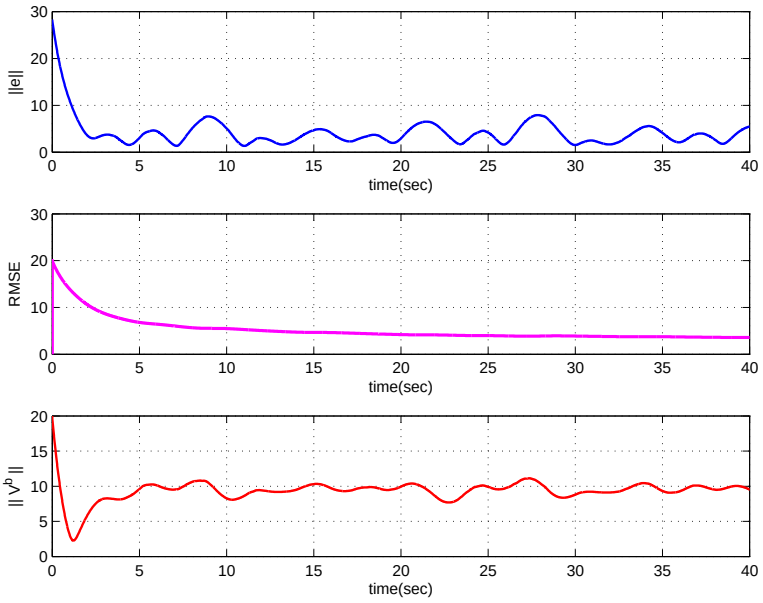


Figure 4.3: Distance error, RMSE, and velocity profile of helical trajectory tracking using non-adaptive control

4.6.2 Multi Agent Formation Flight

Consider the sample formation depicted in Fig. 3.1, and the task of solving Problem 2 for this formation. The initial positions of the agents are $p_{10} = (1, 0, 0)$, $p_{20} = (0.5, 0.866, 0)$, $p_{30} = (0, 0, 0)$, $p_{40} = (0.5, 0.2887, 0.8165)$ and $p_{50} = (0.5, 0.2887, -0.8165)$. The desired final positions of the p_1, p_2 and p_3 are $p_{1f} = (-3.9697, 2.7428, 1.0)$, $p_{2f} = (-4.8696, 2.5995, 1.0)$ and $p_{3f} = (-4.2134, 1.8462, 1.0)$. In the simulations, we have used the same angular sensor noise effects as in Section 4.6.1; the adaptive control gain Γ is taken as $0.5I$ for A_1 and $10I$ for all other agents, and all the desired inter-vehicle distances are taken as 1 m.

Here, the formation is tested for P and PI based non-adaptive controller as given in (4.18). As can be seen from Fig. 4.6 and Fig. 4.7, the non-adaptive controlled formation settled after 5 sec and the inter-agent distance requirements are violated. However, immersion and invariance based adaptive controller provides fast tracking of the trajectories, with a settlement time around 0.1sec, and inter-agent distance maintenance is successful.

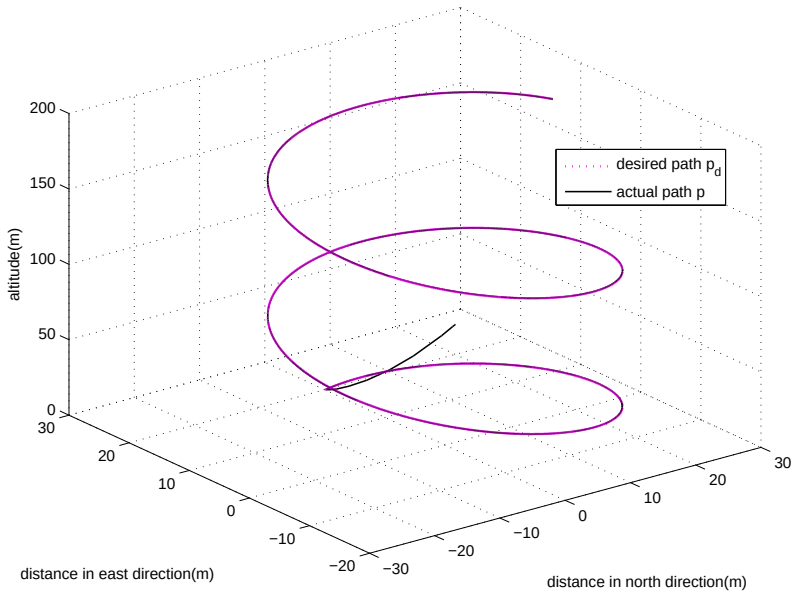


Figure 4.4: Immersion and invariance based adaptive helical trajectory tracking

4.7 Conclusion

This chapter presents an adaptive control design for three dimensional motion control of autonomous vehicles of UAV or AUV type, based on a particular immersion and invariance approach. It is demonstrated via simulation results that the designed adaptive controller successfully handles the affects of the noises in the Euler angle measurements provided by the navigation units of such autonomous vehicles and meets the specified three dimensional trajectory tracking task. In addition, the designed adaptive controller is also applied to motion control of autonomous multi-vehicle formations and the performance of the adaptive controllers are validated with number of simulations.

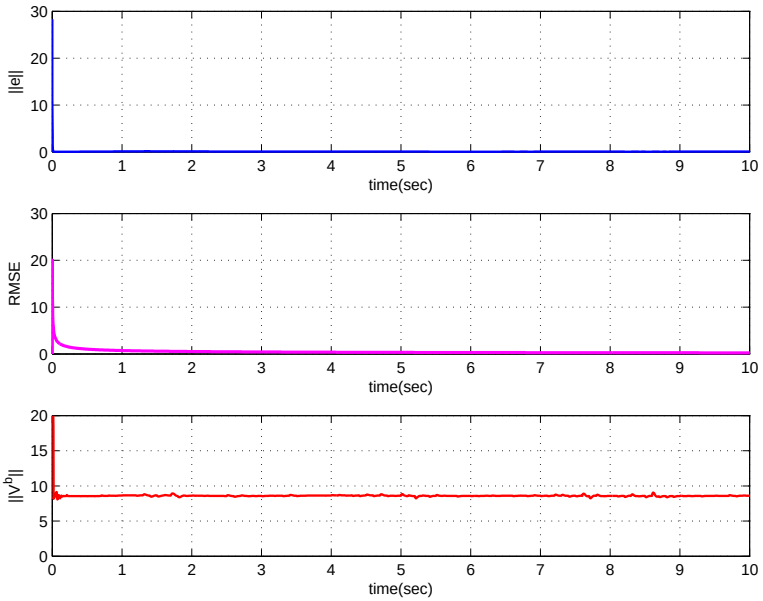


Figure 4.5: Distance error, RMSE, and velocity profile of helical trajectory tracking using immersion and invariance based adaptive control

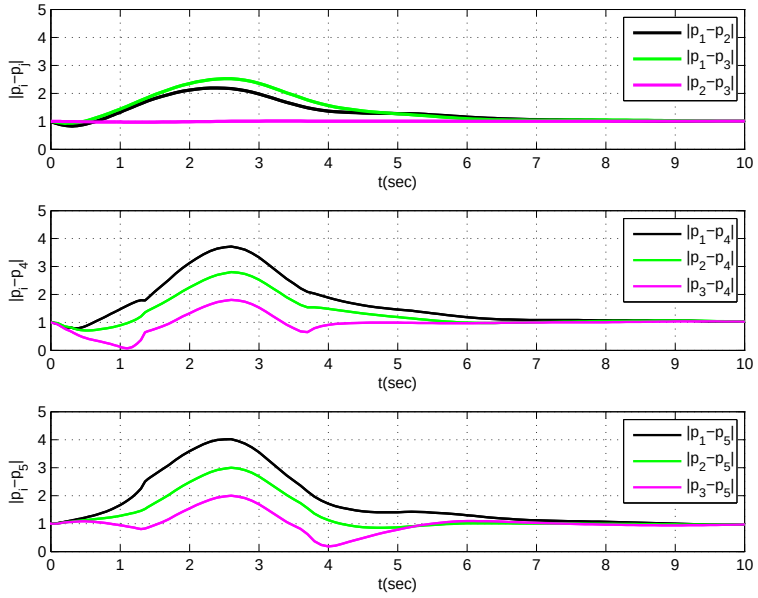


Figure 4.6: Inter-vehicle distances of leader-follower based 5-agents scenario using non-adaptive control

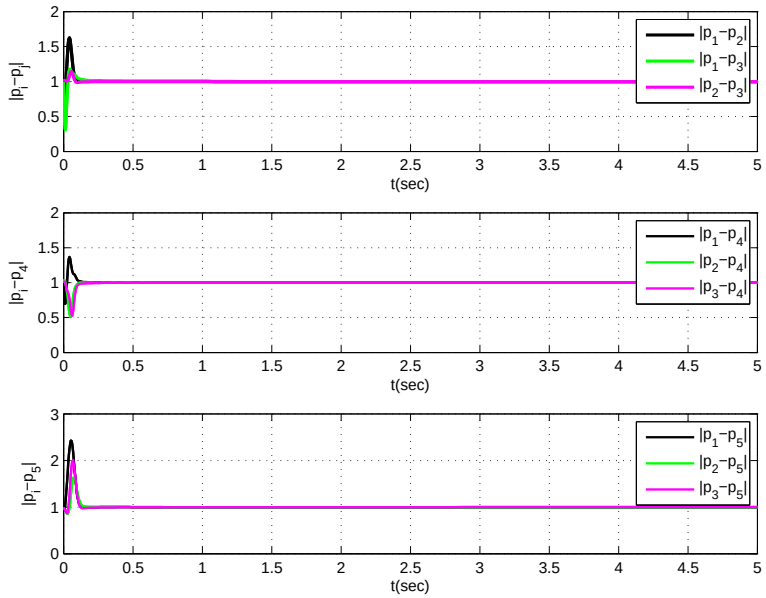


Figure 4.7: Inter-vehicle distances of leader-follower based 5-agents scenario using immersion and invariance based adaptive control

Chapter 5

Design and Implementation of String Stable Driving for Highway and Urban Missions

5.1 Introduction

The importance of intelligent transportation systems (ITS) emerged with increased traffic congestion on urban streets and highways, limited highway capacity and transportation infrastructure, climbing air pollution in cities, and loss of time and money in the traffic. These are fast growing problems of most crowded cities. Additionally, human error is credited for 90% of vehicle accidents. Driver assistance systems aim preventing from accidents due to lack of human concentration and distraction [83].

Developments in autonomous driving technology and the decreasing cost of sensors and computation enable the practical implementation of cooperative driving techniques. Cooperative driving and platooning of commercial vehicles offer significant advantages, such as reduced traffic congestion [1]. Many automobile companies have started to produce different car models including driver assistance systems and adaptive cruise control (ACC) support, which enables implementation of high level cooperative algorithms with simple modifications. The implemented technology and general experience in the 2007 DARPA Urban Challenge is the proof of the feasibility of autonomously navigating vehicles in urban traffic. There are many other projects dedicated to driver assistance and driving safety in Europe, Japan, and the US supported by governmental organizations and private sector [2].

The Dutch Organization for Applied Scientific Research (TNO), one of the leading European in-

stitutions for collaborative driving research and technologies, organized the GCDC 2011 in Helmond, the Netherlands, where 8 different teams and more than 15 institutes participated. There were two main missions in GCDC 2011: urban and highway challenges. The vehicles were trying to demonstrate the most effective cooperative driving capability and communication system for predetermined traffic scenarios. GCDC experience and demonstrations of TNO with their autonomous Toyota Prius fleet clearly showed the benefits of collaborative autonomous driving over standard human-controlled vehicles in an urban and highway traffic environments [47, 56].

While each individual vehicle controlled its longitudinal motion autonomously, steering of ve-

Figure 5.1: Test vehicle for testing platooning strategies

hicles was still controlled by the human driver. All teams were required to transmit their position and state data of their vehicles through wireless communication technology. This data is received by other teams for designing their own longitudinal collaborative control strategies in a heterogeneous vehicle set. The availability of position and state data of preceding vehicles in the platoon is necessary for stronger string stability and the implementation of different longitudinal collaborative cruise control strategies. If sudden changes in the velocity of a lead vehicle are attenuated by the vehicles upstream in the platoon, the longitudinal dynamics of the platoon are called string stable. If the velocity changes are amplified, the longitudinal dynamics are string unstable [58]. Despite having many adaptive controlled commercial cars supported with additional driver assistance systems on the current automobile market, their technologies are not sufficient to overcome the string instability in daily traffic. Considering this aspect, one major

benefit of adding inter-vehicle communication is to improvement of string stability of the vehicle platoons. This improvement leads us to reduce traffic congestion in a daily urban and highway traffic.
There were mainly two different scenarios; (a) urban and (b) highway in GCDC. The performance in safe and string stable driving of 8 teams were evaluated during 18 consecutive heats for two days in the challenge. Each heat incorporated both urban and highway scenarios. While starting each heat, the platoon position of each vehicle was replaced for better evaluation of their performances. Additionally, vehicles were trying to adjust their speed according to set points broadcast by lead vehicle, considering safety and evaluation criteria. Each vehicle was responsible for optimizing its acceleration profile in the platoon using own sensory information and other vehicles' wireless data during GCDC heats, considering safety requirements and evaluation criteria detailed in Section 5.6.
We have participated in the challenge as members of the team Futurum, which is formed by students and researchers from Fontys Automotive Research Center, the University of Twente, and the University of Waterloo. The main focus of Team Futurum in GCDC was the practical implementation of theoretical control and communication concepts for consistent and smooth Cooperative Adaptive Cruise Control (CACC). We first use simplified car dynamics and sensor models in Matlab/Simulink setting in order to define optimal control coefficients in computer simulations. However, it was not possible to include realistic noise and disturbance effects via computer simulations for various mission traffic scenarios. Thus, practical tests of theoretical concepts can show our dynamical constraints and help configuring our control and modeling parameters for better CACC implementation. We used customized tools of xPC Target-Box for longitudinal control of the Smart Fortwo as depicted in the lower portion of Fig. 5.1.
This chapter presents our theoretical control, hardware design, and analysis studies, some of which were implemented and tested by Team Futurum during GCDC 2011. The background of urban and highway missions is given in Section 5.2, implementation of high level collaborative controllers is presented in Section 5.3, Futurum vehicle and additional hardware design strategy is in Section 2.4.3, low level control design is described in Section 5.5, control performance of Team Futurum during GCDC 2011 is evaluated in Section 5.6, and concluding remarks are provided in Section 5.7.

5.2 Background

Consider the urban and highway mission scenarios of GCDC illustrated in Fig. 5.2. The performance in safe and string stable driving of 8 teams were evaluated during 18 consecutive heats for two days in the challenge. Each heat incorporated both urban and highway scenarios. In different

runs, the platoon position of each vehicle was changed for better evaluation of the performances. Additionally, vehicles were trying to adjust their speed according to set points broadcasted by the lead vehicle, considering safety and evaluation criteria. Each vehicle is responsible for optimizing its acceleration profile in the platoon using own sensory information, wireless state information of preceding vehicles, and commands from road side units (RSUs) during each heat considering safety requirements and evaluation criteria detailed in Section 5.6.

5.2.1 Urban and Highway Missions

Vehicle teams were first split into two platoons at the beginning of *urban scenario* as depicted in Fig. 5.2. Front and back platoons waited for the green light signal, which was transmitted via RSUs. At first, the back platoon was triggered with a green light. Then, vehicles started to move, obeying the string stability condition described in Section 5.6. In a typical run, while platoon 1 is approaching, the lead vehicle starts to accelerate with a signal of green light. Thus, other participants automatically accelerate in order to follow lead vehicle. The urban mission run ends when all vehicles cross the finish line. This scenario is for testing urban behaviors of collaborative vehicles and for evaluating interactions between separate mini platoons.
The *highway scenario* is for testing the ability of keeping stable strings under various aggressive acceleration and deceleration periods introduced by the GCDC lead vehicle. This acceleration profile (shockwave) is determined as the main reason of traffic congestion problem. Both urban and highway missions are illustrated by detailed figures in [47].

5.3 High Level Control Implementation

In this section, we present the implementation of ACC and different CACC controllers, and compare the behaviors of different high level controllers using simulation results (see Table 5.1).

5.3.1 System Modeling

CACC model presented in [58] is used as a base model of the control implementation. The vehicle dynamics are modeled as a longitudinal moving mass with a first order actuator system and actuator delay for validating our controllers.

$$x_i = \frac{1}{m_i s^2} \frac{K_i}{\tau_i s + 1} e^{-t_{di} s} f_i \tag{5.1}$$

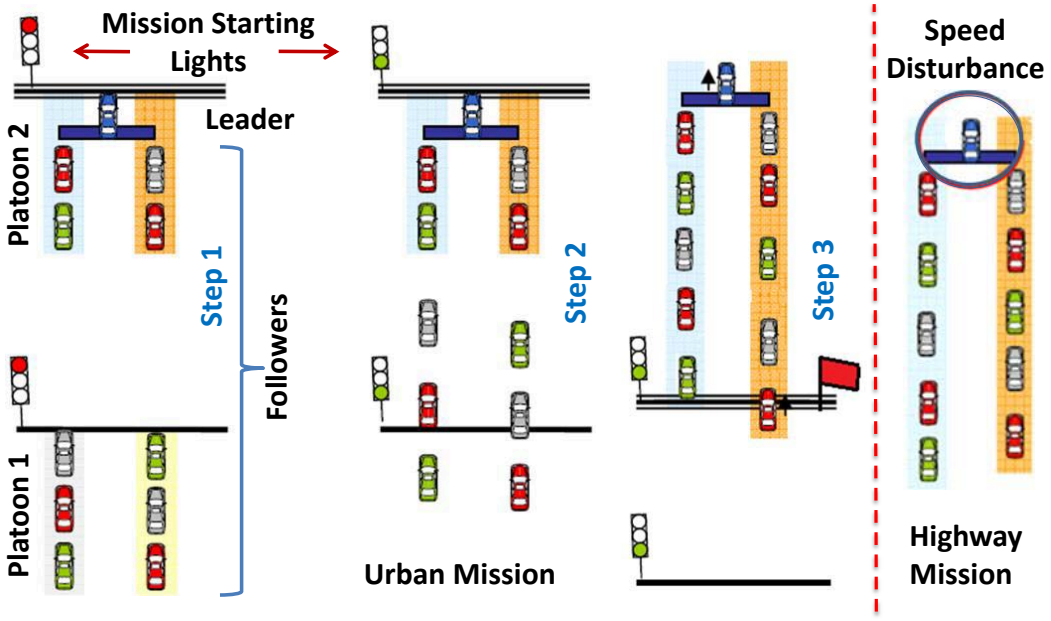


Figure 5.2: The overview of sample urban and highway traffic scenarios

where K_i, τ_i, t_{di}, and f_i represent the steady state gain, time constant of the actuator dynamics, the actuator delay, and actual force, respectively. There is some delay associated with the actuator dynamics (i.e. braking system, engine etc.) as seen in (5.1). For stabilizing the vehicle's longitudinal motion and keeping a safe distance between vehicles in a platoon, both position error and velocity error are considered and command force f_{di} is generated as the output of high-level controller and input of actuator control block. The position error, velocity error, and desired relative distance are defined, respectively, as:

$$e_{x_i} = x_{i-1} - x_i - x_{r,di}, \tag{5.2}$$

$$e_{v_i} = v_{i-1} - v_i, \tag{5.3}$$

$$x_{r,di}(t) = x_{r,0,i} + h_{di}\dot{x}_i(t) \tag{5.4}$$

where (i) represents ego vehicle, and $(i-1)$ represents preceding vehicle, $x_{r,0,i}$ and h_{di} are minimum relative distance between vehicles at standstill and desired time headway, respectively. Actuator dynamics limit our bandwidth and directly effect our controller activity. In this sense, the faster actuator dynamics we have, the better controller performance we derive. Additionally, we use first order low-pass filtered values of $\dot{x}_i$ measurements due to high-frequency noises in

real-time tests.

5.3.2 Cooperative Driving of Smart Fortwo Vehicles

The high level control in cooperative platooning is split into four main parts. These are cruise control, ACC and CACC as in Fig. 5.4, and manual control mode. These different control settings are necessary according to the order of the Smart in the platoon and the information flow during highway and urban scenarios of GCDC. Our control configuration receives the inertial and kinematic information related to the motion of the Smart by means of internal wheel encoders, accelerometers, gyros, GPS, Lidar, and other internal sensors. The traffic light, maximum speed, and the states of preceding vehicles are obtained with the help of wireless transceivers. The general control laws for these three main collaborative driving methods are given as follows:

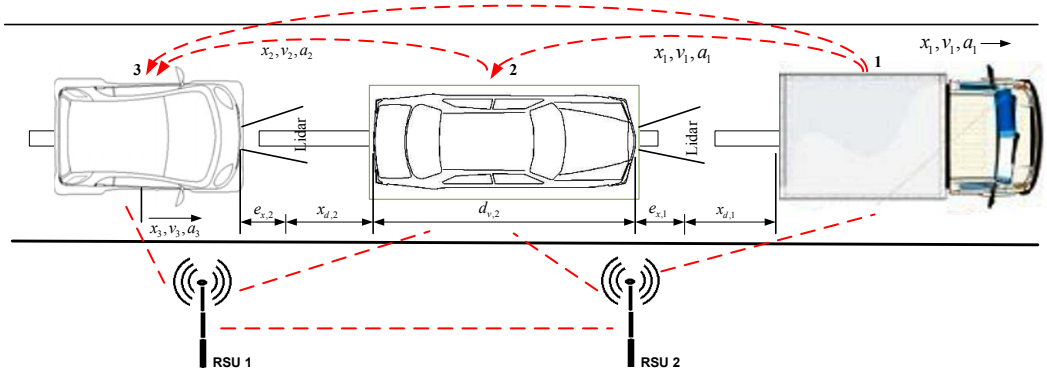


Figure 5.3: Three vehicle string in a platoon with embedded sensors, V2V, and V2I wireless communication

For conventional cruise control (no predecessor case):

$$f_{di} = m_i k_{di}(v_{di} - v_{enc,i}) \tag{5.5}$$

where f_{di} is the command force for the actuators of the ego-vehicle, k_{di} is a constant proportional gain, v_{di} is the desired cruise speed, and $v_{enc,i}$ is the measured velocity with wheel encoders. In the ACC case, the predecessor is detected via lidar or radar and there is no vehicle to vehicle communication [56]. This representation is referred to distance control mode in the literature and the model can be seen in Fig. 5.4. The desired acceleration is obtained with the PD regulation of relative velocity and relative distance errors between ego vehicle and the preceding vehicle as in below:

$$f_{di} = m_i(\omega_{K,i} e_{vi} + \omega_{K,i}^2 e_{xi}) \tag{5.6}$$

where e_{vi} and e_{xi} are relative velocity error and distance error. The control signal f_{di} gives desired force of the ego vehicle for making more stable while moving in the platoon cooperatively. ACC in (5.6) is designed as a PD filter with a breakpoint at $\omega_{K,i}$ rad/s, which assumes satisfying the condition $\frac{1}{\tau_i} \gg \omega_{K,i}$ [57]. This controller also has a mass factor that eliminates the mass constant in (5.1). Actuator dynamics limit our bandwidth and directly effects our controller activity. In this sense, the faster actuator dynamics we had, the better controller performance we derived.
In the CACC case illustrated in Fig. 5.3, the preceding vehicle's acceleration is obtained via the wireless communication link and the control law for ACC in (5.6) is modified as:

$$f_{di} = m_i(\omega_{K,i}e_{vi} + \omega_{K,i}^2 e_{xi}) + F_i D_i a_{i-1} \tag{5.7}$$

where F_i is feed-forward filter and details of the design of F_i filter is described in Chapter 6. The schematic control diagram for the ACC and CACC model can be seen in Fig. 5.4. The other variables are: Optimal design of F_i can be achieved with using vehicle mass, actuator and spac-

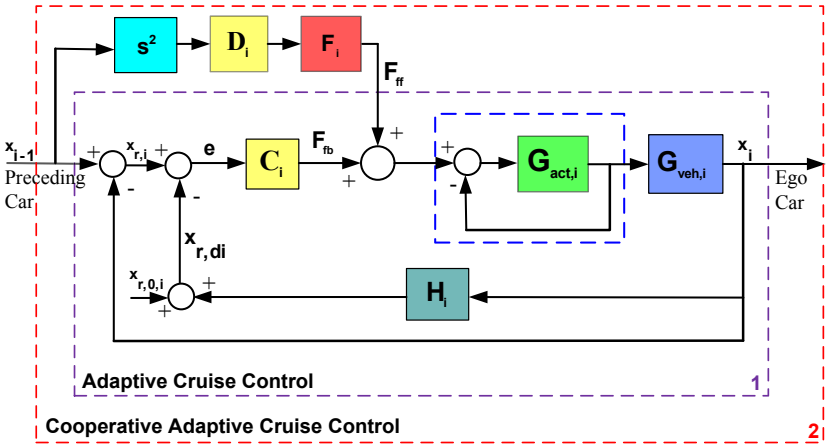


Figure 5.4: ACC and CACC Structure using in GCDC 2011

ing policy dynamics. Manual control is activated when the Smart car exceeds the maximum and minimum controlled acceleration limits in the course of urban and highway challenges. These control laws in (5.5, 5.6, 5.7) are tested in simulative scenarios for different spacing policies and then implemented in our Hardware-in-the-Loop platform [84].
The control law in (5.7) is modified considering all accelerations of front vehicles in a platoon with different controller weight factors for improving string stability performance of the platoon [85].

Table 5.1: Definition of variables in ACC and CACC Designs

Symbol	Definition
H_i	spacing dynamics ($H_i = h_{d,i}s$)
x_i	the position of the ego vehicle
x_{i-1}	the position of the preceding vehicle
e	relative distance error
$x_{r,0,i}$	desired relative distance at standstill
$x_{r,di}$	desired relative distance
D_i	communication delay dynamics
C_i	corresponding ACC feedback controller
F_i	feed-forward filter

$$f_{di} = m_i(\omega_{K,i}e_{vi} + \omega_{K,i}^2 e_{xi}) + F_i D_i \left(\sum_{j=1}^{i-1} \sigma_{d,j} a_j \right) \tag{5.8}$$

$\sigma_{d,i}$ is a priority factor of accelerations all preceding vehicles in a platoon and the controller in (5.8) gives better results when we increase the direct preceding's acceleration weight and lower the leader's acceleration weight factor. The detailed design of feed-forward filter coefficient for guaranteed-stability of highway missions is given in Chapter 6.

The real implementation of the cooperative control is first testing our controllers and defining optimal control coefficients using dynamic vehicle, sensor, and actuator models. The high level control block is responsible for the generation of the command force f_{di} for the Smart Fortwo car which is used as the input to the low level control for actuators, such as brake and throttle system. PD filter based ACC and CACC control in (5.6) and (5.7) are adjusted with tuning $\omega_{K,i}$ to the optimal value considering the time constant τ_i of actuator dynamics and spacing policy dynamics. Spacing policy defines optimal relative distance and the desired time headway between our vehicle and the preceding vehicle. Then, we validated the optimality of these estimated controller coefficients by physical tests while moving our Smart vehicle in the platoon with different scenarios.

5.3.3 Simulations of ACC and CACC Implementation

In this subsection, we are incorporating 10 identical vehicles in platoon and implement highway scenario with predefined speed profile. The dynamic model parameters K_i, time constant τ_i, ac-

tuator delay θ_i, and communication delay t_{d_i} are selected as 1, 0.8s, 0.02s, and 0.2s, respectively. First, ACC and CACC performance is tested for time headway, standstill inter-vehicle distance, and PD filter coefficient $\omega_{K,i}$, which are obtained as 0.7s, 10m, and 1, respectively. ACC and CACC algorithms in Fig. 5.4 are taken as described in (5.6), and (5.7).

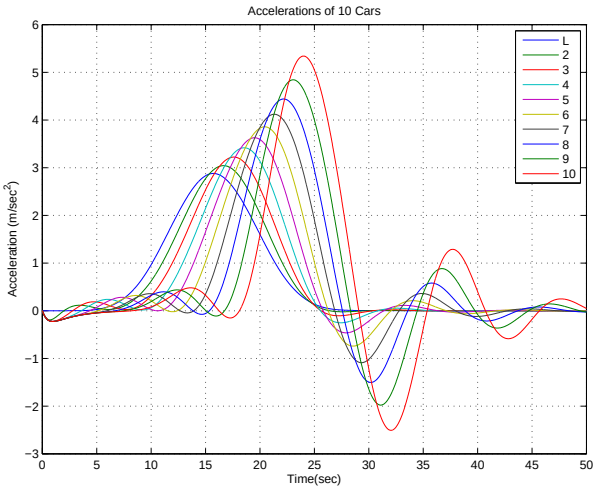


Figure 5.5: ACC Performance of 10 vehicle platoon

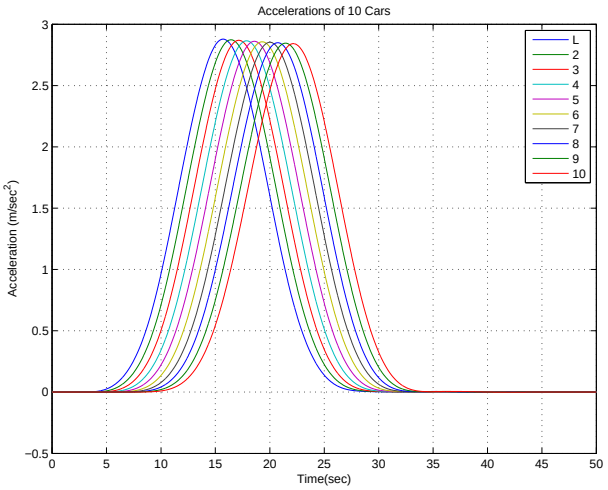


Figure 5.6: CACC Performance of 10 vehicle platoon

As it can be seen easily in Fig. 5.5 and Fig. 5.6, CACC has better transient and steady state

response than ACC model. Another remarkable aspect can be seen in the acceleration response of the vehicles. In case of ACC, it takes a considerably longer period of time before the last vehicle finally starts to accelerate and settles, compared to CACC. This can be described with the feed-forward affect. Feed-forward contribution in CACC case improves both string-stability and response time of the vehicles in the platoon.
For the simulation of CACC+ algorithm in (5.8), we are incorporating 10 identical vehicles with changing and restricting some parameters. In the new scenario, actuator delay θ_i, communication delay t_{d_i}, standstill inter-vehicle distance, and $\omega_{K,i}$ are defined as 0.2s, 2s, 5m, and 2, respectively. Then, the performance of CACC algorithm is compared with a CACC+ algorithm as depicted in Fig. 5.7 and Fig. 5.8. For the selection of feed-forward weight factor c_{di} in CACC+ algorithm, we raised the priority of direct preceding vehicles and reduce the importance of other front vehicles respectively.

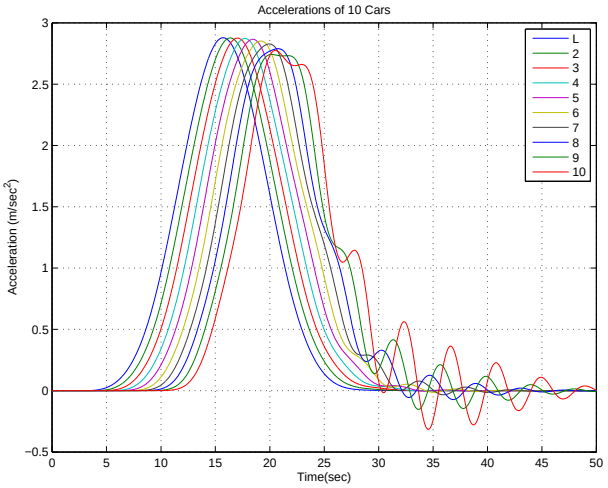


Figure 5.7: CACC Performance for higher delays

The motion of the platoon is string stable for lower standstill inter-vehicle distance, higher actuator and communication delay parameters in the CACC+ algorithm as depicted in Fig. 5.8. However, the changes in the acceleration of a lead vehicle using same dynamic parameters are amplified by the vehicles upstream in the platoon as illustrated in Fig. 5.7.

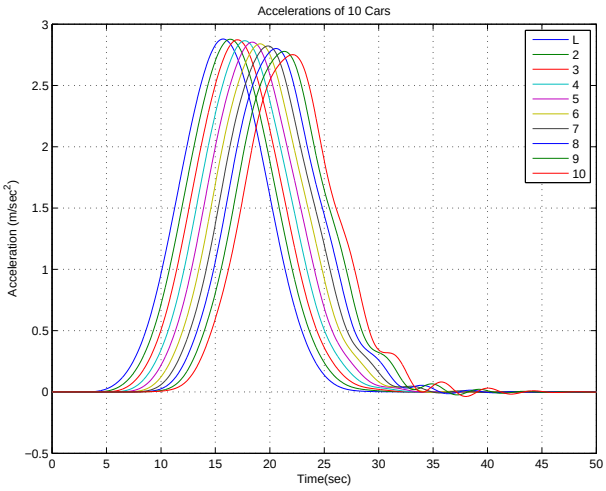


Figure 5.8: CACC+ with feed-forward coefficients for higher delays

5.4 Design and Implementation of Cooperative Driving

The Fontys Automotive Research center is provided two vehicles for testing the cooperative algorithms and improving platooning strategies. Electronic modification of these Smart cars were conducted by Fontys for autonomous longitudinal driving. There are mainly four parts of our autonomous vehicle architecture for implementing advanced CACC algorithms. These are front control panel, sensor suite, communication box, and embedded computer (xPC Target) as depicted in Fig. 5.9. The switching algorithm and mechanism was designed and added on Smart vehicle for switching between longitudinal autonomous control by an embedded computer to human driver as illustrated in Fig. 5.9.

The front control panel is for switching between human pilot and autonomous control, and emergency cases. The autonomous mode take the control of throttle, brake pedal, and gear shifting with the help of embedded computer and controller area network bus (CanBus). The stream of ego vehicle's sensory information and states is provided by CanBus interface, and other preceding vehicles' information received by wireless communication unit (ComBox) [45].The flashing lights at the top of the Smart vehicle is for warning other platoon drivers whether our current mode is autonomous or human driving. Further details of smart vehicle modification and additional sensor suite can be found in Section 2.4.3 [11].

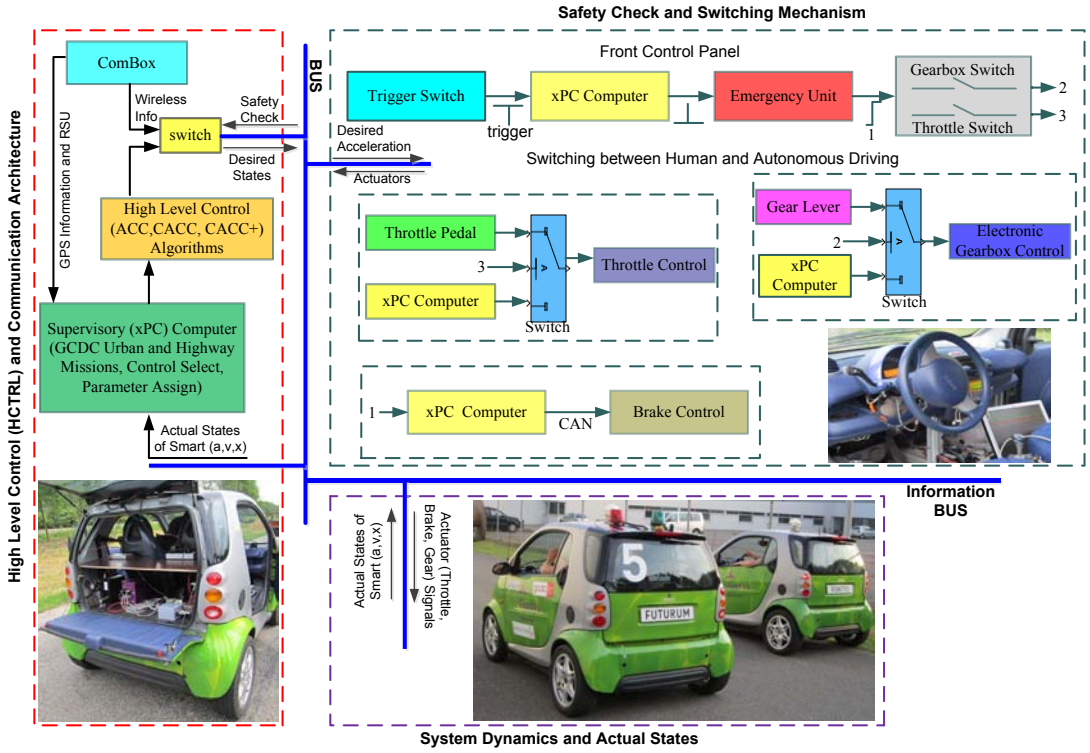


Figure 5.9: Smart vehicle's functional diagram during GCDC

5.5 Low-Level Control Implementation

Low level control system is responsible for the stability of the brake, throttle and the gearbox systems. High level control provides the necessary acceleration, which is directly correlated with the desired force. We adjust our speed and acceleration by measuring the longitudinal acceleration and comparing with the desired acceleration from high level control block. The dynamics resulting from low level control are called closed loop actuator dynamics as depicted with a dashed square in Fig. 5.4.

5.5.1 Throttle, Brake, and Gearbox Control

Many low level control experiments was conducted at the Automotive Campus at Helmond, since Smart was not allowed on public roads with a modified brake system. The maximum driving speed during these experiments was limited to 50 km/h in third gear for safety concerns.

Longitudinal acceleration measurements include high frequency noise, which was reduced by applying a digital moving average filter over 20 samples at 100 Hz sampling frequency. This filter results in some delay, which is a prominent drawback in the feedback control loop. Because of mitigating these delay term and disturbance effect, PID loop was used in order to control actuators. Controller coefficients were determined by considering the dynamics of our throttle and braking actuators.

$$u_i = k_{Di}e_{ai} + k_{Pi}e_{vi} + k_{Ii}e_{pi} \tag{5.9}$$

u_i is control signal for actuators, e_{ai}, e_{vi}, and e_{pi} represent the acceleration error, velocity error, and the position error, respectively. The feedback control performance we achieved with this type of engine is as depicted in Fig. 5.10 during one of the GCDC heats. Braking and throttle control performance are improved with the contribution of feed-forward loop as illustrated in Fig. 5.10.

While shifting gears (takes almost two seconds), there is no transmitted torque to the driven wheels, resulting in a small negative acceleration of the vehicle. Therefore, autonomous throttle control was disabled during gearshift for maintaining the throttle position and reactivated throttle control from the last known throttle pedal position. When there is no gearshift status signal perceived from gearbox, the Smart car is switched to manual mode.

At the beginning, shifting algorithm was shifted gears at a fixed engine revolutions per minute (RPM). Later, the desired vehicle acceleration is also integrated to perform downshifts when driving at relatively low engine RPM and to allow upshifts at higher engine RPM in case of high desired acceleration. This modification improves the acceleration profile. Thus, the Smart are able to use its maximal power and remain in one gear for a longer time. However, the combination of a small 600 cc engine (max. 37kW) with a six speed gearbox and relatively long shifting times is really far from ideal and desired control performance.

The desired acceleration from high level control was limited between $-4.0\ m/s^2$ and $2.5\ m/s^2$, which is one of the critical GCDC safety requirements. The control system performance shows better characteristics for negative desired accelerations, but not that good for positive acceleration demand as depicted in Fig. 5.10. The vehicle is not able to reach the desired acceleration even at full throttle (i.e. 2.7V) as in Fig. 5.10. This problem is originated from limited engine torque. Region A shows a gearshift, which is because of higher desired acceleration than measured acceleration. Region B shows an example of integrator windup, which is noticed after the GCDC challenge. The output of throttle control is saturated at 2.7V (full throttle of Smart car). However, the original unsaturated output of the integrator block is higher than this saturation limit especially for 5 seconds of full throttle period. Therefore, the throttle input could not decrease quickly after the sign of control error was changed. So, the output of the control remains at the saturation limit (i.e. full throttle) while the output of the integrator higher than the saturation limit. As a result, measured acceleration is higher than the desired acceleration and the

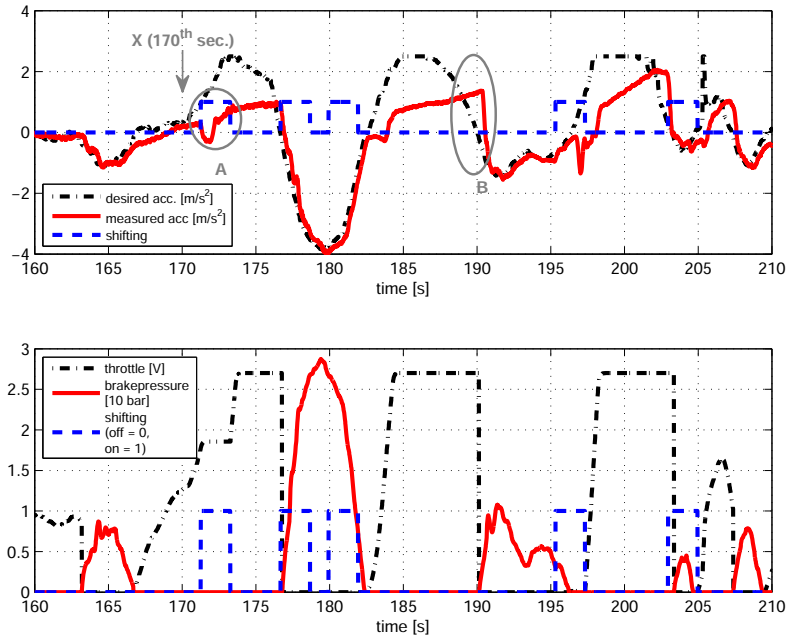


Figure 5.10: Low level control performance during one of the heats in GCDC.

Smart increase brake pressure. Consequently to avoid integral wind up problem, the integration had better to be stopped as soon as we are at full throttle and the control signal saturates.

5.6 Control Performance in Platooning

In urban scenario, vehicle teams are split into two platoons. Front and Back platoons are waiting for traffic lights turn to green and these trigger lights are communicating between each other with RSU units. At first, the back platoon is triggered with a green light. Then, vehicles start to move with keeping the conditions for a stable and safe platoon. While platoon 1 approaching, the GCDC lead vehicle begins accelerating with a signal of green light. Thus, other participants automatically accelerate and try to follow lead vehicle as well. The urban mission ends when all vehicles crosses the finish line. This scenario is for testing urban behaviors of collaborative vehicles and for evaluating interactions between separate mini platoons. The highway mission is for testing the speed regulation and string stability capability of vehicles under various rapid acceleration and deceleration periods by GCDC lead vehicle. In highway mission, GCDC lead vehicle accelerates and decelerates in different time spans. This acceleration profile (shockwave)

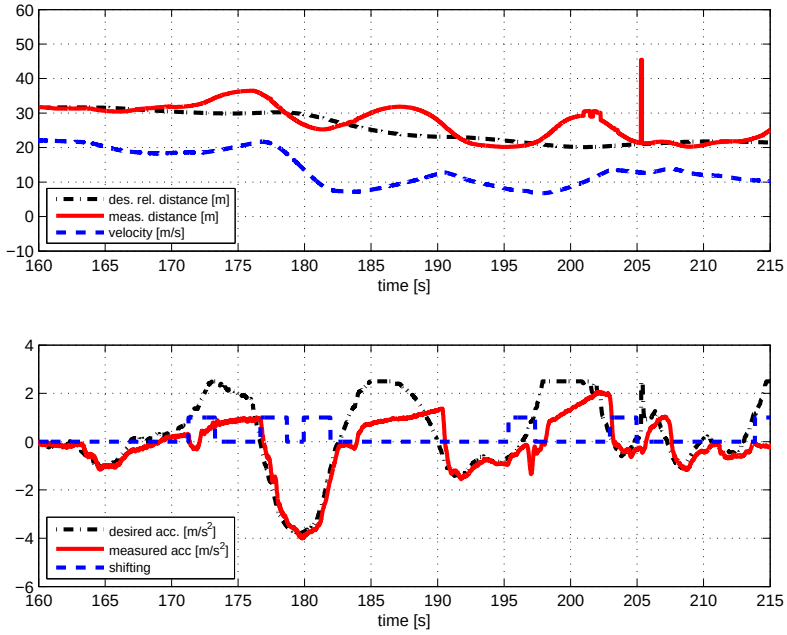


Figure 5.11: Vehicle velocity and high level control input, high level control output, and measured acceleration in GCDC heats.

is the main reason of traffic congestion problem and vehicles are evaluated with the ability of keeping string stability and safety conditions in this part. Both urban and highway scenarios as in Fig. 5.2 are expressed detailed in [47].

5.6.1 Safety Requirements

The vehicles are capable of setting their velocity up to 80km/h, keeping max. acceleration (a_{min}) and min. acceleration (a_{min}) requirements. All vehicles have to incorporate longitudinal autonomous controller, but the steering is handled by human driver. Furthermore, each driver can interrupt autonomous mode with pushing emergency button and autonomous mode must maintain safety distance between preceding vehicle. Standstill distance, time headway, max. acceleration, and minimum acceleration are obtained as 10m, 0.6s, $-4.5m/s^2$, and $2m/s^2$ during all heats in GCDC, respectively [47].

5.6.2 Evaluation Criteria

The participant teams are split into two streams (Team A and Team B), positioned in two lanes during GCDC competition. Every team is replaced in the platoon for testing capability in crucial positions at the beginning of each heat. The main conditions for the evaluation of the vehicles' performances can be summarized as the platoon length L_p while last vehicle passes the finish line in the urban scenario, the maximum gap length $L_{g,max}$ reached, the platoon length variation v_{L_p}, and string stability during the highway part of the scenario. The vehicles' acceleration shockwave attenuation in the upstream is calculated with string stability formulation. This is estimated with calculating infinity norm of the fourier transform division of i'th vehicle's acceleration $A_i(j\omega)$ to the lead vehicle's acceleration $A_1(j\omega)$ (The condition: $\left\| \frac{A_i(j\omega)}{A_1(j\omega)} \right\|_{H_\infty} \leqslant 1$) [47].
The detail of mathematical calculations of upper terms is given in Chap. 6. The actual calculation of the performance criteria is executed in discrete time and discrete frequency domain with the help of video based monitoring measurements.

5.6.3 Performance of Platooning Tests

Fig. 5.11 demonstrates velocity and spacing profile for sixty seconds of the heat during GCDC. Additionally, this figure includes acceleration profile of the vehicle as depicted in Fig. 5.10. At the beginning, the desired relative distance and measured relative distance are approximately overlapped. After 170^{th} seconds, the difference between desired relative distance and measured relative distance starts to increase, because of a positive acceleration in the motion of preceding vehicle (as pointed at X). At that time, the gear is shifted up, so throttle control is disabled for two seconds. After the gearshift operation, throttle control is re-enabled. Then, the output of throttle control goes up to full throttle and remains at the maximum for the following three seconds as it can be seen in Fig. 5.10. Through this time period, the relative distance error and the velocity of Smart increase. Braking is necessary to slow the Smart down and decrease the relative distance. As it can be seen in Fig. 5.10, the predecessor is braking at this time, so the Smart has to brake even harder. Furthermore, the smart performs gear shifting when it brakes. After that braking, speed starts to increase once again, but the Smart cannot be able to reach the desired acceleration even at full throttle. So, this acceleration error increase the difference between desired relative distance and measured relative distance.

These error dynamics is strongly related with the slow response of the actuators and integrator windup in throttle control. In addition to this, the performance corruptions are because of limited velocity, limited engine torque, gearshifting and limited acceleration of the vehicle. The throttle output cannot be decreased immediately when the acceleration error changes sign due to integrator wind-up.

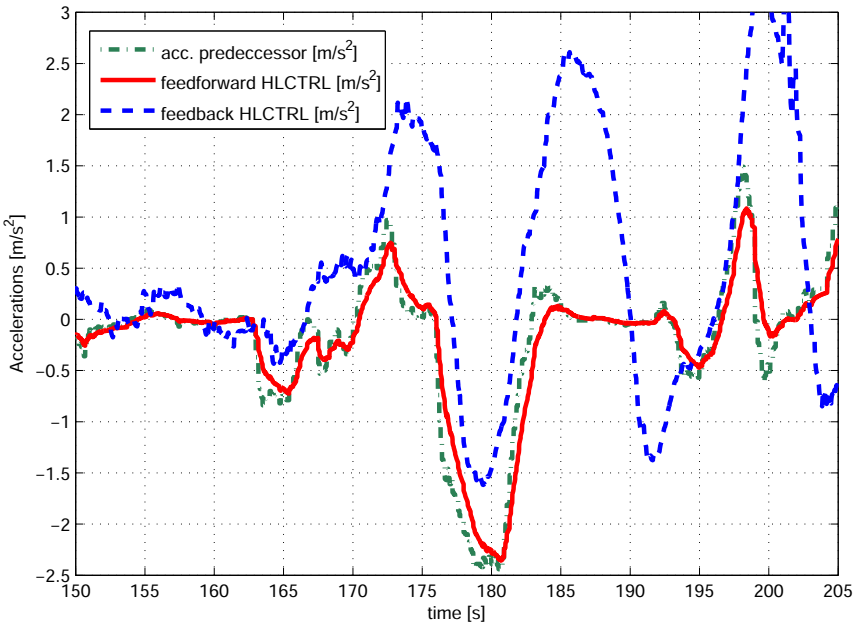


Figure 5.12: Acceleration of predecessor and ego vehicle with feed-back or feed-forward contribution

The acceleration of the predecessor, the ego vehicle's acceleration with feed-forward filter and the ego vehicle's acceleration with feedback controller are depicted in Fig. 5.12. The total acceleration tracking affect of feed-forward filter and feedback controller is as depicted in Fig. 5.10. In Section 5.3, the actuator dynamics (i.e. low level control) are modeled as a first order system. Feed-forward filter also uses this low-level actuator dynamics and the spacing policy dynamics for calculating a feed-forward acceleration, which contributes a relative distance and relative velocity tracking.

These figures do not directly point out the benefit of CACC to a standard ACC controller, but demonstrates one of the real heats of GCDC with CACC controller. However, early tests conducted with both Smarts driving behind each other up to $50\ km/h$ maximum speed demonstrates the performance difference very clearly. ACC is a feedback only controller, whose inputs are relative distance and relative velcotiy error. But, CACC controller includes feed-forward acceleration of the preceding vehicle as described in Section 5.3.2. In ACC control, the relative distance error demonstrates highly oscillating behaviour, which is damped with the additional feed-forward filtered of the predecessor's acceleration in CACC control structure. In addition, Fig. 5.6.3 is depicted a shot from one of the GCDC heats.

In conclusion, the implemented high level and low level control structure includes an oscillating

Figure 5.13: One of the GCDC heats at A270 highway in Netherlands.

characteristic, which disrupts the most important string stability profile of the motion. Disturbances from the predecessor are being amplified, when it is supposed to be reduced. The primary reasons for these oscillations are the slow response of throttle control and integrator windup. The bandwidth of throttle control (i.e. low level control) is not high enough compared with the bandwidth of high level control dynamics. Therefore, the orders from high level control cannot be fulfilled as fast as required by low level control. Furthermore, the small engine in the Smart is not strong enough to catch desired accelerations over approximately $1.5\ m/s^2$ in third gear and higher. So, we plan to improve low level control of actuators, the control characteristics with using Kalman filtering when processing the sensory information, and implementing different high level control structures instead of PD based CACC structures. Moreover, we plan to employ different car models instead of current 10 year old Smart Fortwo car.

5.7 Conclusion

In this chapter, first we described our road vehicle design and its sensory modification for better CACC implementation on Highway Platoons. Then, high level and low level control strategies are also described for better autonomous longitudinal control of the road vehicles. Furthermore, interaction of low level control dynamics and CACC in real driving tests are evaluated using specific requirements. The shortcomings and accident risks originated from human driver inter-

ference are improved by means of ACC controllers and wireless negotiation between vehicles. These improvements can be achieved with simple technological modifications on modern commercial cars and activating RSU infrastructure for administering the traffic and wireless communication [47].
Next, the validation of CACC controllers during real driving tests and simulation results of modifed CACC model are given as the evaluation of the recent CACC approaches with different topologies. The shortcomings and accident risks arisen from human driver interference are improved by means of CACC controllers. In the next chapter, the focus is giving the formal definitions of constant and variable inter-vehicle spacing strategies and analysis of the collaborative driving algorithms with guaranteed stability during platooning.

Chapter 6

Proposed Collaborative Driving Strategies for Stability Guaranteed Highway Missions

6.1 Modeling of One Dimensional Formation

In this chapter, we give a better insight to the collaborative driving techniques at highways detailing the general problem of one-dimensional formation design of multi-vehicle platoons. Similar to the three-dimensional cohesive formations in Chapter 3, the focus is to keep the inter-vehicle distances in the desired bounded region all the time while introducing aggressive acceleration/deceleration, depending on the leader vehicle of the platoon. In the *on-line trajectory generation algorithm* described in Section 3.3, all agents determine the desired positions according to the location of leader, initial followers, and desired inter-agent distances for cohesive motion in three dimensions; nevertheless, the calculation of the desired positions for each agent in the formation of vehicles at highways is made via spacing policy dynamics with respect to direct preceding car as partially described in Chapter 5 [32, 68, 76].

Here, we specifically consider one-dimensional formation preserving the inter-agent distances of each agent-pair in a platoon of vehicles at highways. For the better understanding of this particular networked system, we consider decentralized control of a non-hierarchical platoons while a specific acceleration/deceleration profile introduced by independent leader agent as illustrated in Fig. 6.1. Additionally, we assume that assumptions A1 and A2 in Section 3.2.3 are valid for all vehicles, but assumption A3 is only valid for leader vehicle in platooning.

For the sake of simplicity, first we ignore the actuator dynamics, and transmission-delays and design each-agent A_i of the one-dimensional highway platoon with second order translational

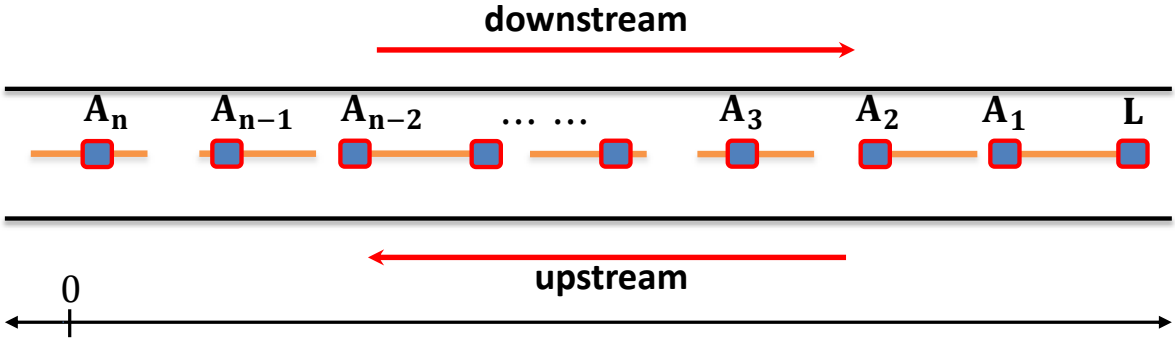


Figure 6.1: Traffic flow model of one-dimensional vehicle platoon at highways: First agent is leader vehicle, which defines the acceleration and deceleration profile of the platoon

dynamics, which are:

$$\dot{x}_i = v_i, \quad \forall t \geq 0 \tag{6.1}$$

$$m_i \dot{v}_i = u_i \tag{6.2}$$

where x_i, v_i, m_i, and u_i represent the position, velocity, mass, and control force of the i^{th} vehicle A_i, respectively. This control force is generated for each agent by individual Lyapunov-based trajectory tracking controllers. The main focus is designing a string-stable controller in order to keep inter-agent distance with the preceding vehicle while rapid acceleration/deceleration actions are introduced in the downstream traffic. Each agent A_i is assumed to have a control law of the form

$$u_i = f_i(e_{x_i}(t), e_{v_i}(t), x_{r,di}), \quad x_{r,di} > 0 \tag{6.3}$$

where e_{x_i}, e_{v_i}, and $x_{r,di}$ are position error, velocity error, and desired constant inter-agent distance, respectively, similar to [32].

6.1.1 Lyapunov-Based Decentralized Control Design

In this subsection, we present distributed control design considering translational dynamics given in (6.1) and (6.2) for non-hierarchical platoons, which are using same type of controller (6.3) for each agent. Consider the Lyapunov function

$$V_{L_i} = \frac{1}{2}(k_i^2 \|e_{x_i}\|^2 + \|e_{v_i}\|^2) \tag{6.4}$$

The error functions in the Lyapunov function are defined as

$$\begin{aligned} e_{x_i} &= x_i - x_{id} && (6.5)\\ x_{id} &= x_{i-1} - x_{r,di} && (6.6)\\ v_{id} &= -k_i e_{x_i} && (6.7)\\ e_{v_i} &= v_i - v_{id} && (6.8) \end{aligned}$$

where x_{i-1} and v_{id} are current position of the predecessor and desired velocity for the ego vehicle, respectively. The design of Lyapunov-based low level controllers are to generate the control force $u_i(t)$ based on forcing V_{L_i} in (6.4) to decay to zero.

$$\begin{aligned} \dot{e}_{x_i} &= v_i \\ &= -k_i e_{x_i} + e_{v_i} && (6.9) \end{aligned}$$

$$\begin{aligned} \dot{e}_{v_i} &= \dot{v}_i - \dot{v}_{id} \\ &= a_i + k_i \dot{e}_{x_i} \\ &= a_i - k_i^2 e_{x_i} + k_i e_{v_i} && (6.10) \end{aligned}$$

where $a_i = u_i/m_i$. The derivative of Lyapunov function in (6.4)

$$\begin{aligned} \dot{V}_{L_i} &= k_i^2 e_{x_i}\dot{e}_{x_i} + e_{v_i}\dot{e}_{v_i} \\ &= (-k_i^3 e_{x_i}^2 + k_i^2 e_{x_i} e_{v_i}) + e_{v_i}(a_i - k_i^2 e_{x_i} + k_i e_{v_i}) \\ &= -k_i^3 e_{x_i}^2 + e_{v_i} a_i + k_i e_{v_i}^2 && (6.11) \end{aligned}$$

To have $\dot{V}_{L_i} = -2k_i V_{L_i}$, we choose

$$u_i = m a_i = -2 m_i k_i e_{v_i} \tag{6.12}$$

For the string stability of our platoon driving in the upstream traffic, we assume desired velocity for the ego-vehicle as the actual velocity of the predecessor. As described in Chapter 5, we are able to measure intervehicle distance and velocity difference with our preceding vehicle by means of lidar sensor. However, we can guarantee the stability of the platoon without using intervehicle spacing error. Since we are aware of our current velocity, we can calculate the value of k_i using accurate lidar measurements. The Lyapunov based non-cooperative adaptive cruise control design proves the importance of intervehicle velocity error over intervehicle spacing error for string stability of the vehicle platoon, when we ignore the actuator dynamics. As a result, our novel control approach is simpler for guaranteed stability than those provided in [56–58].

6.1.2 Simulations for Lyapunov Based Platooning

We are incorporating 7 identical vehicles with the dynamics given in (6.1) and (6.2). Then, we implement the platooning scenario with predefined speed profile. The dynamic model parameter K_i, maximum, and minimum accelerations are selected as 1, $4.2m/s^2$, and $-8.4m/s^2$, respectively. Performance of the Lyapunov based cruise control (6.12) is tested for $x_{r,di}$ desired standstill inter-vehicle distance, and k_i control coefficient are obtained as 8m, and 1, respectively.

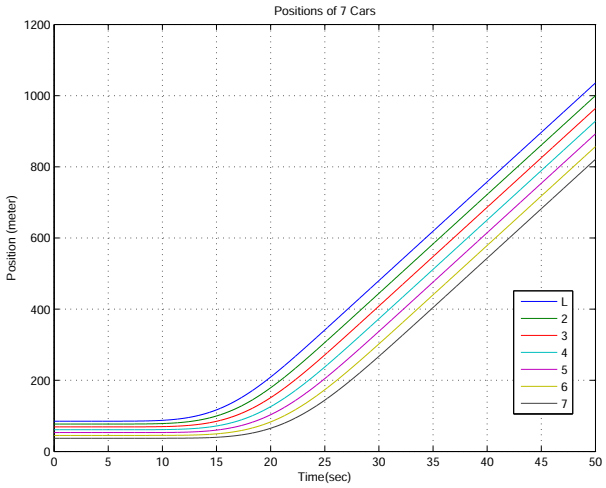


Figure 6.2: Positions for 7 vehicles platoon with Lyapunov based cruise control

As it can be seen easily in Fig. 6.2, Fig. 6.3, and Fig. 6.4, Lyapunov based cruise control of vehicles modeled with second-order translational dynamics provides a string stable response only with considering intervehicle velocities. Another remarkable aspect of our highway platooning is being always string-stable for constant intervehicle spacing without any feed-forward contribution, if we ignore actuator dynamics of the vehicles in the platoon.

6.2 Formal Problem Definition for Highway Platooning

In this section, we analyze the platooning problem at highways considering constant and velocity dependent inter-vehicle spacing, and various communication strategies. We first consider n consecutive vehicles $A_1, A_2, ..., A_n$ (where $m \geq 3$) on one-dimensional axis as illustrated in Fig. 6.1 with the full longitudinal dynamics in (5.1). Each vehicle has front lidar or radar to measure

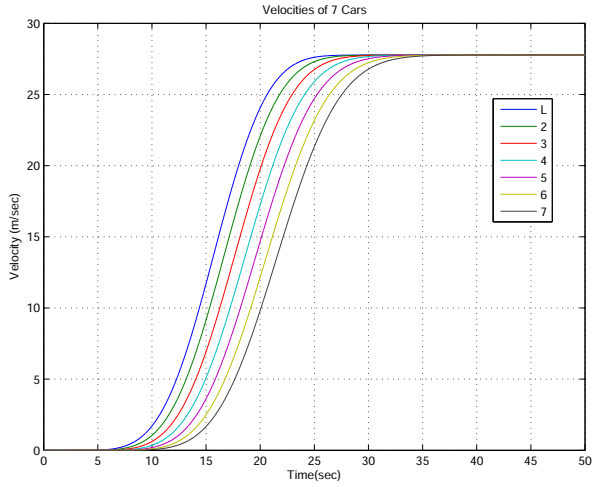


Figure 6.3: Velocities for 7 vehicles platoon with Lyapunov based cruise control

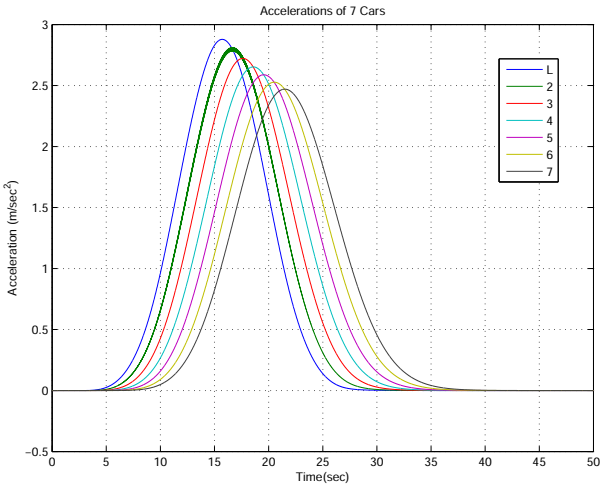


Figure 6.4: Accelerations for 7 vehicles platoon with Lyapunov based cruise control

its relative position and velocity with the direct preceding vehicle. Additionally, each agent is able to transmit its vehicle states (such as instant position, velocity, and acceleration) and receive those off all other members of the platoon. The traffic is moving towards the downstream direction and indices $i \in \{1, \ldots, n\}$ of agents A_i increase in the upstream direction. This means considering initial position of each agent is increased towards downstream direction in the positive x-axis. We intend to keep the order of vehicles same and attenuate the vehicles' acceleration

shockwave in the upstream direction according to the evaluation metrics described in Section 6.2.1 for the string stability of our platoon driving [57, 58, 68, 76].

6.2.1 Evaluation Criteria

The vehicle performance can be assessed with four metrics, such as the platoon length L_p while the last vehicle passes the finish line, the maximum gap length $L_{g,max}$ reached in the driving cycle, the platoon length variation v_{L_p}, and the string stability during highway driving [47].
To illustrate, the real-life calculation of the following performance criteria is made in discrete time with the help of video based monitoring measurements and collected wireless data. The mathematical definitions of evaluation metrics are given as:

1. The platoon length L_p and total gap length L_g

$$L_p(t) = x_{lead}(t) - x_f, \tag{6.13}$$

$$L_g(t) = L_p(t) - \sum_{i=1}^{m} L_i \tag{6.14}$$

2. The maximum gap length $L_{g,max}$

$$L_{g,max} = \max_{t\in[t_s,t_f]} \left(L_p(t) - \sum_{i=2}^{m} L_i \right) \tag{6.15}$$

3. Platoon length variation v_{L_p} during highway mission is described as

$$v_{L_p} = \frac{1}{t_s - t_f} \int_{t_s}^{t_f} \left(L_p(t) - L_s(t)\right)^2 dt, \tag{6.16}$$

$$L_s = \sum_{i=2}^{m} L_i + (m-1)(d_0 + h v_{lead}(t)) \tag{6.17}$$

where t_s, t_f, d_0, h, v_{lead}, and $L_s(t)$ are defined as start time, finish time, standstill distance, headway time, and the velocity of lead vehicle, and safety platoon length of the highway driving, respectively.

4. The evaluation of vehicle state (position, velocity, or acceleration) shockwave attenuation in the upstream is calculated with string stability formulation. This is estimated with calculating infinity norm of the Fourier transform division of ego vehicle's state $\Gamma_i(j\omega)$ to the leader's state $\Gamma_L(j\omega)$.

$$\left\| \frac{\Gamma_i(j\omega)}{\Gamma_L(j\omega)} \right\|_\infty \leqslant 1 \tag{6.18}$$

These metrics are main criteria for evaluating the stability characteristics of vehicles in the highway platoons. The aim is to minimize all these four conditions at the same time. For better match with these stability criteria, we consider spacing error propagation in the upstream platoon in the following spacing strategies and control design.

6.2.2 Spacing Requirements and Strategies

Spacing policy dynamics become so prominent in defining achievable traffic capacity and string stability. We consider two different spacing strategies; (i) constant and (ii) velocity dependent inter-vehicle spacing policies. Therefore, adaptive control laws are compared using two different spacing policy dynamics. In this study, each vehicle is deemed having identical dynamics and actuator capacity in the platoon.

Constant Inter-Vehicle Spacing Policy

Consider a platoon of $n+1$ vehicles lined up at a highway as shown in Fig 6.1. The vehicle order is increased in the upstream platoon. We denote the first vehicle as leader vehicle tagged with notation L and last vehicle as agent A_n. In the constant spacing, we define the desired inter-vehicle spacing as a constant separation distance, which is independent from the velocity of the ego vehicle. Since the tracking condition being strict, all platoon members need accurate sensory and wireless information in order to match its velocity better with its predecessor. Besides, we have to optimize desired spacing not to exceed the vehicles' acceleration/deceleration capacity in heterogeneous platoons [68, 76].
Define the measured inter-vehicle spacing of agent A_i as,

$$\varepsilon_i = x_{i-1} - x_i - d_{i-1} \tag{6.19}$$

where d_{i-1} is the length of the preceding vehicle. Here, x_i is considered as the front position of vehicle i. Then, the spacing error e_i of agent A_i for constant spacing policy is defined as,

$$e_i = x_{i-1} - x_i - x_{r,di} \tag{6.20}$$

where $x_{r,di}$ is the desired constant distance for inter-vehicle spacing, which also includes length of the preceding vehicle.

Velocity Dependent Inter-Vehicle Spacing Policy

We add constant time-gap on top of desired constant spacing distance in the velocity dependent spacing policy. It means the desired spacing is changing linearly with the controlled vehicle velocity. Since the control implementation with constant spacing strategy is not that robust for changing acceleration and deceleration profiles, we increase the flexibility of the tracking adding controlled vehicle's velocity and modifying (6.20) as follows:

$$e_i = x_{i-1} - x_i - x_{r,di} \tag{6.21}$$

$$x_{r,di}(t) = x_{r,0,i} + h_{di}\dot{x}_i(t) \tag{6.22}$$

where $x_{r,0,i}$ and h_{di} are minimum relative distance at standstill and desired time headway, respectively. The time headway h_{di} denotes the time for vehicle A_i to reach the position of its direct preceding vehicle A_{i-1} with a constant speed when the A_{i-1} is at standstill [56–58, 68].
Despite having higher advantage for string stable control implementation at highway platoons, the constant time headway policy reduces the traffic throughput at higher speeds. In order to overcome these deficiencies, some other spacing strategies can be recommended, such as hybrid strategies, which are combining constant spacing and variable spacing methods, or variable time headway, which decreases slowly as the controlled vehicle's velocity increases [76, 88]. However, we do not consider these additional strategies in the following control laws.

6.3 Control Design

In our control design, we consider the vehicular system and error dynamics as given in (5.1)−(5.3) and spacing policy dynamics as described in Section 6.2.2. Our control approaches can be split into two, one for non-cooperative driving and the other for cooperative driving according to the dependent on sensory data and wireless data transmission with other vehicles in the platoon. In both control approaches, we are dealing with individual vehicle stability, string stability of the whole platoon, and zero steady-state spacing error. In the following control design study, we consider 7 identical vehicles equipped with the same sensory set driving successively on a single lane. Since we have actuator time delay and wireless transmission delay, we ignore additional pure actuator delay t_{d_i} parameter in the identified model (5.1). For the sake of simplicity, we assume the mass term m_i as being 1 in system model G_i, feedback control law C_i, and feedforward

control law F_i. Consequently, the ultimate model transfer functions are obtained as:

$$G_i(s) = \frac{K_i}{s^2(\tau_i s + 1)}, \qquad for\ i > 1 \tag{6.23}$$

$$C_i(s) = \frac{U_i(s)}{E_i(s)} = K_{d_i} s + K_{p_i} = \omega_{K,i}(s + \omega_{K,i}) \tag{6.24}$$

$$H_i(s) = 1 + h_{di} s. \tag{6.25}$$

where K_{v_i}, K_{p_i}, and $E_i(s)$ are proportional, derivative control parameters, and Laplace transform of inter-vehicle spacing error, respectively. The feedback controller is designed as a PD filter with the bandwidth $\omega_{K,i}\ rad/s$ as provided in Chapter 5. We ensure the stability of the driving with the selected PD bandwidth using root-locus method and frequency based Bode analysis graphs. In addition to the feedback controller $C_i(s)$, we consider the actuator delay τ_i and time headway h_{di} of spacing policy dynamics in the design of feed-forward filter F_i. Besides, the distance based sampled σ_i priority factor is designed using low-pass filter characteristics and is given in Section (6.4) with details. Finally, we used first order pade-approximation for modeling communication delay D_i in order to analyze our highway platoon in frequency domain [68, 76].

6.3.1 Non Cooperative Driving

In non-cooperative control approach, ego vehicle does not communicate or cooperate with outer vehicles. The only equipment we have is onboard sensors and radar data to get ego vehicle's attitude, and spacing with the preceding vehicle. For the string stability, we investigate the spacing error propagation in the upstream platoon. To this end, the spacing error model and string stability model for spacing error and vehicle position are identified as follows:

$$E_i(s) = X_{i-1}(s) - X_{r,0,i}(s) - H_i(s)X_i(s)$$

$$E_i(s) = C_i(s)G_i(s)E_{i-1}(s) - C_i(s)G_i(s)H_i(s)E_i(s) - \frac{1}{s}X_{r,0,i}$$

$$SS_{E_i} = \frac{E_i(s)}{E_{i-1}(s)} = \frac{C_i(s)G_i(s)}{1 + C_i(s)G_i(s)H_i(s)} \tag{6.26}$$

$$SS_{X_i} = \frac{X_i(s)}{X_{i-1}(s)} = SS_{E_i} \tag{6.27}$$

The string stability functions of displacement error and position are clearly obtained from (6.21)−(6.25) as the driving relation of ego vehicle with its direct preceding. The control model for non co-operative driving is as depicted in Fig. 6.5, when feed-forward control law F_i is equal to 0.

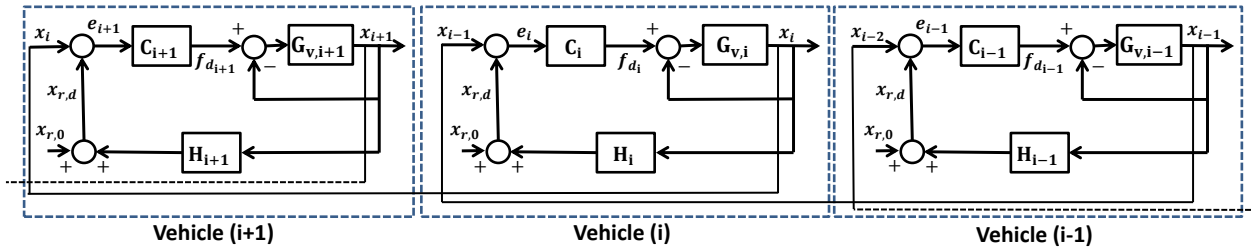


Figure 6.5: Structure of non cooperative control with three vehicles

6.3.2 Cooperative Driving

In the cooperative driving mode, we are capable of communicating with the targeted preceding vehicles in the platoon. This ability brings us a convenience in the anticipation of the desired control signal. Here identify the string stability models of the following three different cooperative control models. Then, the formal string stability analysis of these three driving modes for constant and velocity dependent inter-vehicle spacing are provided in Section 6.5. One of the most important requirements in the following control approaches is defining the optimal value of the feed-forward coefficient as $0 < \sigma_i \leq 1$. Additionally, the string stability functions of displacement error and position are defined with the help of spacing policies and models given in (6.21)−(6.25), (6.35), and (6.36). The different communication topologies of cooperative driving modes can be clearly seen in Fig. 6.6, Fig. 6.7, and Fig. 6.8.

Preceding Vehicle Only

In the preceding vehicle only control, we only consider the wireless acceleration information from the direct preceding vehicle and the feed-forward coefficient σ_i is obtained as 1. The control structure of preceding vehicle only cooperative model can be seen in Fig. 6.6. The spacing error model, string stability functions of spacing error and positions are given as follows:

$$\begin{aligned} E_i(s) &= X_{i-1}(s) - X_{r,0,i}(s) - H_i(s)X_i(s) \\ E_i(s) &= G_i(s)(C_i(s)E_{i-1}(s) + s^2 D_i(s)F_i(s)X_{i-2}(s)) - \\ &\quad H_i(s)(G_i(s)(C_i(s)E_i(s) + s^2 D_i(s)F_i(s)X_{i-1}(s)) - \frac{1}{s}X_{r,0,i} \\ \frac{E_i(s)}{E_{i-1}(s)} &= \frac{C_i(s)G_i(s)(1 - s^2 H_i(s)D_i(s)F_i(s)G_i(s))}{1 + C_i(s)G_i(s)H_i(s)} \qquad (6.28) \\ \frac{X_i(s)}{X_{i-1}(s)} &= \frac{G_i(s)(C_i(s) + s^2 D_i(s)F_i(s))}{1 + C_i(s)G_i(s)H_i(s)} \qquad (6.29) \end{aligned}$$

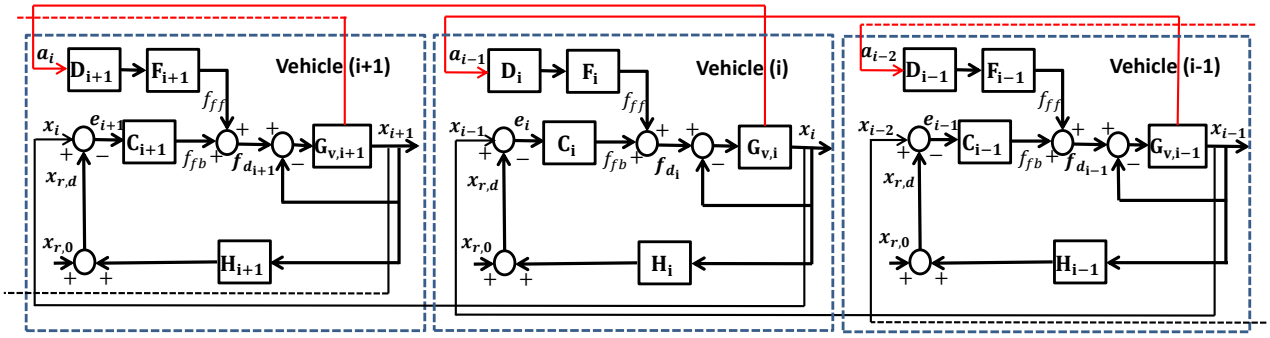


Figure 6.6: Structure of cooperative preceding vehicle only control

Preceding Vehicle and Leader

In this control model, the ego vehicle communicates with preceding and leader vehicles and receive their current acceleration for the better estimation of desired vehicle acceleration for the actuator model. The feed-forward coefficients $\sigma_{i,1}$ and $\sigma_{i,L}$ are obtained as $1/2$. The spacing error model, string stability functions of spacing error and positions are given as follows:

$$\begin{aligned} E_i(s) &= G_i(s)(C_i(s)E_{i-1}(s) + s^2 D_i(s)(F_{i,2}(s)X_{i-2}(s) + F_{i,L}(s)X_L(s))) - \\ &\quad H_i(s)(G_i(s)(C_i(s)E_i(s) + s^2 D_i(s)(F_{i,1}(s)X_{i-1}(s) + F_{i,L}(s)X_L(s))) - \frac{1}{s}X_{r,0,i} \\ \frac{E_i(s)}{E_{i-1}(s)} &= \frac{C_i(s)G_i(s)(1 - s^2 H_i(s)D_i(s)F_{i,1}(s)G_i(s))}{1 + C_i(s)G_i(s)H_i(s)} \qquad (6.30) \\ \frac{X_i(s)}{X_{i-1}(s)} &= \frac{G_i(s)(C_i(s) + s^2 D_i(s)F_{i,1}(s))}{1 + C_i(s)G_i(s)H_i(s)} \qquad (6.31) \end{aligned}$$

All Preceding Vehicles

In the all preceding vehicles model, we consider all instant acceleration information of the preceding vehicles for the estimation of ego vehicle's acceleration. The detailed description of

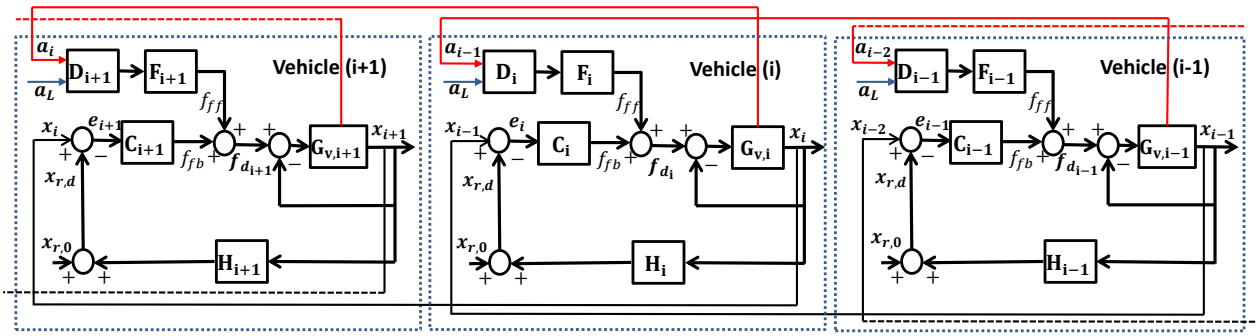


Figure 6.7: Structure of cooperative preceding vehicle and leader control

feed-forward coefficients σ_i is given in Section 6.4. The spacing error model, string stability functions of spacing error and positions are given as follows:

$$\begin{aligned} E_i(s) &= G_i(s)(C_i(s)E_{i-1}(s) + s^2 D_i(s)(F_{i,2}(s)X_{i-2}(s) + F_{i,3}(s)X_{i-3}(s) + \cdots \\ &\quad + F_{i,L}(s)X_L(s))) - H_i(s)(G_i(s)(C_i(s)E_i(s) + s^2 D_i(s)(F_{i,1}(s)X_{i-1}(s) \\ &\quad + F_{i,2}(s)X_{i-2}(s) + \cdots + F_{i,L}(s)X_L(s))) - \frac{1}{s}X_{r,0,i} \\ \frac{E_i(s)}{E_{i-1}(s)} &= \frac{C_i(s)G_i(s)(1 - s^2 H_i(s)D_i(s)F_{i,1}(s)G_i(s))}{1 + C_i(s)G_i(s)H_i(s)} \qquad (6.32) \\ \frac{X_i(s)}{X_{i-1}(s)} &= \frac{G_i(s)(C_i(s) + s^2 D_i(s)F_{i,1}(s))}{1 + C_i(s)G_i(s)H_i(s)} \qquad (6.33) \end{aligned}$$

As it can be seen clearly, all spacing errors, and string stability functions in different cooperative modes are obtained differently based on the feed-forward coefficient σ_i and existence of various acceleration data from wireless channels.

6.4 Inter-Vehicle Distance Based Priority Filtering

As generally described in Section 5.3 and mentioned in (5.8), the modified CACC controller is given with the following formulation:

$$f_{di} = m_i \left(k_{di} e_{vi} + k_{pi} e_{xi}\right) + F_i D_i \sigma_i \qquad (6.34)$$

Thus, the modified CACC controller considers velocity and spacing error with direct preceding vehicle, and the accelerations of all preceding vehicles related to the inter-vehicle distances.

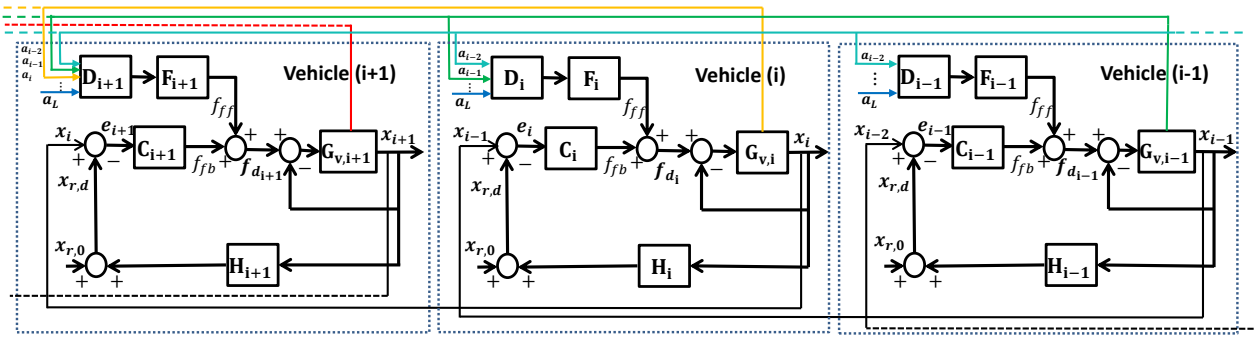


Figure 6.8: Structure of cooperative all preceding vehicles control

In addition, the σ_i coefficient also includes the acceleration of related preceding vehicle. As a result, we focus on redesigning an optimal filter F_i for smoother driving under aggressive accelerations using the wireless state information of direct preceding vehicle and leader or all preceding vehicles as illustrated in Fig. 7.2. The wireless communication delay $D_i(s)$, feedforward filter F_i, and distance based sampled priority coefficient σ_i can be given as:

$$D_i(s) = e^{-\theta_i s} \approx \frac{-\frac{\theta_i}{2}s + 1}{\frac{\theta_i}{2}s + 1}, \qquad for\ i > 1, \tag{6.35}$$

$$F_i(s) = m_i \frac{\tau_i s + 1}{1 + h_{di} s}, \quad for\ i > 1, \tag{6.36}$$

$$\sigma_i = \sum_{j=1}^{n-1} a_j \left\{ 1 + \left(\frac{\lambda(n-j)x_{r,0}}{\omega_f} \right)^2 \right\}^{-\frac{1}{2}} \tag{6.37}$$

m_i, θ_i, and a_j are mass of ego vehicle, wireless transport delay, and vehicle's acceleration, respectively [58, 85]. λ is used for determining the smoothness of low-pass filter type coefficient as illustrated in Fig 6.9. Filtering break frequency ω_f is selected with respect to frequency response dynamics of the vehicles in the platoon and the effect of ω_f on σ_i coefficient can be seen in Fig. 6.10. We optimized the selection of feedforward coefficients from the preceding vehicles with a descending order in our previous studies [11]. Therefore, the normalized σ_i wireless state coefficient is designed to increase the importance of the closer vehicles, but lower the priority of the farther vehicles in the preceding part of our platoon. The model of the priority multiplier in (6.37) is similar to the magnitude response characteristics of standard low-pass filter considering intervehicle distances instead of frequency.

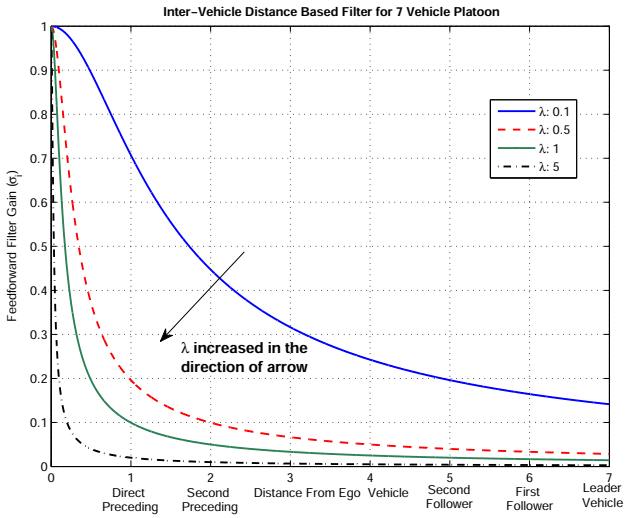


Figure 6.9: The inter-vehicle distance based filter coefficient depends on different λ values in 7 vehicle platoon

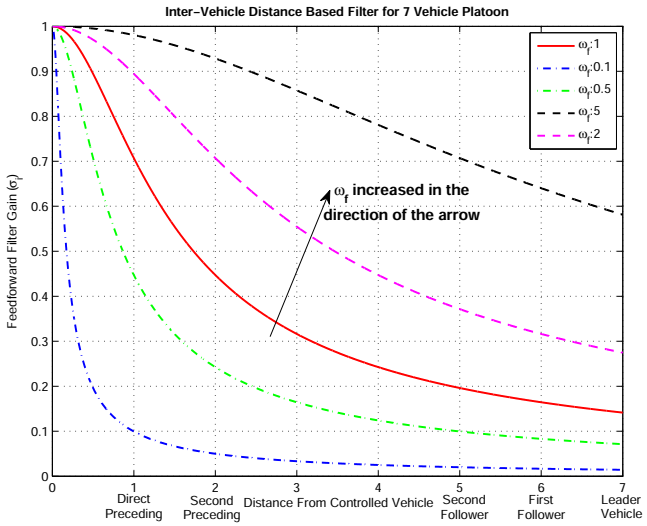


Figure 6.10: The inter-vehicle distance based filter coefficient depends on different ω_f values in 7 vehicle platoon

6.4.1 Frequency Domain Analysis of Priority Filtering

In this subsection, we discuss the characteristics of the priority filter employing 7 identical D Class Sedan vehicles in platoon and implement highway scenario with a predefined speed profile. The model parameters of D Class Sedan type vehicle is given in Chapter 2. For the simulation purposes, $x_{r,0,i}$, h_{di}, $\omega_{K,i}$ parameters, and priority sampling parameters λ, ω_f are selected as $5m$, $0.5s$, 3, and 1.6, 1, respectively. All other parameters are given in Table 2.1. Additionally, weak string stability condition is considered for the positions ($SS_X i$) of the vehicles in the platoon.

As it can be seen in Fig. 6.11, the Bode magnitude plots of conventional CACC (preceding

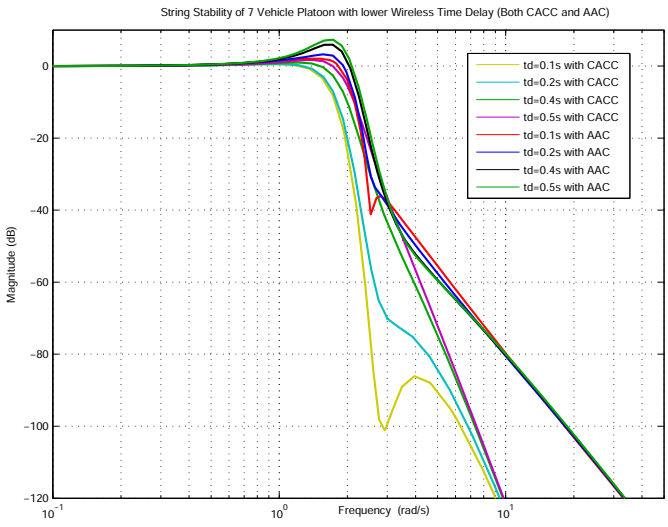


Figure 6.11: String stability of preceding vehicle only (CACC) and all preceding vehicles (AAC) algorithms with lower wireless time delays (between 0.1sec and 0.5sec)

vehicle only) and Advisory Adaptive Control (all preceding vehicles) both give stronger string stable characteristics for lower wireless time delays, but the string stability gets worse for higher delays in preceding vehicle only design. However, our distance-based filter design increase the robustness of the string stability condition for higher delays in the system as illustrated in Fig. 6.12. Therefore, the conventional preceding vehicle only design gets less robust and string unstable for specific frequency range with higher wireless communication delays, nevertheless the extension considering all preceding vehicles provides more robustness in string stability of the platoon driving for same type of higher wireless delays [85].

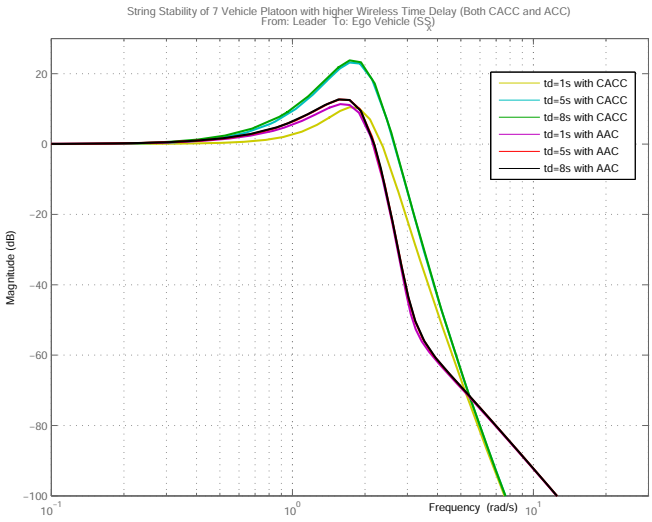


Figure 6.12: String stability of preceding vehicle only (CACC) and all preceding vehicles (AAC) algorithms with higher wireless time delays (between 1sec and 8sec)

6.5 String Stability Analysis

In this section, we analyze the string stability models for non cooperative and cooperative driving in Section 6.3 and state the stability criteria for highway platooning. At first, we define the transfer functions for non cooperative and cooperative modes. Since spacing error and position string stability functions are in the same characteristic, we define the stability criteria in platooning according to vehicle positions. In the following functions, we assume that $K_{pi} = K_{di}^2 = \omega_{K,i}^2$ as described in Section 5.3.2.

For non cooperative (adaptive cruise controlled) mode:

$$\begin{aligned} SS_{X_i} &= \frac{K_{di}s + K_{pi}}{\tau_i s^3 + (1 + h_{di}K_{di})s^2 + (K_{di} + h_{di}K_{pi})s + K_{pi}} \\ &= \frac{\omega_{K,i}s + \omega_{K,i}^2}{\tau_i s^3 + s^2 + \omega_{K,i}s + \omega_{K,i}^2} \quad if\ h_{di} = 0 \end{aligned} \tag{6.38}$$

For the stability analysis of (6.38), we use Routh stability test. According to the Routh stability criterion in Table 6.1, we find τ_i and $\omega_{K,i}$ as greater than 0, and $(\omega_{K,i} > \tau_i\omega_{K,i}^2)$ for the string stability of the highway platoon. Therefore, we obtain $\tau_i \gg \frac{1}{\omega_{K,i}}$ since $\omega_{K,i}$ selected greater than 1 using routh test and frequency based methods.

For preceding vehicle only (cooperative adaptive cruise controlled) mode:

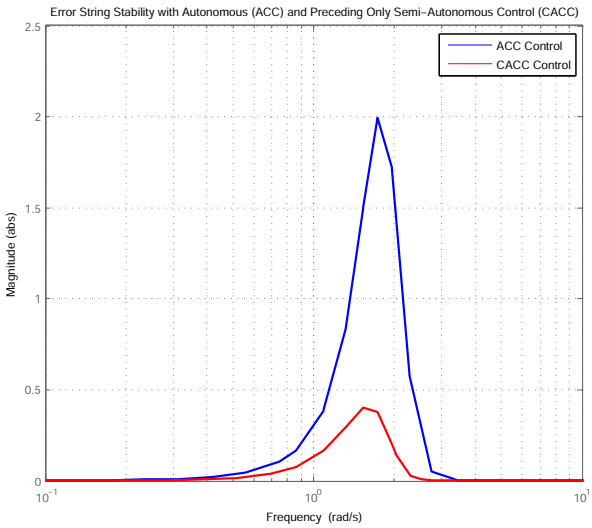


Figure 6.13: Error string stability comparison of non cooperative (ACC) and preceding vehicle only (CACC) modes assuming $t_d = 0.2$ and $\omega_{K,i} = 3$ (Ego Vehicle to Leader)

Table 6.1: Routh table of non cooperative control with constant spacing

s^3	τ_i	$\omega_{K,i}$
s^2	1	$\omega_{K,i}^2$
s^1	$(\omega_{K,i} - \tau_i\omega_{K,i}^2)$	0
s^0	$\omega_{K,i}^2$	0

$$
\begin{aligned}
SS_{X_i} &= \frac{(K_{di}s + K_{pi}) + s^2\sigma_i(\frac{-\theta_i s+2}{\theta_i s+2})(\frac{\tau_i s+1}{h_{di}s+1})}{\tau_i s^3 + (1 + h_{di}K_{di})s^2 + (K_{di} + h_{di}K_{pi})s + K_{pi}} \\
&= \frac{(\omega_{K,i}s + \omega_{K,i}^2)(\theta_i s + 2) - s^2(\theta_i s - 2)(\tau_i s + 1)}{(\theta_i s + 2)(\tau_i s^3 + s^2 + \omega_{K,i}s + \omega_{K,i}^2)} \quad if\ h_{di} = 0\ and\ \sigma_i = 1 \\
SS_{X_i} &= 1 \qquad if\ \theta_i = 0,\ h_{di} = 0,\ and\ \sigma_i = 1
\end{aligned}
\tag{6.39}
$$

First of all, if we ignore the communication delay, our preceding vehicle only controlled platoon is always string stable for constant spacing policy. For the general sting stability analysis of (6.39), we use Routh stability test again. For the sake of routh stability criteria, we first need to

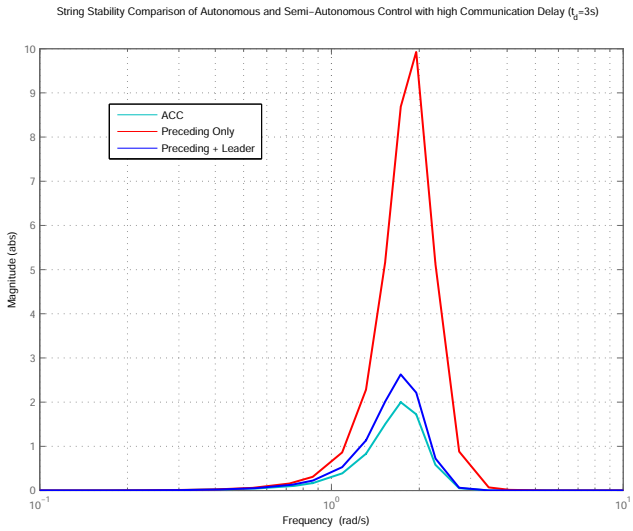


Figure 6.14: Error string stability of non cooperative (ACC), preceding vehicle only, and direct preceding+Leader cooperative modes with high communication delay assuming $t_d = 3$ and $\omega_{K,i} = 3$ (ego vehicle to the leading vehicle)

identify the first column parameters a_1 and b_1 of Table 6.2 as in (6.40) and (6.41).

$$a_1 = \frac{(2 + \theta_i\omega_{K,i})(2\tau_i + \theta_i + \omega_{K,i}\tau_i\theta_i)}{2\tau_i + \theta_i} \tag{6.40}$$

$$b_1 = \frac{a_1\omega_{K,i}(2 + \theta_i\omega_{K,i}) - 2\omega_{K,i}^2(2\tau_i + \theta_i)}{a_1} \tag{6.41}$$

Then, we obtain τ_i and $\omega_{K,i}$ as 0.8 and $3rad/s$, respectively. According to the (6.40) in Table 6.2, we have to obtain communication delay θ_i of the preceding vehicle only control lower than

Table 6.2: Routh table of preceding vehicle only mode with constant spacing

s^4	$\tau_i\theta_i$	$2 + \theta_i\omega_{K,i}$	$2\omega_{K,i}^2$
s^3	$2\tau_i + \theta_i$	$\omega_{K,i}(2 + \theta_i\omega_{K,i})$	0
s^2	a_1	$2\omega_{K,i}^2$	0
s^1	b_1	0	0
s^0	$2\omega_{K,i}^2$	0	0

1.33 to make string-stable platooning. So, we can define the relations of the terms with the help of string stable models, routh stability criterion, and frequency based analyses. Although the preceding vehicle only cruise control is more string stable than non cooperative control for lower communication delays as illustrated in Fig. 6.13, preceding vehicle only control corrupts for higher time delays as it can be seen in Fig. 6.14. For this reason, we get wireless state data from the leader vehicle in order to increase string stability as depicted in Fig. 6.14.

We illustrate the higher robustness of velocity dependent spacing policy dynamics over constant

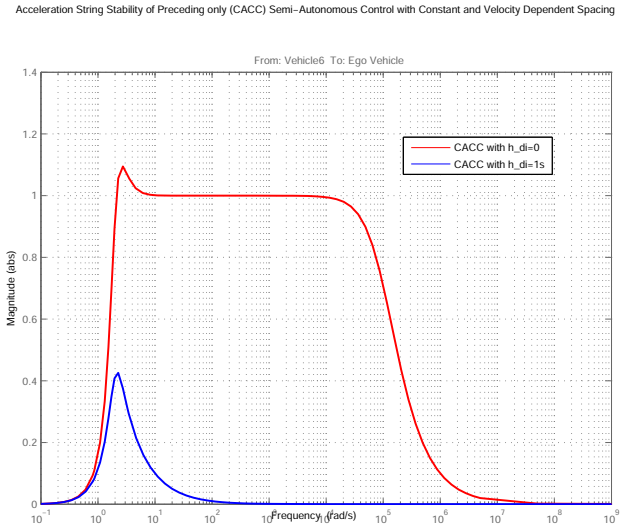


Figure 6.15: Acceleration string stability with constant and velocity dependent spacing policies of preceding vehicle only mode assuming $t_d = 0.2$ (from ego vehicle to direct preceding vehicle)

spacing policy for preceding vehicle only control in Fig. 6.15.

Finally, formal error string stability of all cooperative driving modes can be compared for higher time delays for two different control coefficients $\omega_{K,i}$ with the help of Fig. 6.16 and Fig. 6.17. As a result, the direct preceding and leader vehicles have higher importance for increasing the robustness of cooperative control to wireless communication delays.

6.6 Conclusion

In this chapter, we have first given the formal problem definition of one dimensional formation considering road vehicle platoons as described generally in Chapter 5. Then, we have designed

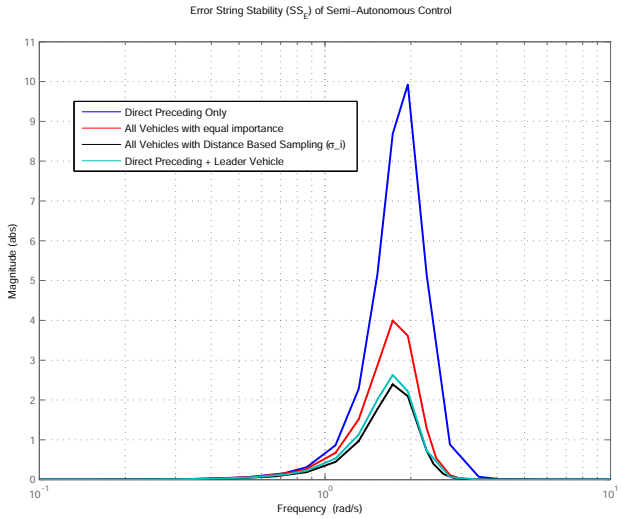


Figure 6.16: Error string stability of different cooperative driving modes with time delay $t_d = 3s$ and control breakpoint at $\omega_{K,i} = 3$

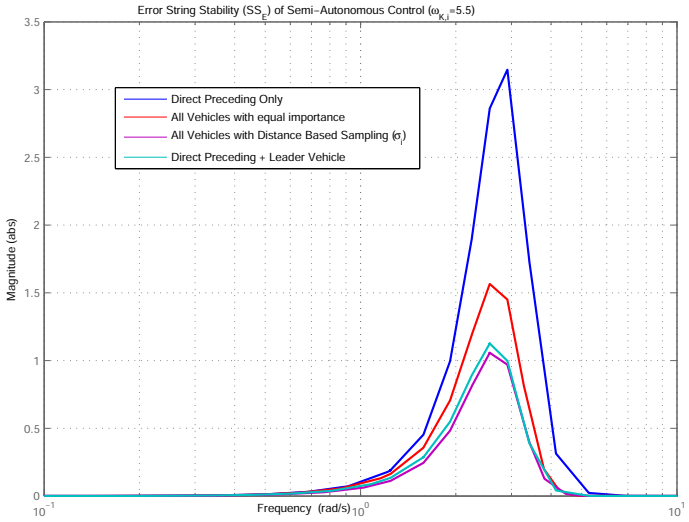


Figure 6.17: Error string stability of different cooperative driving modes with time delay $t_d = 3$ and control breakpoint at $\omega_{K,i} = 5.5$

Lyapunov based controller for vehicles considering second order translational dynamics and ignoring actuator dynamics at highways. This model proves the importance of the intervehicle

velocity error measurement over intervehicle distance error in control design for guaranteed stability. We have considered two different spacing strategies, namely constant and velocity-dependent. Next, we find error and string stability functions for non cooperative and cooperative driving modes in order to analyze control requirements for string stable platoons considering evaluation criteria. Additionally, the frequency based design of novel intervehicle distance sampled priority coefficient for better feed-forward filtering is given with its characteristics. Finally, we define the simplified models for different driving modes for finding the stability interval of control coefficients. Thus, we are able to make fair comparison on the string stability of various scenarios as illustrated in Section 6.5 with the help of identified control coefficients.

Chapter 7

Summary and Future Directions

In this thesis, we studied one-dimensional and multi-dimensional formations for the coordination of multi-aerial platforms and multi-road vehicles considering their dynamical models. At first, we gave the design and implementation of autonomous control on single ground and aerial platforms in Chapter 2. To this end, we provided all design stages of longitudinal modeling for Smart Fortwo car, small fixed-wing UAV model using experimental tests and commercial softwares, such as CarSim, Matlab/Simulink, and Tornado. Then, the modeling of quadrotor vehicle was described and related avionics are mentioned for autonomous flight. These various platforms were modeled for testing distributed cohesive motion algorithms and platooning strategies.
We designed linear based cohesive motion controller for distributed flight vehicle formations in Chapter 3. For this purpose, we gave the metrics and constraints of three dimensional formation and provided novel algorithm for real time trajectory generation of each agent in formation flight. We tested and analyzed the cohesive motion controllers using different platforms, such as generic fixed-wing UAVs, quadrotors, and piccolo-controlled fixed-wing UAVs.
Chapter 4 summarized the kinematic characteristics of UAV formations in three-dimensions. Then, we presented the immersion and invariance based control design to keep the tracking of helical trajectories under the assumption of measurement errors with specific constraints. In this chapter, we also provided the advantages of immersion and invariance based adaptive controller over non-adaptive controllers. The application of adaptive maneuver control to formation control was given with mathematical expressions.
In Chapters 5 and 6, we designed string stability guaranteed cooperative adaptive cruise controllers for urban and highway missions considering vehicle dynamics, actuator models, and wireless transmission. We first ignored actuator delays and wireless transmission in order to see the effects of model parameters and error measurements in string stable control design. In addition, we developed an intervehicle distance based prioritization technique in defining the

coefficients of feed-forward filter for string stable platoons under higher wireless transmission delays. We also described our metrics considering two different spacing strategies and provided the formal string stability analysis of non cooperative and cooperative driving.
We are planning to enhance our designs considering sensory measurement problems from different sources, and optimizing the controllers considering the errors due to the integration of different avionics and electronics on real testbeds. As a separate future direction, the string stable designs for one dimensional motions can be extended to three dimensional motion using UAV platforms with realistic constraints.
Next we elaborate two different future research areas for robust vehicle platooning on urban roads and highways: $i)$ use of dynamic road-side units on highways and $ii)$ detailed study of vehicle ordering in the platoon.

i) Employing Dynamic Road-Side Unit

First, we consider three vehicle platoons P_i, $i = 1, 2, 3$ on a highway, where the actual vehicle state information is communicated within each platoon, but there is no wireless transmission between platoons. Hence, although there is no awareness of the existence of the front platoon up until the back of the front platoon enters the range of the lidar sensor, we are capable of running CACC algorithm between the local vehicles of each platoon P_i. Noting that we are unable to implement CACC algorithm between the separate platoons without external support, we expect to see the benefits of using dynamic road-side units and transmission of the acceleration and other states of the preceding platoon to the follower platoons as illustrated in Fig. 7.1.

ii) Platoon Topology and Vehicle Ordering

Consider the vehicle ordering strategies for better platoon topology illustrated in Fig. 7.2. We have four vehicles in a platoon and there are mainly four different platoon topologies shaped with respect to the wireless information flow [66]. Topology 4 in Fig. 7.2 includes all other topologies by means of prioritizing the filter weight coefficients of all wireless preceding accelerations, accordingly.
The aim of using different wireless topologies is to increase robustness of platoon stability, when higher transmission delays, actuator delays, or other sensory disturbances are introduced. String stability condition and safety can be easily provided when we order vehicles according to their maximum acceleration and deceleration capacity. The order of vehicle models in heterogeneous traffic seems important for string stability of the platoon under the constant spacing policy. Thus, vehicles with a lower acceleration/deceleration capacity such as trucks or vans should be ordered in the front part of the platoon, while higher acceleration/deceleration capacity vehicles should

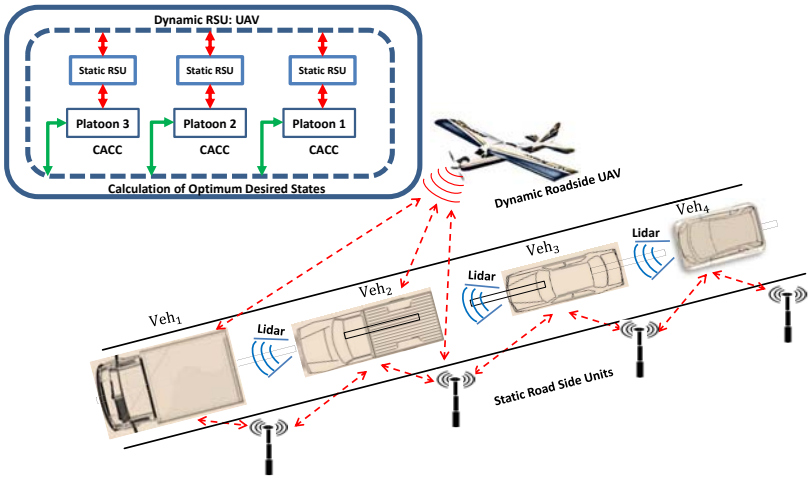


Figure 7.1: The highway traffic surveillance and control scenario with static and dynamic road side agents and Interaction chart between dynamic roadside unit, static roadside unit, and vehicle platoons

be located at the back of the same platoon. Otherwise, higher constant spacing for the trucks or vans for the string stability of the platoon would be required.

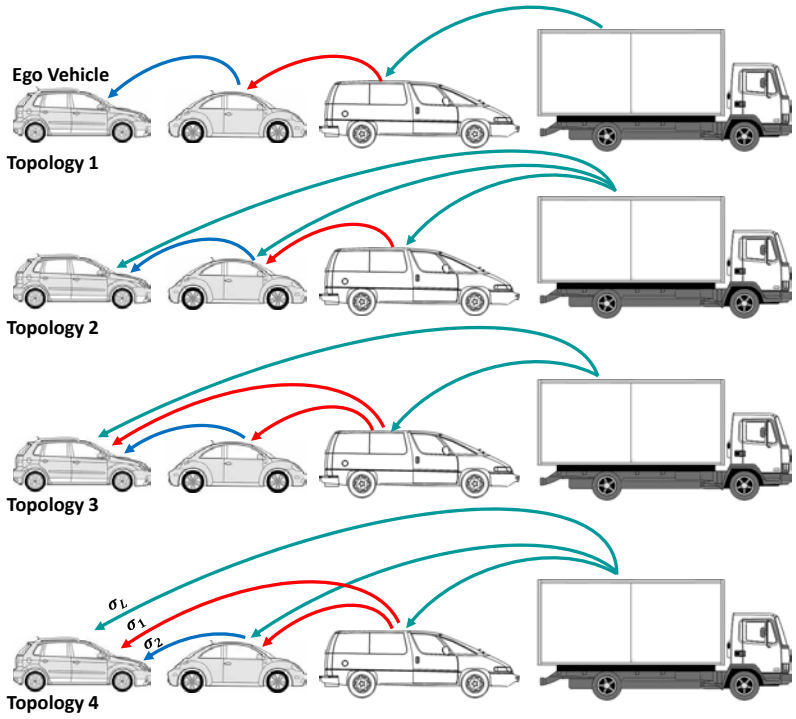


Figure 7.2: Different platooning topologies according to wireless transmission of front vehicle states: 1) Conventional CACC, 2) Preceding vehicle + Leader, 3) Ego vehicle receives from all preceding vehicles, the others receives only the direct preceding's data, 4) All preceding vehicles

APPENDICES

Appendix A

Path Planning Strategy

In our mission flight, the path planning strategy has two different segments, which are straight path following and coordinated turn. The unit vector in the direction of $\overline{w_i w_{i+1}}$ is given as:

$$q_i = \frac{w_{i+1} - w_i}{\|w_{i+1} - w_i\|} \tag{A.1}$$

The angle between $\overline{w_{i-1} w_i}$ and $\overline{w_i w_{i+1}}$ is $(\pi - 2\alpha)$ as in Fig. A.1. Then, we calculate the

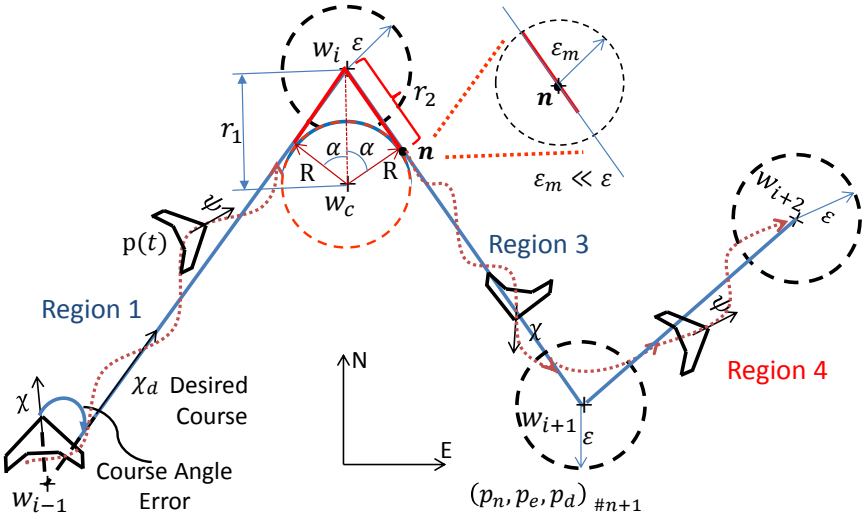


Figure A.1: Path planning strategy during traffic flow surveillance at highways

distances between two spheres r_1 and r_2 as

$$[r_1 \;\; r_2]^T = \mathrm{R}\,[\sec\alpha \;\;\; \tan\alpha]^{\mathrm{T}} \tag{A.2}$$

As the result of these calculations, the central location w_c of the flight turning reference circle is given as:

$$w_c = w_i - \mathrm{R}\sec\alpha\left(\frac{\mathrm{q_{i-1}} - \mathrm{q_i}}{\|\mathrm{q_{i-1}} - \mathrm{q_i}\|}\right) \tag{A.3}$$

$$n = w_i + (\mathrm{R}\tan\alpha)\,q_i \tag{A.4}$$

We first start with the straight trajectory $\overline{w_{i-1}w_i}$ following as soon as we enter the circular region of w_i with a radius ε. Then, we calculate the circular path with using next waypoint location and (A.2), (A.3), and (A.4). Since we enter the region of *n* with a radius ε_c, our UAV start to switch following the straight path $\overline{nw_i}$ in Region 3 of Fig. 2.2. In region 2, we implement coordinated turn maneuver with the help of aileron deflection. The relation between the rate of course angle and roll angle during the coordinated turn can be given as $\dot{\chi} = (g/V)\tan\phi$, [14]. Course angle χ is equal to yaw angle ψ, when we ignore wind or sideslip.

Appendix B

Intersection of Three Spheres in Three Dimensions

Consider three spheres $S(s_1, r_1), S(s_2, r_2), S(s_3, r_3)$ with centers located at $s_1 = (0, 0, 0)^T$, $s_2 = (x_2, 0, 0)^T$ and $s_3 = (x_3, y_3, 0)^T$, the intersection $B(s_1, r_1) \cap B(s_2, r_2) \cap B(s_3, r_3)$ of whose interiors is non-empty. The surface equations of these three spheres are given, respectively, by

$$x^2 + y^2 + z^2 = r_1^2 \quad \text{(B.1)}$$

$$(x - x_2)^2 + y^2 + z^2 = r_2^2 \quad \text{(B.2)}$$

$$(x - x_3)^2 + (y - y_3)^2 + z^2 = r_3^2 \quad \text{(B.3)}$$

The intersection of $S(s_1, r_1)$ and $S(s_2, r_2)$ is a circle perpendicular to the x axis. The x coordinates of the points on this circle can be calculated by subtracting (B.1) from (B.2):

$$(x - x_2)^2 - x^2 = r_2^2 - r_1^2. \quad \text{(B.4)}$$

Solving (B.4) for x gives

$$x = x_c = (x_2^2 + r_1^2 - r_2^2)/2x_2. \quad \text{(B.5)}$$

Substituting (B.5) into (B.1), we obtain the equation for the circle $S(s_1, r_1) \cap S(s_2, r_2)$:

$$y^2 + z^2 = r_1^2 - x_c^2. \quad \text{(B.6)}$$

The second step is to find the two intersection points of this circle and $S(s_3, r_3)$, whose existence is guaranteed by the fact that $B(s_1, r_1) \cap B(s_2, r_2) \cap B(s_3, r_3) \neq \emptyset$. The common x coordinate of the intersection point is given by (B.5), which actually is the equation for the plane containing

$S(s_1, r_1) \cap S(s_2, r_2)$. Hence, using (B.3), (B.5) and (B.6), we find the y and z coordinates of the two possible intersection points (x_c, y_c, z_c) in $S(s_1, r_1) \cap S(s_2, r_2) \cap S(s_3, r_3)$ as follows:

$$y_c = \frac{r_1^2 - r_3^2 + x_3^2 + y_3^2}{2y_3} - \frac{x_3}{y_3} x_c. \tag{B.7}$$

From (B.6), we can find the z coordinate:

$$z_c = \pm\sqrt{r_1^2 - x_c^2 - y_c^2}. \tag{B.8}$$

Next we find the intersection of $S(p_1, r_1)$, $S(p_2, r_2)$, $S(p_3, r_3)$ for the general case where the centers are located at $p_1 = (p_{1x}, p_{1y}, p_{1z})^T$, $p_2 = (p_{2x}, p_{2y}, p_{2z})^T$ and $p_3 = (p_{3x}, p_{3y}, p_{3z})^T$. The corresponding surface equations are

$$\begin{aligned}
(x - p_{1x})^2 + (y - p_{1y})^2 + (z - p_{1z})^2 &= r_1^2, && \text{(B.9)}\\
(x - p_{2x})^2 + (y - p_{2y})^2 + (z - p_{2z})^2 &= r_2^2, && \text{(B.10)}\\
(x - p_{3x})^2 + (y - p_{3y})^2 + (z - p_{3z})^2 &= r_3^2. && \text{(B.11)}
\end{aligned}$$

The idea is to find the rotation and translation mappings that can map p_1, p_2 and p_3 to s_1, s_2 and s_3, assuming $\|p_i - p_j\| = \|s_i - s_j\|, \forall i, j \in \{1, 2, 3\}$, use the formulas (B.5), (B.7), (B.8) to find the mapped intersection points, and then inverse map these points to find the intersection points in the actual coordinate frame. Therefore, we need to find a homogeneous transformation matrix T such that

$$\begin{bmatrix} s_i \\ 1 \end{bmatrix} = T \begin{bmatrix} p_i \\ 1 \end{bmatrix}, \ \forall\, i \in \{1, 2, 3\}, \tag{B.12}$$

where s_1, s_2, s_3 are as given in the beginning of the appendix; $x_2 = \|p_2 - p_1\|$, $\|s_3 - s_1\| = \|s_3\| = \|p_3 - p_1\|$. It can be easily verified that

$$T = \begin{bmatrix} R & -p_1 \\ 0 & 1 \end{bmatrix}, \ R = \begin{bmatrix} R_x^T & R_y^T & R_z^T, \end{bmatrix}^T \tag{B.13}$$

$$R_x = \frac{p_2 - p_1}{x_2}, \ R_y = \frac{x_2 p_3 - x_3 p_2}{x_2 y_3}, \ R_z = R_x \times R_{y_j}, \tag{B.14}$$

and hence

$$T^{-1} = \begin{bmatrix} R^T & R^T p_1 \\ 0 & 1 \end{bmatrix} \tag{B.15}$$

Therefore, the intersection of $S(p_1, r_1)$, $S(p_2, r_2)$, and $S(p_3, r_3)$ can be found applying the following steps:

1. Use (B.12) $-$ (B.14) to find s_1, s_2, and s_3.

2. Use (B.5) $-$ (B.8) to find x_c, y_c, and z_c (two sets of solutions)

3. Apply
$$\begin{bmatrix} p_c \\ 1 \end{bmatrix} = T^{-1} \begin{bmatrix} x_c, & y_c, & z_c, & 1 \end{bmatrix}^T \tag{B.16}$$
with T^{-1} given in (B.15) to find the intersection point p_c (two solutions).

References

[1] M. Alfraheed, A. Dröge, M. Klingender, D. Schilberg, and S. Jeschke. Longitudinal and lateral control in automated highway systems: Their Past, Present and Future. In *Intelligent Robotics and Applications*, pages 589–598. Springer, 2011.

[2] K. Allred, M. Freitas, J. Njord, J. Peters, and B. Warner. Safety applications of intelligent transportation systems in Europe and Japan. January 2006.

[3] I. Bayezit. Microavionics design and microavionics control implementation for cross-compatible platforms. Master's thesis, Electrical Engineering, Istanbul Technical University, Turkey, Dec. 2008.

[4] I. Bayezit, M. Amini, B. Fidan, and I. Shames. Cohesive motion control of autonomous formations in three dimensions. In *Proc. 6th Int. Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pages 205–210, December 2010.

[5] I. Bayezit, S. Ates, M. Aksugur, and G. Inalhan. The autonomy of fixed-wing aerial vehicles and experimental design steps on implmenting autonomous navigation, landing and take off operations for the Trainer 60 model aircraft. In *Proc. 5th Ankara International Aerospace Conference*, Aug. 2009.

[6] I. Bayezit and B. Fidan. Nonlinear maneuvering control of rigid formations of fixed wing UAVs. In *Proc. Second International Autonomous and Intelligent Systems Conference*, pages 124–133, June 2011.

[7] I. Bayezit and B. Fidan. Distributed cohesive motion control of flight vehicle formations. *IEEE Trans. on Industrial Electronics*, 60(12):5763–5772, 2013.

[8] I. Bayezit, B. Fidan, M. Amini, and I. Shames. Distributed cohesive motion control of quadrotor vehicle formations. In *Proc. ASME International Mechanical Engineering Congress and Exposition*, pages 797–805, November 2010.

[9] I. Bayezit, B. Fidan, and Jan P. Huissoon. Adaptive control of vehicle platoons with wireless state information. In *Proc. IFAC Workshop on Advances in Control and Automation Theory for Transportation Applications*, Sep. 2013.

[10] I. Bayezit, G. Inalhan, B. Fidan, and Jan P. Huissoon. Experimental modeling and adaptive control of an unmanned aerial vehicle as roadside assistance. In *Proc. 16th Int. IEEE Conf. on Intelligent Transportation Systems*, pages 1784–1790, 2013.

[11] I. Bayezit, T. Veldhuizen, B. Fidan, Jan P. Huissoon, and H. Lupker. Design of string stable adaptive cruise controllers for highway and urban missions. In *Proc. IEEE 50th Annual Allerton Conf. on Communication, Control, and Computing*, pages 106–113, 2012.

[12] S. Bayraktar, G.E. Fainekos, and G.J. Pappas. Experimental cooperative control of fixed-wing unmanned aerial vehicles. In *Proc. 43rd IEEE Conference on Decision and Control*, volume 4, pages 4292–4298, Dec. 2004.

[13] R.W. Beard and T.W. McLain. *Navigation, Guidance, and Control of Small and Miniature Air Vehicles*. Brigham Young University, 2010.

[14] R.W. Beard and T.W. McLain. *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press, 2012.

[15] B. Bethke, M. Valenti, and J.P. How. UAV task assignment. *IEEE Robotics Automation Magazine*, 15(1):39–44, March 2008.

[16] S. Bouabdallah, P. Murrieri, and R. Siegwart. Design and control of an indoor micro quadrotor. In *Proc. IEEE Int. Conference on Robotics and Automation, ICRA*, volume 5, pages 4393–4398, April 2004.

[17] A.E. Bryson. *Control of Spacecraft and Aircraft*. Princeton Univ, 1994.

[18] G. Cai, B.M. Chen, K. Peng, M. Dong, and T.H. Lee. Modeling and control of the yaw channel of a UAV helicopter. *IEEE Trans. on Industrial Electronics*, 55(9):3426–3434, Sept. 2008.

[19] S.A. Cambone. *Unmanned Aircraft Systems Roadmap 2005-2030*. Defense Technical Information Center, 2005.

[20] L. Chaimowicz, N. Michael, and V. Kumar. Controlling swarms of robots using interpolated implicit functions. In *Proc. IEEE International Conference on Robotics and Automation, ICRA*, pages 2487–2492, April 2005.

[21] H. Chao, Y. Cao, and Y. Chen. Autopilots for small unmanned aerial vehicles. *International Journal of Control, Automation and Systems*, 8(1):36–44, 2010.

[22] H. Chen, X. Wang, and Y. Li. A survey of autonomous control for UAV. In *IEEE International Conference on Artificial Intelligence and Computational Intelligence*, volume 2, pages 267–271, 2009.

[23] M. Chyba. Increasing underwater vehicle autonomy by reducing energy consumption. *Ocean Engineering*, 36(1):62–73, Jan. 2009.

[24] B.T. Clough. Metrics, schmetrics! How the heck do you determine a UAV's autonomy anyway. Technical report, DTIC Document, 2002.

[25] M. Cook. *Flight Dynamics Principles*. Butterworth-Heinemann, 2007.

[26] A. Das, M. van Eenennaam, T. Veldhuizen, I. Bayezit, F. Liu, and G. Tillema. Team futurum in the grand cooperative driving challange. Technical Report 1, University of Waterloo, Fontys Automotive and University of Twente, High-Tech Automotive Campus, Helmond, May 2011.

[27] M.D.L.C. de Oliveira. Modeling, identification and control of a quadrotor aircraft. Master's thesis, Control Engineering, Czech Technical University, Prague, June 2011.

[28] S. Drake, K. Brown, J. Fazackerley, and A. Finn. Autonomous control of multiple UAVs for the passive location of radars. In *Proc. 2nd Int. Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pages 403–409, Dec. 2005.

[29] M.O. Efe. Robust low altitude behavior control of a quadrotor rotorcraft through sliding modes. In *Proc. 15th IEEE Mediterranean Conference on Control and Automation*, pages 1–6, June 2007.

[30] B. Etkin and L.D. Reid. *Dynamics of Flight: Stability and Control*. Wiley, 1995.

[31] J. Ferruz, M. Vega, A. Ollero, and V. Blanco. Reconfigurable control architecture for distributed systems in the hero autonomous helicopter. *IEEE Trans. on Industrial Electronics*, 58(12):5311–5318, Dec. 2011.

[32] B. Fidan and B.D.O. Anderson. Switching control for robust autonomous robot and vehicle platoon formation maintenance. In *Proc IEEE Mediterranean Conference on Control & Automation*, pages 1–6, 2007.

[33] B. Fidan, B.D.O. Anderson, C. Yu, and J.M. Hendrickx. Persistent autonomous formations and cohesive motion control. In Petros A. Ioannou and Andreas Pitsillides, editors, *Modeling and Control of Complex Systems*, chapter 8, pages 247–275. CRC, Dec. 2007.

[34] B. Fidan and I. Bayezit. Immersion and invariance based adaptive control of three dimensional autonomous vehicles maneuvers. In *Proc. 10th IFAC Adaptation and Learning in Control and Signal Processing*, 2010.

[35] T.I. Fossen. *Guidance and Control of Ocean Vehicles*. Wiley, 1994.

[36] K.S. Fu, R.C. Gonzalez, and C.S. Lee. *Robotics: Control, Sensing, Vision, and Intelligence*. McGraw Hill, 1987.

[37] V. Gazi and B. Fidan. Coordination and control of multi-agent dynamic systems: Models and approaches. *Springer Swarm Robotics*, 4433:71–102, May 2007.

[38] V. Gazi and K.M. Passino. Stability of a one-dimensional discrete-time asynchronous swarm. *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, 35(4):834–841, Aug. 2005.

[39] M.S. Grewal, L.R. Weill, and A.P. Andrews. *Global Positioning Systems, Inertial Navigation, and Integration*. Wiley-Interscience, 2007.

[40] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas. Cooperative air and ground surveillance. *IEEE Robotics Automation Magazine*, 13(3):16–25, Sept. 2006.

[41] G.M. Hoffmann, S.L. Waslander, and C.J. Tomlin. Quadrotor helicopter trajectory tracking control. In *Proc. AIAA Guidance, Navigation, and Control Conf.*, pages 1–14, April 2008.

[42] J.P. How, B. Bethke, A. Frank, D. Dale, and J. Vian. Real-time indoor autonomous vehicle test environment. *IEEE Control Systems Magazine*, 28(2):51–64, April 2008.

[43] R.N. Jazar. *Advanced Dynamics*. Wiley, 2011.

[44] E.G. Jones, M.B. Dias, and A. Stentz. Time-extended multi-robot coordination for domains with intra-path constraints. *Springer Autonomous Robots*, 30(1):41–56, Jan. 2011.

[45] A. Jundi, S. Boumaiza, F. Karray, and F. Sattar. Adaptive communication gateway for vehicular networking applications. In *Proc. ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications*, pages 9–16, 2012.

[46] R. Kianfar, P. Falcone, and J. Fredriksson. Reachability analysis of cooperative adaptive cruise controller. In *Proc 15th IEEE International Conference on Intelligent Transportation Systems*, pages 1537–1542, 2012.

[47] M. Kwakkernaat and S. de Hair-Buijssen. GCDC 2011 rules & technology document. Technical Report 2.0, TNO, 5700 AT Helmond The Netherlands, April 2011.

[48] H. Li, F. Karray, O. Basir, and I. Song. A framework for coordinated control of multiagent systems and its applications. *IEEE Trans. on Systems, Man and Cybernetics, Part A: Systems and Humans*, 38(3):534–548, 2008.

[49] C. Lin, C. Huang, C. Lee, and M. Chao. Identification of flight vehicle models using fuzzified eigensystem realization algorithm. *IEEE Trans. on Industrial Electronics*, 58(11):5206–5219, Nov. 2011.

[50] T. Madani and A. Benallegue. Backstepping sliding mode control applied to a miniature quadrotor flying robot. In *Proc. 32nd IEEE Annual Conference on Industrial Electronics*, pages 700–705, Nov. 2006.

[51] D. Mclean. *Automatic Flight Control Systems*. Prentice Hall, 1990.

[52] N. Michael, J. Fink, and V. Kumar. Experimental testbed for large multirobot teams. *IEEE Robotics Automation Magazine*, 15(1):53–61, March 2008.

[53] N. Michael, J. Fink, and V. Kumar. Cooperative manipulation and transportation with aerial robots. *Autonomous Robots*, 30(1):73–86, September 2010.

[54] Y. Morel and A. Leonessa. Direct Adaptive Tracking Control of Quadrotor Aerial Vehicles. In *Proc. ASME International Mechanical Engineering Congress and Exposition*, 2006.

[55] R.M. Murray. Recent research in cooperative control of multivehicle systems. *ASME Journal of Dynamic Systems, Measurement, and Control*, 129:571, 2007.

[56] G. Naus, R. Vugts, J. Ploeg, R. van de Molengraft, and M. Steinbuch. Cooperative adaptive cruise control, design and experiments. In *American Control Conference*, 2010.

[57] G.J.L. Naus, R. Vugts, J. Ploeg, M.J.G. Van de Molengraft, and M. Steinbuch. Towards on-the-road implementation of cooperative adaptive cruise control. In *Proc. 16th World Congress & Exhibition on Intelligent Transport Systems and Services*, Stockholm, Sweden, September 2009. ITS World.

[58] G.J.L. Naus, R.P.A. Vugts, J. Ploeg, M.J.G. Van de Molengraft, and M. Steinbuch. String-stable CACC design and experimental validation: A frequency-domain approach. *IEEE Trans. on Vehicular Technology*, 59(9):4268–4279, 2010.

[59] R.C. Nelson. *Flight Stability and Automatic Control*. McGraw-Hill, 1997.

[60] C. Nowakowski, S.E. Shladover, D. Cody, and et al. Cooperative adaptive cruise control: Testing drivers' choices of following distances. Technical report, California PATH Program, Institute of Transportation Studies, University of California at Berkeley, November 2010.

[61] R. Olfati-Saber and R.M. Murray. Distributed cooperative control of multiple vehicle formations using structural potential functions. In *Proc. IFAC 15th Triennial World Congress*, pages 346–352, July 2002.

[62] R. Ortega, L. Hsu, and A. Astolfi. Immersion and invariance adaptive control of linear multivariable systems. *Elsevier Systems and Control Letters*, 49(1):37–47, May 2003.

[63] S.S. Patankar, D.E. Schinstock, and R.M. Caplinger. Application of pendulum method to UAV momental ellipsoid estimation. In *Proc. AIAA 6th Aviation Technology, Integration and Operations Conference*, 2006.

[64] A. Peterson, L. Nielson, G. Xerogiannakis, and J. Helbo. Robust flight controller. Technical report, Aalborg University, Institute of Electronic Systems, June 2003.

[65] J.T. Platts. Autonomy in unmanned air vehicles. *Aeronautical Journal*, 110(1104):97–105, 2006.

[66] J. Ploeg, A.F.A. Serrarens, and G.J. Heijenk. Connect & drive: design and evaluation of cooperative adaptive cruise control for congestion reduction. *Journal of Modern Transportation*, 19(3):207–213, 2011.

[67] Z. Qu, J. Wang, and R.A. Hull. Cooperative control of dynamical systems with application to autonomous vehicles. *Trans. on Automatic Control*, 53(4):894–911, 2008.

[68] R. Rajamani. *Vehicle Dynamics and Control*. Mechanical Engineering Series. Springer, 2nd edition, 2012.

[69] W. Ren and R. Beard. *Distributed Consensus in Multi-Vehicle Cooperative Control: Theory and Applications*. Springer, 2008.

[70] G.N. Roberts and R. Sutton. *Advances in Unmanned Marine Vehicles*. Inspec/IEE, 2006.

[71] J. Roskam. *Methods for estimating stability and control derivatives of conventional subsonic airplanes*. The University of Virginia, 1971.

[72] W.J. Rugh and J.S. Shamma. Research on gain scheduling. *Automatica*, 36(10):1401–1425, 2000.

[73] I. Sadeghzadeh, A. Mehta, A. Chamseddine, and Y. Zhang. Active fault tolerant control of a quadrotor UAV based on gainscheduled PID control. In *Proc. 25th IEEE Canadian Conference on Electrical & Computer Engineering (CCECE)*, pages 1–4, 2012.

[74] S. Sandeep, B. Fidan, and C. Yu. Decentralized cohesive motion control of multi-agent formations. In *Proc. 14th Mediterranean Conference on Control and Automation*, pages 1–6, June 2006.

[75] R. Schoop, R. Neubert, and A.W. Colombo. A multiagent-based distributed control platform for industrial flexible production systems. In *Proc. IEEE 27th Annual Conference on Industrial Electronics Society*, volume 1, pages 279–284, Nov. 2001.

[76] P. Seiler, A. Pant, and K. Hedrick. Disturbance propagation in vehicle strings. *IEEE Trans. on Automatic Control*, 49(10):1835–1842, 2004.

[77] D.H. Shim, Hoam Chung, and S.S. Sastry. Conflict-free navigation in unknown urban environments. *IEEE Robotics Automation Magazine*, 13(3):27–33, Sept. 2006.

[78] A Smith. *Wealth of Nations*, volume Vol. I of *Economics*. W. Strahan and T. Cadell, London, 1776.

[79] R.S. Smith and F.Y. Hadaegh. Distributed estimation, communication and control for deep space formations. 1(2):445–451, March 2007.

[80] R.S. Smith and F.Y. Hadaegh. Distributed estimation, communication and control for deep space formations. 1(2):445–451, March 2007.

[81] B.V. Sorensen, C.K. Nielsen, J.B. Vilhelmsen, R.B. Persson, S.H. Hansen, and S.K. Jensen. Autonomous model airplane. Technical report, Aalborg University, Institute of Electronic Systems, May 2005.

[82] B.L. Stevens and F.L. Lewis. *Aircraft Control and Simulation*. Wiley, 2003.

[83] J.R. Treat. Tri-level study of the causes of traffic accidents: An overview of final results. In *Proc 21st Annual Conference of the American Association for Automotive Medicine*, pages 391–403, 1977.

[84] B. van Arem, H. Driever, P. Feenstra, J. Ploeg, G. Klunder, I. Wilmink, A. Zoutendijk, Z. Papp, and B. Netten. Design and evaluation of an integrated full-range speed assistant. Technical report, TNO, 2007.

[85] T.H.A. van den Broek, J. Ploeg, and B.D. Netten. Advisory and autonomous cooperative driving systems. In *Proc. IEEE International Conference on Consumer Electronics*, pages 279–280, 2011.

[86] J. Wang, H. Mounier, A. Cela, and S.I. Niculescu. Event driven intelligent PID controllers with applications to motion control. In *Proc. 18th IFAC World Congress*, volume 18, pages 10080–10085, Aug. 2011.

[87] S.L. Waslander, G.M. Hoffmann, J.S. Jang, and C.J. Tomlin. Multi-agent quadrotor testbed control design: Integral sliding mode vs. reinforcement learning. In *Proc. IEEE International Conference on Intelligent Robots and Systems*, pages 3712–3717, July 2005.

[88] D. Yanakiev and I. Kanellakopoulos. Variable time headway for string stability of automated heavy-duty vehicles. In *Proc 34th IEEE Conference on Decision and Control*, volume 4, pages 4077–4081, 1995.

[89] C. Yu, J.M. Hendrickx, B. Fidan, B.D.O. Anderson, and V.D. Blondel. Three and higher dimensional autonomous formations: Rigidity, persistence and structural persistence. *Elsevier Automatica*, 43(3):387–402, March 2007.

[90] S. Zhai, B. Fidan, S.C. Oztürk, and V. Gazi. Single view depth estimation based formation control of robotic swarms: Obstacle avoidance, simulation, and practical issues. In *Proc. 16th Mediterranean Conference on Control and Automation*, pages 1162–1167, June 2008.

[91] Y. Zhang, B. Fidan, and P.A. Ioannou. Backstepping control of linear time-varying systems with known and unknown parameters. *IEEE Trans. on Automatic Control*, 48(11):1908–1925, Nov. 2003.