TÉCNICO LISBOA

Event-Triggered Control for Modular Aerial Vehicles

Miguel Ângelo Gomes Esteves da Costa

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisors: Prof. Pedro Daniel Graça Casau

Prof. Rita Maria Mendes de Almeida Correia da Cunha

Examination Committee

Chairperson: Prof. João Luís Da Costa Campos Gonçalves Sobrinho

Supervisor: Prof. Pedro Daniel Graça Casau

Member of the Committee: Prof. David Alexandre Cabecinhas

November 2023

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

I want to express my deepest gratitude to my family, whose unwavering support has shaped me into the person I am today. Thank you to my parents for providing invaluable guidance throughout my educational journey and opening up possibilities that they themselves never had. It is through their relentless efforts and sacrifices that I've been able to pursue the future I envisioned, and everything I have today was made possible by their dedication.

A heartfelt thank you goes out to my friends and loved ones, who have been an integral part of these transformative five years. They've been my anchor in the toughest moments and have offered the companionship that fueled my pursuit of a bachelor's, a master's, and most recently, this thesis. I can say with confidence that without the friendships I forged during my time at Tecnico, I wouldn't be where I am today.

I extend my thanks to the community and dedicated faculty at Instituto Superior Técnico. Their support has been instrumental throughout these five years, igniting my own passion for learning. I am indebted to my advisor, Pedro Casau, and co-advisor, Rita Cunha, for their unwavering guidance through the challenges posed by this thesis. Professor Rita Cunha, who taught me in multiple courses, instilled in me a deep interest and passion for the subject of control and robotics. Professor Pedro Casau, who has provided exceptional support from the integrating project to this final thesis, has been by my side for over a year now. His profound knowledge and, most importantly, his consistent availability and willingness to help have been invaluable. I consider myself incredibly fortunate to have chosen Professor Pedro Casau as my supervisor.

To each and every one of you – Thank you.

Abstract

This thesis addresses the limitations of small-sized Unmanned Aerial Vehicles (UAVs) by proposing a modular approach, where multiple UAV modules form a single vehicle. This method aims to overcome the constraints of individual UAVs. The thesis focuses on proving the theoretical feasibility and effectiveness of this modular vehicle concept through simulations. It employs a control strategy involving full state feedback control within an event-triggered methodology, treating the system as an open hybrid automaton — a mathematical model that defines a hybrid system. The study uses a coaxial helicopter as the main module type and incorporates Event-Triggered Control and Linear Quadratic Regulator controller logic.

The simulations demonstrate great results, for the scenarios of both connected and disconnected modules. The modular vehicle concept is envisioned to be scalable, potentially broadening its applications across various domains. This thesis not only enhances the versatility of UAVs but also depicts the Event-Triggered Control paradigm for designing and controlling autonomous modular vehicles, in an effort to broaden their capabilities for tasks previously constrained by their size and load-bearing capacities.

Keywords

Hybrid control systems, event-triggered control, linear quadratic regulator, open hybrid automata.

Resumo

Esta tese aborda as limitações dos Veículos Aéreos Não Tripulados (VANTs) de pequeno porte, propondo uma abordagem modular, onde vários módulos de VANTs formam um único veículo. Este método visa superar as restrições dos VANTs individuais. A tese foca em comprovar a viabilidade teórica e eficácia deste conceito de veículo modular por meio de simulações. Emprega uma estratégia de controle que envolve retroalimentação de estado completo numa metodologia acionada por eventos, tratando o sistema como um autómato híbrido aberto — um modelo matemático que define um sistema híbrido. O estudo utiliza um helicóptero coaxial como o principal tipo de módulo e incorpora uma lógica de Controlo Acionado por Eventos e um controlador do tipo Regulador Quadrático Linear.

As simulações demonstram excelentes resultados, para ambas situações, de dois módulos conectados e desconectados. O conceito de veículo modular é imaginado com o objetivo ser escalável, ampliando potencialmente as suas aplicações em diversos domínios. Esta tese não só aprimora a versatilidade dos VANTs, mas também mostra a implementação do paradigma de Controlo Acionado por Eventos para o controlo de veículos modulares autónomos, buscando alargar as suas capacidades para tarefas anteriormente limitadas pelo seu tamanho e capacidade de carga.

Palavras Chave

Controlo de sistemas híbridos, controlo acionado por eventos, regulador quadrático linear, autómato híbrido aberto.

Contents

List of Figures

Acronyms

UVS	Unmanned Vehicle System
UAV	Unmanned Aerial Vehicle
ETC	Event-Triggered Control
TTC	Time-Triggered Control
LQR	Linear Quadratic Regulator
AV	Aerial Vehicle
EU	European Union
MAS	Maritime Aerial Surveillance
ARE	Algebraic Riccati Equation
OHA	Open Hybrid Automata

Nomenclature

Mathematical Notation

$\mid x \mid^{\circ}_{\mathcal{A}}$	The distance between x and a set $\mathcal{A}$ or, $\inf_{y \in \mathcal{A}} \mid x - y \mid$
$B(a; \delta)$	The open ball in $\mathbb{R}^n$ with center $a \in \mathbb{R}^n$ and radius $\delta > 0$.
$f(B)$	The image of B under f
$f : \mathbb{R}^m \to \mathbb{R}^n$	A function that assigns, to each point $x \in \mathbb{R}^m$, a point in $\mathbb{R}^n$.
$f^{-1}(Y)$	The preimage of Y under f
$M : \mathbb{R}^m \rightrightarrows \mathbb{R}^n$	A set-valued mapping associating, with every point $x \in \mathbb{R}^m$, a subset of $\mathbb{R}^n$.
$M : X \rightrightarrows \mathbb{R}^n$	A set-valued mapping associating, with every point $x \in X$, a subset of $\mathbb{R}^n$.
$\operatorname{dom} f$	The domain of function f.
$\dot{x}$	Derivative in time of x.
$\exists$	Existential quantifier
$\forall$	Universal quantifier
$\mathbb{R}$	The set of real numbers
$\mathbb{R}^n$	The set of real n-dimentional vector
$\mathbb{R}^n \setminus X$	Complement of set $X \in \mathbb{R}$

$\mathbb{R}^{n \times m}$	The set of real $n \times m$ matrices
$\sup S$	The least upper bound of S.
A^T	Transpose of matrix A
A^{-1}	Inverse of matrix A
I	Identity matrix of appropriate dimention
$x \in X$	x is an element of set X
$X \subset Y$	X is a subset of Y
$X \cup Y$	Union of the sets X and Y

Physics constants

τ	Maximum time interval between events
A	State matrix
B	Input matrix
K	Feedback gain matrix
p_i	Position vector of the i-th module
u	Input Vector
$u^\star$	Desired calculated input
v_i	Velocity vector of the i-th module
x	State vector
$x^\star$	Desired calculated state
y	Output vector

1

Introduction

Contents

In today's world, nearly every industry faces the imperative of optimizing production costs and speed. The pressure to produce more, and at an accelerated pace, has driven industries to find and adopt strategies to meet these demands. In this sense, the industrial revolution and technological advances have changed the way almost every industry operates. The world of automation and the need for greater efficiency has significantly increased with the advent of robotics and, in particular, fully automated Unmanned Vehicle System (UVS) [8]. These systems can offer many advantages over conventional manned vehicles, including lower fuel costs, reduced hardware requirements, and safer options when operating in hostile and high-risk environments. In this sense, we refer to a UVS as any system that deploys vehicles that do not rely on human interaction. The era of fully operated systems that depend solely on human workers is long outdated. When looking at almost any production line, we find ourselves in a hybrid work environment where some aspects are done by humans and some fully automated tasks are performed by UVSs. This dynamic highlights the continuous technological evolution in this field, underscoring its critical significance for investment and inquiry. Integrating human expertise with cutting-edge automation not only boosts productivity but also drives ongoing progress in industrial processes.

In the realm of system engineering, devising an optimal control strategy stands as a central hurdle. Control, in this context, refers to the art of guiding and regulating the behavior of a system to achieve desired outcomes. Each algorithm possesses its unique strengths and weaknesses, further complicating the selection process. An Event-Triggered Control (ETC) approach, is a type of controller methodology where the computation of the actuator signals happens only at events. In general, these events can be triggered by various factors, such as reaching a maximum time limit or experiencing a significant deviation from the objective. This methodology is particularly desirable in systems that rely on the intercommunication between modules. This is because ETC limits sensor and control computations and/or communications to instances where it is actually required. By using this approach, we are able to achieve a very efficient and economical approach in terms of energy consumption of the modules, but most importantly, in terms of the computation-intensive controllers [9].

A Linear Quadratic Regulator (LQR) control strategy is a well known control algorithm and commonly used strategy when controlling Unmanned Aerial Vehicles (UAVs). The rationale behind this strategy relies on defining a quadratic cost function as part of an optimization problem that computes the state and input values that minimize this same function. Tweaking the quadratic cost function parameters affects the prioritization of what is being minimized, alters the feedback gain, and consequently the system response. This control methodology can be combined with a hybrid system to create a hybrid automaton that functions as a multimode controller with different feedback gains and/or state-space matrices [10].

The system under study is defined as a hybrid system, meaning it is not limited to a single time

domain, whether it be discrete or continuous. As we are implementing a multimode algorithm, there are instances during continuous motion and control where impulsive dynamics must be considered. For example, when changing modes of operation, elements such as feedback gain or even the state space dynamics may change. This means the system cannot be restricted to continuous flows, as we are also dealing with collisions that alter the dynamics of the system [11].

The concept of modular aerial vehicles relies on the idea of multiple independent flying modules, which can either act independently or coupled together to function as one single UAV. These vehicles can be rearranged to meet different requirements for different tasks. The main advantage of a modular system compared to a generic all-purpose UAV is its versatility. Each module can have a single specialization, which means that depending on the configuration of the modules, we can control aspects like thrust power and types of sensory capabilities. Though this is believed to be a step forward and a possible solution to many of the problems found in UAVs, there is still a long way to go until we achieve full autonomy [12].

In this thesis, the main focus is on the development of a hybrid controller consisting of independent flight modules that can be arranged in various ways. This allows them to meet the requirements of a particular flight mission. This touches not only on the issue of controlling an aircraft, but also on controlling a swarm of many modules. These can operate together or separately, each with a specific primary objective. Introducing new obstacles to consider, such as communication and interaction between UAVs, but on the other hand, the main benefits are that by integrating many modules together, we can significantly enhance the maximum load capacity (when compared to the single vehicle approach) and can also specialize different modules with distinct attributes (sensory and actuating capabilities). This opens new doors that were previously inaccessible, making drone technology a valuable tool in areas where it was previously limited. Some domains where this approach has much to offer, like agriculture and border/traffic control, will be elaborated upon in detail in the following sections. In addition to an introductory overview of the control system to be established, the main objective of this thesis is to ensure the feasibility of the work to be done and to propose the results to be obtained. In this context, some preliminary results and simulations are presented and shown to support the presented idea.

1.1 Motivation

One of the main advantages of studying UAVs is to compare these unmanned systems with their conventional counterparts. UAVs distinguish themselves by not requiring human pilots. This fundamental characteristic allows them to surmount many of the limitations encountered with manned vehicles. The absence of an onboard pilot enables the production of smaller units, reducing the required size.

Additionally, as they are controlled remotely or autonomously, they do not necessitate an onboard user interface or other amenities designed for housing a pilot. Consequently, UAVs entail lower hardware costs, making them more cost-effective to assemble. Moreover, their compact and lightweight design leads to substantial reductions in operational expenses, as the smaller and lighter frame significantly diminishes fuel consumption. Furthermore, the absence of an onboard pilot results in reduced expenses related to human resources and associated costs.

1.1.1 Modular Aerial Vehicles

As much as UAVs offer many solutions and remove many limitations when compared to conventional human-piloted vehicles, there are many other obstacles that must be considered. Firstly, due to their small frames, all electronic components such as sensors, actuators, computing units, and even fuel units must be kept as small and light as possible. This causes problems when trying to reduce costs due to the more expensive hardware required.

In many situations, employing UAVs may not be the most appropriate solution for the task at hand. For instance, when significant force is required, utilizing UAVs is often not a viable option. It is this constraint that inspired the concept of multiple UAVs functioning as a unified entity—a bundled approach to UAV control, where a collection of modules operates as a larger and more capable vehicle.

Consider, for instance, industrial settings where the efficient management of hefty objects necessitates the use of forklifts and hydraulic elevators. An effective solution for enhancing the relocation of such masses could involve the deployment of a modular vehicle. Unlike traditional equipment, UAV modules offers unparalleled freedom of movement, capable of adhering to any desired trajectory. While conventional aerial vehicles may fall short in terms of force and propulsion. Together, these modules function as a single, more powerful vehicle capable of exerting greater forces.

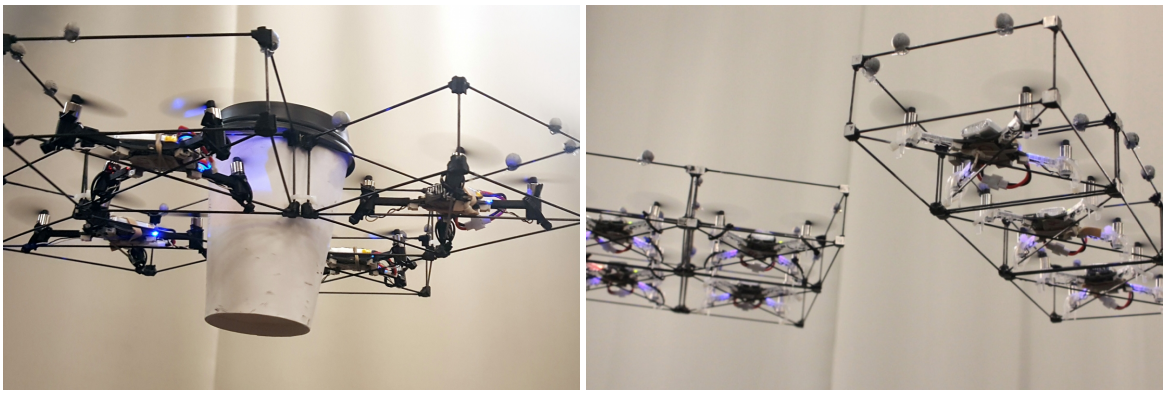

Figure 1.1: Examples of UAV modules formations for different goals. On the left, the “gripper” works in order to move the bucket and on the right multiple ways of creating platforms. Images from [1].

The previous statement focuses mainly on the benefits of working with multiple units of UAVs that complement each other, increasing the amount of force exerted by the vehicles. Another key motivation behind this modular approach is the idea of equipping different flying modules with different sensors and actuators, with each unit specialized in a certain subtask of the overall objective. For example, if the main objective is to find a certain object and act on it – for example, landing at a certain location – we may want specialized hardware such as RGB cameras, thermal cameras, LiDAR sensors, etc, depending on the scenario. Instead of relying on one overly complicated vehicle equipped with various components, it is possible for different vehicles with different capabilities to be used based on the specific task. This solution offers a wide range of capabilities that can be adjusted to various requirements, ultimately leading to a better distribution of the overall resources.

When using UVSs, the obstacles include not only the physical characteristics involved, but also the computing requirements. This is especially important when dealing with fully automated devices, i.e. when a UVS is used for a specific task without requiring human assistance. This requires not only powerful computational capacity, but also robust control algorithms capable of handling a wide variety of situations, such as following a desired trajectory, avoiding obstacles, dealing with external inputs and other undesired interactions, while achieving the desired goal quickly and cost-effectively. Many methodologies have been considered when studying the problem in question. There are a variety of control approaches that can be used when controlling a vehicle. For instance for simple systems, an open-loop control startegy may suffice, but that is not the case when working with more complex systems such as the one in question.

1.1.2 UAVs Applied to Traffic Control

Traffic control and transportation optimization in large cities is an example of a great use of UAVs that is currently being studied and can make a significant contribution. By utilizing, UAVs it is possible to perform a wide area surveillance of the vehicles and their movement, obtaining information and data that can be used by the drivers to better direct them to their final destination.

Currently, the data acquisition of this matter is of various natures and through many approaches. Ranging from the simple manual counting, survey stations and automatic traffic meters, but also more advanced recent methods like data obtained from mobile phone GPS data. [13]. Nevertheless, the point of the matter is the interest revolving this type of data acquisition originates from the importance behind this data. Traffic control, and subsequently the data behind it, has great influence on the inner working of a metropolis. The efficiency (and security) of the main transport system can ensure the growth or the downfall of a working society. By optimizing the flow of traffic and keeping congestions and mass accidents from happen, the overall working of cities and public health can be maintained.

The forever evolving methods of traffic control can have much to gain from the implementation of UAVs into their current system. It is common for the use of manned aerial vehicles such as helicopters for traffic control, though these are expensive pieces of equipment, and require an onboard pilot. On the other hand, by utilizing, UAVs it is possible to perform a wide area surveillance of the vehicles and their movement, obtaining information and data that can be used by the drivers to better direct them to their final destination. This aids in better distribution and overall enhances the user's experience [2]. Having this solution for the traffic control problem would not be viable when utilizing manned vehicles. This is because the dependence on human operation would limit the fast and instantaneous response of a UAV system, that is required due to the nature of the data being collected.

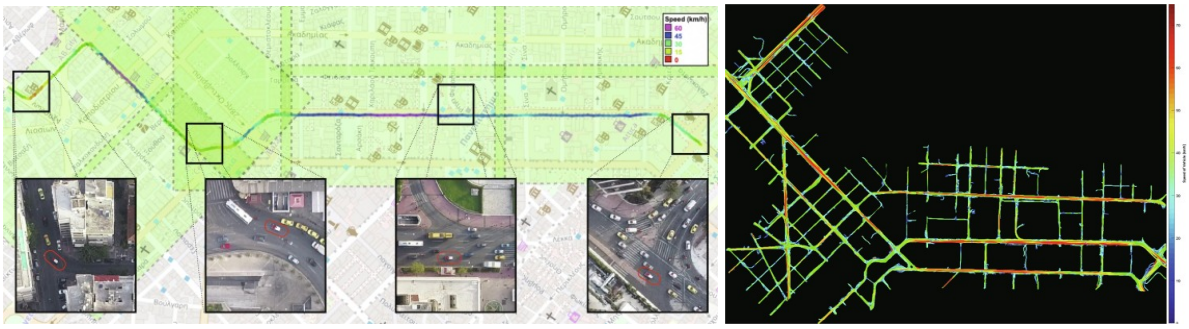

Figure 1.2: Examples of UAV imaging being utilized for vehicle identification, and converting this data to traffic heat maps. Images from [2].

In regard to using UAV for data acquisition in this particular situation, there are some issues that can be tackled by a multimodule approach. For instance, the quality of data acquisition is strongly affected by the conditions of the environment. Dependent on the location or the meteorological conditions (for example, low visibility due to fog), the sensors required might differ. Accordingly, it is evident that relying on RGB cameras alone to identify vehicles is insufficient and that a variety of sensors is needed to achieve accurate results.

In order to further ensure the reliability of data acquisition in adverse weather conditions or low-light scenarios, a wide range of sensors is utilized. LiDAR sensors excel at detailed object detection and identification. Hyperspectral cameras are employed to capture and process information across the full electromagnetic spectrum, resulting in detailed landscape matrix contrast detection [14]. The modular approach proves valuable when dealing with multiple types of sensory information that may or may not be necessary for specific scenarios. Not every task or area in a given city will demand every type of sensor data acquisition. For this reason, the concept of specified modules with different capabilities becomes advantageous.

By employing a modular approach, we ensure that the sensory capabilities of the UAVs will be

sufficient for adverse conditions while also optimizing the allocation of available resources. However, the nature of the task at hand (traffic control) necessitates the use of multiple vehicles, as it relies on the concept of a swarm of diversely distributed vehicles simultaneously collecting data and communicating with each other. Once again, it is evident that the methodology studied and adopted in this thesis can be beneficial to the relevant industry. Communication between modules is a crucial aspect for effectively managing traffic in an area. As we will soon discuss in 1.2, one of the main motivations for using ETC is when working with intercommunicating devices, where communication plays a pivotal role in the system's operation.

1.1.3 UAVs Applied to Territorial Control

Frontex, the European Border and Coast Guard Agency, plays a crucial role in promoting, coordinating, and developing European border management, aligning itself with the European Union (EU) Charter of Fundamental Rights and the concept of integrated border management. An illustrative case in point is the integration of Maritime Aerial Surveillance (MAS) into its operations, solidifying its status as an indispensable service provided to national authorities [15].

The agency, in collaboration with national authorities, is increasingly turning to airborne surveillance to reinforce the monitoring of activities at Europe's maritime borders. Actively seeking state-of-the-art products and services, Frontex is prioritizing investments in un/manned airborne platforms, such as Remotely Piloted Aircraft Systems (RPAS). Additionally, they are exploring advancements in maritime surveillance sensors, encompassing electro-optics, radar sensors, Automatic Identification System (AIS) technology, and mobile phone detection equipment, with the intention of creating specialized vehicles equipped with different sensors for specific tasks [16].

This concerted effort signifies Frontex's commitment to staying at the forefront of technology, ensuring both cost-effectiveness and operational efficiency in their surveillance endeavors. The agency's budget has seen substantial growth in recent years, with an allocation of 845 million euros for the year 2023 [15].

Frontex's devoted dedication to innovation and increased investment in cutting-edge technologies is evident. Their emphasis on unmanned vehicles highlights the significant potential for cost-effective and efficient data extraction in monitoring migration paths and border control issues [16]. Environmental and meteorological conditions play a crucial role in data acquisition, reinforcing the relevance of the modular strategy explored in this thesis. The interest in a wide array of specialized sensors, as mentioned earlier, coupled with substantial investments in aerial vehicle technology, further underscores the importance and demand for the modules examined in this thesis.

The sheer volume of sensors and hardware necessary to equip these Aerial Vehicles (AVs) for robust and effective performance in line with the required tasks highlights the need for a more adaptable approach. A multimodule methodology, where each UAV possesses distinct capabilities yet can seamlessly collaborate as a unified vehicle, proves to be highly advantageous.

1.1.4 UAVs Applied to Agriculture Industries

Another area where UAV technology still has significant potential for growth, due to its tremendous usefulness, is agriculture. By mapping crops, identifying anomalies, assisting with irrigation, or collecting produce, unmanned vehicles can not only aid in tasks traditionally performed by humans, but they can also do them autonomously and more efficiently.

Evaluating the overall health of a crop, and identifying the required measures in real time, ensures a more efficient production. Rot detection is already under investigation, as visual analysis of the entire crop often provides an accurate means of identifying anomalies [17]. A wide range of sensors can once again be utilized in order to equip powerful and reliable UAV. Multiple types of visual sensors such as RGB, Multispectral and Thermal are already being tested and used for collecting overall crop analysis, physiological status of the plants, plants surface temperature and water stress indexes [14].

Furthermore, as previously highlighted, the role of produce harvesting stands as a crucial domain where UAVs exhibit notable advancements [18]. Reinforcing the integration of drones within the agricultural sector and their potential for actuation (not only data acquisition). By outfitting these devices with precise and suitable actuators, a spectrum of harvesting tasks can be efficiently executed. It is important to recognize that varying types of crops require specific detachment methods for the produce. Once again, reinforcing the adaptability requirements that the multimodule approach can present.

Figure 1.3: Examples of UAV agriculture mapping and treating. Images from [3].

Another valuable solution being explored is a self-watering system where the vehicle can predict areas where the crop lacks irrigation using a water supply, RGB camera, and heat sensor system.

This technology is already in use by CONSORTIQ, a company specializing in UAV applications across various industries, including agriculture, landscape mapping, infrastructure inspection, and air quality detection [3].

The multimodule approach, as examined in this thesis, has much to offer the agriculture industry. As previously mentioned, increasing the number of modules can enhance the maximum load capacity for transportation, enabling better watering and care of the crops, as well as greater transportation capacity for produce. Moreover, having specific modules equipped with different sensors and actuators can enhance versatility and reduce costs. If only the necessary hardware is used for a specific task, it allows for more efficient operations with lower energy and computational expenses. Additionally, it leaves the unnecessary modules available for other tasks.

1.2 An Introduction to ETC

Recent developments in computer and communications technologies have led to a novel breed of large-scale resource-constrained control systems. In these systems, it is desirable to limit sensing and control computations and/or communications to those moments when the system needs attention instead of continuous high-demand computation.

The concept of real-time control approaches is based on the assumption that, at any given moment, the controller has access to the entire state space. Using this information, the optimal control input is computed in real time. In the previous paragraph, it was mentioned that there is a desire to limit the computation and communication between the components of the system (such as controller units and different modules). This is a crucial aspect to consider, especially in the context of controlling various UAVs using a modular approach. The two most commonly adopted approaches are time-triggered and event-triggered controllers.

The Time-Triggered Control (TTC) aims to reduce communication and computation time between modules and components of the system in an effort to manage energy and computational power. The TTC operates on a predetermined schedule, where tasks are executed at specific time intervals. This means that regardless of the initial or current state of the system, the control computation is already defined. This makes TTC highly predictable and deterministic, which can aid in the implementation of the full control algorithm. On the other hand, while TTC might ensure reliability and predictability, it may not be the most logical and efficient approach for certain dynamic events with inherent irregularities.

A more robust scheduling approach that can be adopted is the ETC, particularly for irregular systems, i.e., systems where the dynamics are quite unexpected. The occasions when the system may require

re-computation can range from highly predictable to completely unpredictable, where the latter might not be compatible with a TTC approach. The prevalence of unforeseen obstacles and/or errors may indicate that an ETC strategy, where there exists a triggering condition given certain (or multiple) condition(s) being met, may be a better approach.

Having an overall idea of these two types of paradigms, we now move to explain how the ETC can function in practice through a more detailed explanation and the help of some practical real world examples. This approach of controlling cyberphysical systems can be found on various modern day systems, ranging from the more simple endeavors like the temperature control in a room [11, Example 1.9], to a more complex system such as autonomous drone navigation [19].

For the case of controlling the temperature in a room, we can analyze how the event-triggering occurs and how it functions in a real-world scenario. Firstly, we establish the event, which is the set of conditions that lead to the control response. In this case, it would be when the room temperature falls outside the desired range. Next, we define the "Event Detector", which in this case is a thermometer, and in most cases, is some sort of sensor. Following that, the "Decision Logic" consists of a set of rules or algorithms that process the information from the event detector and determine the appropriate response. In this practical example, it involves deciding the best approach, such as turning on the heater if the temperature drops to a certain value. Lastly, the "Control Action" is the specific response taken by the control system in response to the given event, such as activating the heater.

A more rigorous formalization of the ETC is described by the block diagram depicted in Figure 1.4.

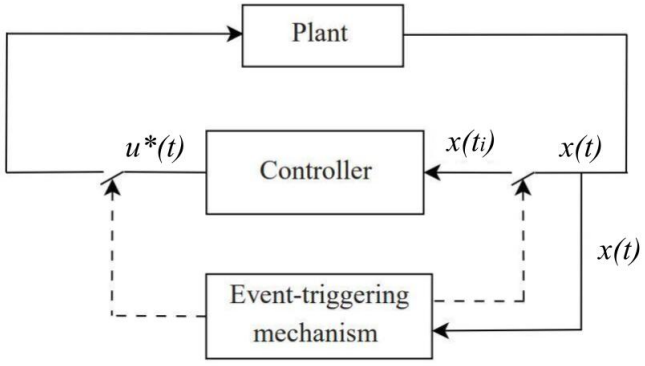

Figure 1.4: Block diagram of an ETC controller. Image from [4].

The diagram illustrates the rationale behind the ETC control system. Upon initial observation, it seems to closely resemble a closed-loop full-state feedback system, similar to the one depicted in Fig-

ure A.1. However, upon closer inspection, it becomes apparent how the event-triggering mechanism interacts with the feedback system. Beginning as an open loop, the plant is supplied with a specific input vector, denoted as $u^*(t)$ for $t \in [t_i; t_{i+1}[$, which drives the state space system until a new specified event occurs. Once this event is confirmed to be true, triggering the conditions that define the ETC mechanism at $t = t_{i+1}$, the loop is closed. The controller unit, defined by its controlling paradigm, then generates a newly defined input vector $u^*(t)$ for $t \in [t_{i+1}; t_{i+2}[$, and the process begins anew.

Having outlined how the ETC functions on a practical level, it becomes evident that what sets apart this methodology from a simpler full-state feedback control is the presence of an event-triggering mechanism that restricts computation and communications to specific, appropriate occasions. This methodology is highly valued for systems that necessitate extensive communication between modules and units, as well as for systems with inherent multimode aspects. For example, in the module-based approach to controlling various UAVs, the dynamics of coupled and free units undergo significant changes. The relationship between input and output for controlling a single vehicle is quite distinct from the dynamics of coupled vehicles. With that said, it is crucial to have this discrete control aspect capable of drastically altering the control law at specific points in time. The event-triggering mechanism typically relies on a sensor of some kind and can calculate a specific condition. If this condition is met, it leads to the so called “Event-Triggering”.

One of the key points to take in consideration when working with ETC is the minimum interval time between two successive events. It must be ensured that the minimum interval time is greater than a certain positive constant. Otherwise, an ETC system will exhibit Zeno behavior, which is unexpected in the definition of the control system. This time parameter should be chosen based on the dynamics of the system being controlled, taking into account factors such as sensor response times, actuator limitations, and the inherent time constants of the processes involved.

1.3 State of the Art

To provide a comprehensive overview of the current advancements and state-of-the-art developments within the scope of this thesis, it is necessary to delve into the main pillars of the control approach. The chosen control strategy depends on the selected control algorithm. In this section, a thorough comparison of the three most commonly used algorithms will be provided. Additionally, we mentioned the utilization of ETC, which stands as one of the most crucial and distinctive aspects of the implementation. An analysis of the present utilization of this technology, coupled with recent examples of its implementation, will be presented to underscore the utility and advantages of this paradigm. Moreover, we will explore the current applications of modular vehicles and highlight how our approach

builds upon and enhances existing instances in this domain. This will further underscore the substantial advancements facilitated by our proposed methodology within the context of modular vehicle control and operation.

1.3.1 Controller Methodology Architectures

When defining a control problem, the most pertinent aspect to take into account is choosing the control algorithm. Selecting the one that best suits the given system is largely dependent on the desired outcome. The most commonly used control strategies are Proportional–Integral–Derivative (PID), Linear Quadratic Regulator (LQR), and Model Predictive Control (MPC).

A PID-based control technique, which consists of proportional, integral and derivative elements, is a commonly used strategy practiced on UAVs, since this controller allows for direct tuning of its parameters given the objective proposed. Also, since the UAVs systems are Multiple Input Multiple Output (MIMO) systems, the PID approach is fitting since it can be handled as a cascaded interconnection of several controllers [10]. When following this control approach firstly, the main objective must be well defined, and then, based on the situation the different controller gains (K_p,K_i,K_d) are defined in a way that ideally suits the problem in question(cf. [20]).

The LQR controller has a great capacity for dealing with MIMO systems and for that reason, just like PID, it is a commonly used strategy when handling UAV platforms. This type of controller revolves around the idea of finding the control variable that minimizes a predefined cost function that, in theory, leads to optimal control. Nevertheless, the LQR algorithm demands that the model be linear, and for that reason it is necessary to linearize the system at the hovering point of the drone in these nonlinear cases. The cost function to be minimized depends on two weight matrices, Q and R. The first matrix determines the weights on state (x) errors, while the latter influences the weights on input (u) values [10].

Finally, the MPC control algorithm has become more often utilized since it offers many advantages where the previous controllers fall short. For instance, this methodology is capable of handling control constraints, state estimation, predictive behavior, noise, and disturbance rejection, and thus became more popular due to this predictive behavior which can aid in reducing the closed loop system uncertainty [10]. Though it offers many useful capabilities, following this control strategy can be a challenging task, when compared to the previous controller implementations. This results in some disadvantages like requiring the plant model to be defined, high computational load, high algorithmic complexity and high number of control parameters [21].

When comparing LQR-based controllers with PID-based controllers, the application of LQR is better

in the context of the study. It shows better performance in tracking and stability, when compared to the latter, and the tuning of a LQR-based controller is considered far simpler because the existence of a solution to the algebraic Riccati equation guarantees the closed-loop stability [10].

When comparing LQR-based controllers with MPC-based controllers, there were some limitations when using MPC. Simulations indicate poorer tracking performance than LQR. One reason behind this can be the fact that LQR provides an infinite time horizon for quadratic cost minimization, while the MPC is always limited to a finite time horizon. Because of the high computational demand, the LQR is found to be more efficient by comparison [10].

1.3.2 Event-Triggered Control

The concept of event-based control can be traced back as far as 1959, when P. Ellis stated, ”Periodic sampling is not required. The most suitable sampling is by transmission of only significant data, such as the new value obtained when the data is changed by a given increment. In certain cases, transmission of data by this means can be used to increase channel capacity” [22]. This novel line of thinking came to improve the state-of-the-art periodic control at the time. The concept became increasingly recognized in the community [23], and much of the interest in today's control systems arose from the works:

- IFAC 1999: Comparison of periodic and event based sampling for first-order stochastic systems [Åström & Bernhardsson, 1999]
- IFAC 1999: A simple event-based PID controller [Årzén, 1999]
- Control Eng. Pract. 1999: Asynchronous measurement and control: A case study on motor synchronization [Heemels et.al, 1999]

During the last decade, event-triggered control (ETC) has become increasingly relevant in real-time control systems. Offering a striking advantage since it provides a strategy in which the control task is executed only when needed. Compared with traditional timed control, ETC can efficiently reduce the number of control tasks executed while maintaining the desired closed-loop performance [24]. This not only ensures an accurate control response, but also minimizes energy and computational costs. This aids the assembly process, requiring less complex and less advanced units due to simpler and scarce calculations [25].

1.3.3 Modular Vehicles

The concept of modular vehicles has seen significant advancements, with notable contributions from the University of Pennsylvania GRASP Lab. Their work, introduced in [1], involved the development

of the ModQuad, a flying modular structure capable of self-assembling in midair. This achievement demonstrated great success and provided a real-world demonstration of the concept of modular aerial vehicles. The results from this work showcased the potential for autonomous reconfiguration in aerial systems.

The Figure 1.1 previously presented showed these quadrotor vehicles in different configurations, as depicted in the work by [1]. The nature of these modules has some particularities that can be considered as problematic for the robustness of the system, and a more detailed image of these can be seen in Figure 1.5.

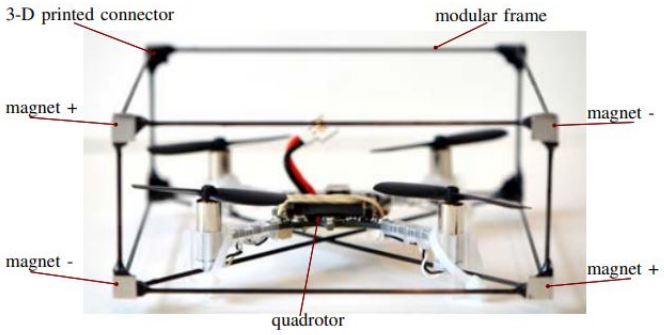

Figure 1.5: Detailed view of the modules built for [1].

Firstly, the method of attachment implies four connection points per pair of connected modules. These connection points were envisioned to work with magnets and have shown great promise, offering a strong connection with scalable intensity (changing the magnetic materials). Another valuable characteristic of these connectors is the magnetic pull that they exert on each other, guiding the modules to the correct arrangement.

On the other hand, as shown in the results and tests conducted in [1], this magnetic force could also work in the opposite situation, where unwanted magnetic forces could compromise the correct connection due to unwanted magnet interaction from unconnected corners.

A final drawback was discovered when considering this approach: the inability to disassemble connected modules. The project achieved very good results when assembling, despite the magnetic issues mentioned, but it was not able to disconnect the modules from each other due to the envisioned design. To create a separation force between two quadrotor modules, they must be able to orient themselves in opposing directions. However, the rigid frame and four-point connection compromised this separation ability.

For tackling the separation obstacle presented in the previous modules, the same research group

devised a solution by re-envisioning the module's frame and magnetic connector positions. In this new configuration found in [5], the connections between modules were kept at one connection point per pair of modules. These new and improved modules are shown in Figure 1.6.

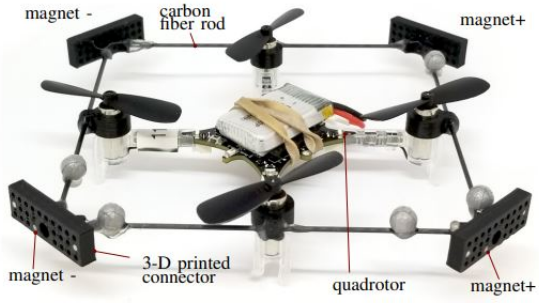

Figure 1.6: Detailed view of the modules built for [5].

The new connection method of one single point of contact allowed for greater flexibility in the relative orientation of the modules, which enabled the modules to create torques and forces of separation due to the more flexible contact point. The process of separation is shown in the images in Figure 1.7.

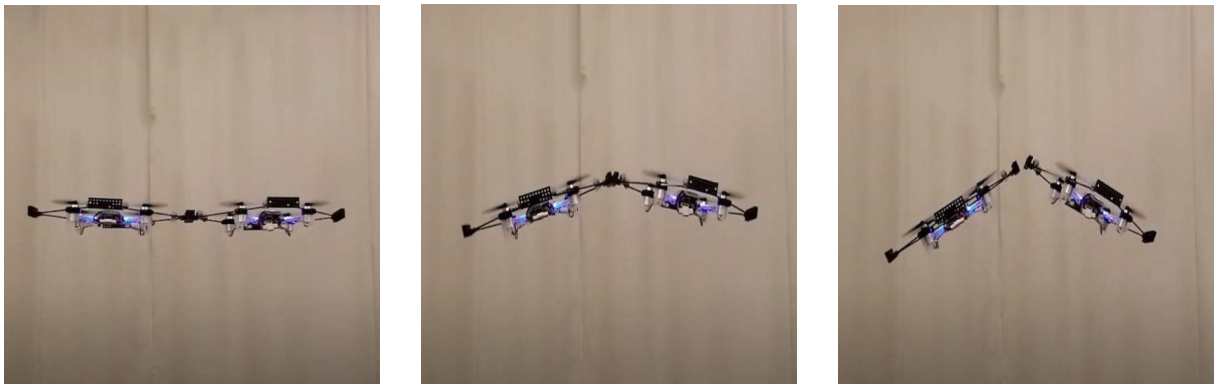

Figure 1.7: Step by step view of disconnecting modules from [5].

Though the new approach solved the disassembling difficulties, the previous problems found in the nature of the (magnetic) connectors were still a drawback to consider.

The existence of four rotors at each corner of the vehicle also makes it more prone to unstable control in the case of malfunctioning of one or more rotors. The more rotors a vehicle relies on, the probability of malfunctioning of one of the components increases, and consequently, the probability of unsteadiness also increases. To tackle this, a solution explored by the research group was the reorganization of the modules depending on which module and which rotor malfunctioned. The results obtained in [26]

showed an improvement in the modular vehicle's stability simply by rearranging the modules in the most stable pattern for the given malfunctions. The results showed significant improvement over the unbalanced modular vehicle where random rotors were at fault.

In this thesis we propose a similar objective as the one from the research group at GRASP labs. One of the contributions and differences when comparing the work done by the team at University of Pennsylvania and this thesis is first the event-triggered control strategy previously mentioned and which we have already talked about. The second and probably the most important contribution is the envisioned new modules. This thesis work relies on the concept of fully actuated modules.

For this particular problem, we envisioned using a coaxial helicopter [6], created by the same research group GRASP Labs, that emulates full actuation over forces and torques (six degrees of freedom) using only two actuators, like the one in Figure 1.8.

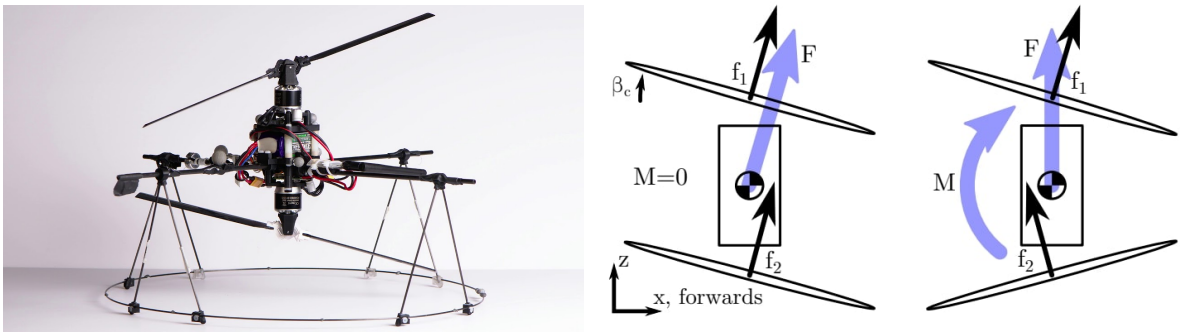

Figure 1.8: Coaxial helicopter, which emulates full actuation over forces and torques using only two actuators. On the left a photo of the vehicle and on the right a diagram that shows the constant orientation of the body. Images from [6]

The distinctive feature of these vehicles lies in their fully actuated design, allowing movement without compromising their orientation, as illustrated in Figure 1.8. Quadrotors are inherently reliant on their orientation for control, making it challenging to have multiple units converge at a desired point while maintaining an upright orientation. Furthermore, attaching them to move as a single unit using attachment tools like magnets can result in various difficulties.

The coaxial helicopter module solves many of the difficulties found in the modules presented by GRASP. More specifically, these modules can exert separating forces without requiring opposing orientations (a problem found in [1]). Additionally, these modules are more stable and resilient to rotor faults since coaxial helicopters can often maintain controlled flight to some extent, whereas a quadrotor may become uncontrollable if it loses a rotor (a problem found in [1] and [5] and attempted to be fixed by [26]). The conclusion drawn from the results in the GRASP study is that many of the drawbacks

and problems found can be addressed by reimagining the problem with the coaxial helicopter modules, shown in Figure 1.8.

The final point to be discussed that arose from analyzing the difficulties found in the studied state-of-the-art literature, was choosing the nature of the attachment mechanism. The mechanism chosen for the ModQuad modules was using magnets. As previously mentioned, the magnetic connectors have many benefits, but on the other hand, may also result in unwanted dynamics and difficulties. Many other solutions were considered, such as hook and loop and male-female port connections. The male-female port connections may require a more precise attachment and have less freedom when defining stronger or weaker connections. On the other hand, strips of hook and loop connectors can have much variety in density and composition, resulting in a wide range of strength [27].

The main downside when considering magnetic connections is the complicated dynamics of the magnetic force induced, introducing unwanted magnetic forces between the modules. With this in mind, the proposed solution and corresponding simulation of the system took into account the physical properties of specific brands of hook and loop connectors [28] to achieve a closer approximation of real-world results.

1.4 Objectives

The main problem statement we envision is “how can we control a swarm of UAVs, that can connect to each other and work as a multimodule vehicle?”. In this thesis, we wish to study the multimodule approach by simulating the control of two vehicles, as they attach, detach and follow specific tasks. We also must ensure that this control strategy is reliable and can resist disturbances and unforeseen errors. In this case, a more complex approach must be taken, yet at the same time, we wish to design an algorithm that is computationally efficient. Having this in mind, there are a lot of factors to be taken into account. Firstly, choosing the control algorithm that best suits the system is one of the first steps. Various types of controllers can be used depending on the context, where some may offer better results and lower estimated errors, and others may require less computing capacity, or use a more economic approach. The simulation results in Chapter 4 show the working fully implemented system, capable of mimicking the dynamics of multiple modules, how they act when separated or coupled and the required dynamics between these two modes.

As mentioned before, the controller of choice is dependent on the situation and the objectives to achieve. From what was stated in Section 1.3 an LQR-based controller was found to be best suited. Though the main algorithm is an LQR, the calculations of the optimal trajectories and input signals will follow an event-triggered approach. Having said this, we find ourselves with what can only be

described as an Event-Triggered Linear Quadratic Controller. This controller is based on the concept of calculating the optimal state and input values for the following instances (calculate for a limited time). The re-calculation of these values will be done either if the calculated instances have run out (time limit reached), or the state has deviated significantly from the objective (desired trajectory). By formulating the system as a hybrid system, or more precisely as a hybrid automaton, it facilitates the existence of multiple modes of operation. These modes including the control of a fully connected modular vehicle or the separated modules.

1.5 Thesis Outline

Having presented the problem statement, the motivation behind it, and introduced the main concepts that will be tackled, the upcoming chapters and sections will delve into a more technical review of the concepts at hand.

In Chapter 2, there is a comprehensive review and explanation of the hybrid system paradigm. The concept of Open Hybrid Automata (OHA) and how it differs from a conventional hybrid system. Finally, closing off this chapter by introducing the concept of event-triggered control of open hybrid automata, and how event-triggered control can be related and implemented via a hybrid system.

Chapter 3 starts by introducing the dynamic properties of the envision UAV modules. In this chapter, a full in-depth state-space representation is proposed for simulating the dynamics of modules, both when these are disconnected and connected. Furthermore, the "Trajectory Generation" section establishes the methodology behind calculating the desired trajectories and corresponding desired inputs that minimize a cost function via an LQR controller. Finishing off this chapter by giving a full understanding of how the event-triggered control system is defined as a hybrid system, specifically an OHA.

Through the usage of MATLAB, the simulations of the final system and its control are shown in Chapter 4, along with the tests made to the efficiency of the control strategy in suboptimal situations. Finishing off the thesis with a conclusion about the designed methodology and the results associated, in Chapter 5.

The Appendix A provides a detailed explanation of the LQR theory and the rationale behind the method, while Appendix B contains additional plots detailing the simulations presented in Chapter 4.

2

ETC of Hybrid Automata

Contents

2.1 Hybrid Systems

Having N modules, each capable of autonomously executing individual tasks, is essential. Furthermore, of greater significance to the thesis objective, is their ability to seamlessly attach to one another, functioning as a unified multimodule vehicle. This objective highlights the necessity of defining these two distinct capabilities as separate operational modes. To achieve a robust automated system adaptable to various starting conditions, it is imperative to integrate both types of operations within a single algorithm.

The idea of having many modes of operation, working for specific moments and conditions, can be solved and implemented in many ways. The proposed methodology for this study is to define the system in question as a hybrid system. A hybrid dynamical system (or hybrid system) is nothing more than a dynamical system that exhibits properties of both continuous and discrete systems.

Considering the change of modes as a response to certain conditions being met, for instance a certain proximity of the modules, or sufficient input values, highlights the usefulness of considering the hybrid properties of the system. Considering the dynamics of the multimode modules as a hybrid system, due to their impulsive dynamics, can facilitate transitions between operating modes. Additionally, the hybrid dynamics also enable the implementation of event-triggered controllers. This allows the controller to calculate the predicted input and state vector only at distinct instances.

The framework of a hybrid system adopted in this thesis, follows the one presented in [11]. Later this model will be related to other useful frameworks like Hybrid Automata and more specifically Open Hybrid Automata (OHA). Firstly, we represent the hybrid system in the form:

$$\begin{cases} x \in C & \dot{x} \in F(x) \\ x \in D & x^+ \in G(x) \end{cases} \tag{2.1}$$

This representation shows that the state of the system, $x \in \mathbb{R}^n$ can change according to a differential inclusion $\dot{x} \in F(x)$ (or the differential equation $\dot{x} = f(x)$, which is a particular case of a differential inclusion) while in the set C, and change according to a difference inclusion $x^+ \in G(x)$ (or difference equation $x^+ = g(x)$, which is a particular case of a difference inclusion) while in the set D. The notation $\dot{x}$ represents the first time derivative of the state x and x^+ represents the value of the state after an instantaneous change. For the sake of simplifying the terminology used in the upcoming sections, we label the behavior of a dynamical system describable by a differential equation or inclusion as “flow”, and the behavior describable by a difference equation or inclusion as “jumps”. Taking this nomenclature into consideration, the following names are given:

- A set $C \subset \mathbb{R}^n$, called the flow set;
- A set-valued mapping $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ with $C \subset \text{dom } F$, called the flow map;
- A set $D \subset \mathbb{R}^n$, called the jump set;
- A set-valued mapping $G : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ with $D \subset \text{dom } G$, called the jump map.

With all the requirements met, the hybrid system is capable of capturing the dynamics of both continuous and discrete systems, within one unified framework. Finally, the defined system with its data as above is represented by the notation $\mathcal{H} = (C, F, D, G)$ or, briefly, by $\mathcal{H}$.

Having presented the definition behind hybrid systems, we shall introduce definitions and assumptions pertinent to the analysis of these systems, with the purpose of finding solutions and proving stability.

From [29, Definition 2.6.1] we take that a subset X of $\mathbb{R}^n$ is said to be open if, for each $x \in X$, there exists $\delta > 0$ such that $B(x; \delta) \subset X$. Given the relationship between open and closed subsets, from [29, Definition 2.6.2] we say that a subset X of $\mathbb{R}^n$ is called closed if its complement $(\mathbb{R}^n \setminus X)$, is open. A subset X of $\mathbb{R}^n$ is said to be bounded if, for each $x \in X$, there exists $r > 0$ such that $X \subset B(x; r)$.

A set by definition is compact if it is found to be simultaneously closed and bounded [29, Theorem 2.6.5].

The set-valued map M is locally bounded at $x \in S$ if there exists a neighborhood U_x of x such that $M(U_x) \subset \mathbb{R}^n$ is bounded. locally bounded (relative to S) if it is locally bounded at each $x \in S$. It is convex-valued if $M(x)$ is convex for each $x \in S$.

Given a subset X of $\mathbb{R}^m$, a set-valued map $M : X \rightrightarrows \mathbb{R}^n$ is said to be outer semicontinuous (relative to X) if its graph, given by

$$\text{gph } M := \{(x, y) \in X \times \mathbb{R}^n : y \in M(x)\},$$

is closed (relative to $X \times \mathbb{R}^n$).

According to [11, Assumption 6.5] we name the following set of requirements as the hybrid basic conditions:

(A1) The sets C and D are closed;

(A2) $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is outer semicontinuous and locally bounded relative to C, $C \subset \text{dom } F$, and $F(x)$ is convex for every $x \in C$;

(A3) $G : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is outer semicontinuous and locally bounded relative to D, $D \subset \operatorname{dom} G$.

When studying and working with hybrid systems, there are some important preliminary notations to take account. Starting off by defining hybrid time domain.

Definition 1. *[11, Definition 2.3] A subset $E \subset \mathbb{R}_{\geq 0} \times \mathbb{N}$ is a compact hybrid time domain if*

$$E = \bigcup_{j=0}^{J-1} ([t_j, t_{j+1}], j)$$

for some finite sequence of times $0 = t_0 \leq ... \leq t_j$. It is a hybrid time domain if for all $(T, J) \in E$, $E \cap ([0, T] \times \{0, 1, ..., J\})$ is a compact hybrid domain.

Having defined the concept of hybrid time domain, we may introduce the concept of a hybrid arc also from [11].

Definition 2. *[11, Definition 2.4] A function $\phi : E \to \mathbb{R}^n$ is a hybrid arc if E is a hybrid time domain and if for each $j \in \mathbb{N}$, the function $t \mapsto \phi(t, j)$ is locally absolutely continuous on the interval $I^j = \{t : (t, j) \in E\}$.*

Lastly, the one missing concept left to be defined is that of a solution to a hybrid system.

Definition 3. *[11, Definition 2.6] A hybrid arc ϕ is a solution to the hybrid system (C,F,D,G) if $\phi(0, 0) \in \bar{C} \cup D$, and*

(S1) for all $j \in \mathbb{N}$ such that $I^j := \{t : (t, j) \in \operatorname{dom} \phi\}$ has nonempty interior

$$\begin{aligned} &\phi(t, j) \in C && \text{for all } t \in \text{int } I^j, \\ &\phi(t, j) \in F(\phi(t, j)) && \text{for almost all } t \in I^j, \end{aligned}$$

(S2) for all $(t, j) \in \operatorname{dom} \phi$ such that $(t, j + 1) \in \operatorname{dom} \phi$,

$$\begin{aligned} &\phi(t, j) \in D, \\ &\phi(t, j + 1) \in G(\phi(t, j)). \end{aligned}$$

2.2 Open Hybrid Automata

The purpose of this section is to present a model of the system as an Open Hybrid Automaton (OHA). Though it is worth mentioning that the notions of Event-Triggered Control (ETC) mechanisms will be addressed in the following Section 2.3. Before delving into this concept, it is necessary to have a strong understanding of the conventional Hybrid Automaton. These systems with their continuous and discrete states can usually be represented by a set Q, and for each $q \in Q$, a flow set $C_q \subset \mathbb{R}^n$, a flow map $F_q : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$, a jump set $D_q \subset \mathbb{R}^n$, and a jump map $G_q : \mathbb{R}^n \rightrightarrows Q \times \mathbb{R}^n$. The suggestive form to represent such a system, parallel to (2.1), is

$$\begin{cases} z \in C_q & \dot{z} \in F_q(z) \\ z \in D_q & (q,z)^+ \in G_q(z), \end{cases} \tag{2.2}$$

yet contrary to (2.1), in this case it is considered that different modes are modeled by distinct jumping and flowing dynamics. We define the state, sets and set-valued maps as,

$$x = \begin{pmatrix} q \\ z \end{pmatrix} \in \mathbb{R}^{n+1} \tag{2.3}$$

and

$$C = \bigcup_{q \in Q} (\{q\} \times C_q) \qquad F(x) = (0, F_q(z))$$
$$D = \bigcup_{q \in Q} (\{q\} \times D_q) \qquad G(x) = G_q(z)$$

The definition of Hybrid Automata is taken from the main literature [11]. These systems with explicit "discrete states" or "logical modes" where, in each logical mode, different jump maps are specified on different subsets of a jump set, or where the jumps are modeled by an automaton, can also be molded into the framework of (2.1). Such systems are usually given by

- A set of modes Q, which is identified here with $\{1, 2, \ldots, q_{\text{max}}\}$;
- A domain mapping Domain, giving for each $q \in Q$ a set $\mathrm{Domain}(q)$ in which the continuous state z may evolve;
- A flow map $f : Q \times \mathbb{R}^n \to \mathbb{R}^n$, which describes the continuous evolution of the continuous state variable z; in fact, it is enough that $f(q, \cdot)$ be defined on $\mathrm{Domain}(q)$, for each $q \in Q$;
- A set of edges $E \subset Q \times Q$, identifying pairs (q, q') such that a transition from q to q' is possible;

- Guard conditions which identify, for each edge $(q, q') \in E$, the set $\mathrm{Guard}(q, q')$ to which the continuous state z has to belong for transitions from q to q' to be enabled;
- Reset map $\mathrm{Reset} : E \times \mathbb{R}^n \to \mathbb{R}^n$, which describes, for each edge $(q, q') \in E$ and continuous state $z \in \mathbb{R}^n$, the jump of the continuous state during a transition from q to q'; in fact, it is enough for $\mathrm{Reset}(q, q', \cdot)$ to be defined on $\mathrm{Guard}(q, q')$.

To capture the dynamics resulting from such a set of data in the format (2.2), for each $q \in Q$, consider

$$\begin{aligned} C_q &= \mathrm{Domain}(q), \\ F_q(z) &= f(q, z) && \forall z \in C_q, \\ D_q &= \bigcup_{(q,q') \in E} \mathrm{Guard}(q, q'), \\ G_q(z) &= \bigcup_{(q' : z \in \mathrm{Guard}(q,q'))} \begin{pmatrix} q' \\ \mathrm{Reset}(q, q', z) \end{pmatrix} && \forall z \in D_q. \end{aligned}$$

The values of F_q and G_q outside of C_q and D_q, respectively, can be taken to be empty. Such a definition of G_q naturally introduces set-valuedness. Indeed, $G_q(z)$ is a set whenever z is an element of two different guard sets $\mathrm{Guard}(q, q')$ and $\mathrm{Guard}(q, q'')$. In fact, $G_q(z)$ is a set in such a case even when all reset maps are identities, in other words, when z does not change during jumps.

The Hybrid Automata described, can be used to model closed-loop systems, but they do not provide a suitable hybrid automata model for plants, because these do not encompass their inputs and outputs. In order to tackle this limitation, we resort to the concept of Open Hybrid Automata that is introduced in [30, Section 3.13.1] and which we adapt to the framework of hybrid dynamical systems in this section.

Similarly to the standard Hybrid Automaton, the data of an OHA also includes a directed graph $G = (Q, E)$ that describes possible mode transitions. However, the dynamics and the domain, guard and reset maps of the open hybrid automaton are modified as follows in order to encompass the influence of an input $u \in \mathbb{R}^k$:

$$\begin{aligned} f^\circ &: Q \times \mathbb{R}^n \times \mathbb{R}^k \to \mathbb{R}^n \\ \mathrm{Domain}^\circ &: Q \rightrightarrows \mathbb{R}^n \times \mathbb{R}^k \\ \mathrm{Guard}^\circ &: E \rightrightarrows \mathbb{R}^n \times \mathbb{R}^k \\ \mathrm{Reset}^\circ &: E \times \mathbb{R}^n \times \mathbb{R}^k \rightrightarrows \mathbb{R}^n \times \mathbb{R}^k, \end{aligned} \tag{2.4}$$

where the superscript in f°, Domain°, Guard° and Reset° emphasizes the fact that this data characterizes an OHA. Lastly, the data of an OHA also includes a function $h : Q \times \mathbb{R}^n \to \mathbb{R}^m$ such that the output of the system is given by $y = h(q, x)$ for each $(q, x) \in Q \times \mathbb{R}^n$.

An Open Hybrid Automaton can be described as a hybrid dynamical system with inputs (cf. [31]) as follows:

$$\begin{cases} \dot{q} = 0 \\ \dot{x} = f^{\circ}(q, x, u) \\ \begin{pmatrix} q^{+} \\ x^{+} \end{pmatrix} \in G^{\circ}(q, x, u) := \bigcup_{\{q' \in Q : (x,u) \in \mathrm{Guard}^{\circ}(q,q')\}} \begin{pmatrix} q' \\ \mathrm{Reset}^{\circ}(q, q', x, u) \end{pmatrix} \end{cases} \begin{matrix} \\ (q, x, u) \in C^{\circ} \\ (q, x, u) \in D^{\circ} \end{matrix} \tag{2.5}$$

where

$$C^{\circ} := \{(q, x, u) \in Q \times \mathbb{R}^n \times \mathbb{R}^k : (x, u) \in \mathrm{Domain}^{\circ}(q)\} \tag{2.6a}$$

$$D^{\circ} := \bigcup_{q \in Q} \left(\{q\} \times \{(x, u) \in \mathbb{R}^n \times \mathbb{R}^k : (x, u) \in \mathrm{Guard}^{\circ}(q, q') \text{ for some } (q, q') \in E\} \right). \tag{2.6b}$$

In a simplified manner, the jump dynamics, or discrete properties of the OHA, can be summarized in the definition of the jump set D° and jump map, G°. For all values of $(q, x, u) \in D^{\circ}$ the OHA dynamics are discrete, and $G^{\circ}(q, x, u) \neq \emptyset$. In other words, the jump set is nothing more than the collection of all possible state, x, input, u, and mode, q, combinations that result in a change of mode. The jump map, G°, is nothing more than the defined response to the OHA's state to a jump. In the case of collision or detachment, there are certain state and mode values to be reassigned (mapped by the Reset°).

The flow dynamics, or continuous properties of the OHA, can be summarized in the definition of the flow set C° and flow map, f°. For all values of $(q, x, u) \in C^{\circ}$ for which the system dynamics are continuous, and $f^{\circ}(q, x, u) \neq \emptyset$. From an informal standpoint, this set encompasses any possible situation where the vehicles are not attaching nor detaching. The flow map, f°, is nothing more than the first-order derivative of the state vector, x, for any and all instances of continuous dynamics (for all $(q, x, u) \in C^{\circ}$).

With all being said, an OHA is identified by its data, as:

$$\mathcal{H}_a^{\circ} := (Q, E, f^{\circ}, \mathrm{Domain}^{\circ}, \mathrm{Guard}^{\circ}, \mathrm{Reset}^{\circ}, h),$$

and, for the sake of regularity, we take Assumption 1.

Assumption 1. *Given an Open Hybrid Automaton* $\mathcal{H}_a^{\circ} := (Q, E, f^{\circ}, \mathrm{Domain}^{\circ}, \mathrm{Guard}^{\circ}, \mathrm{Reset}^{\circ}, h)$, *the following hold:*

1. $\mathrm{Domain}^{\circ}(q)$ *is closed for each* $q \in Q$*;*

2. f° *is continuous;*

3. $\mathrm{Guard}^\circ(q, q')$ *is closed for each* $(q, q') \in E$*;*

4. Reset° *is outer semicontinuous and locally bounded relative to* D°*;*

Remark 1. *The data of an open hybrid automaton in [30, Section 3.13.1] also includes the set of initial states which we have omitted because we consider it to be the whole state space* $Q \times \mathbb{R}^n$*.*

Given a solution, ϕ, to a hybrid system, $\mathcal{H}$ for a given hybrid time E. The solution is said to be complete if $\mathrm{dom}\,\phi$ is unbounded [11, Definition 2.5]. The solution is said to be maximal, $\phi \in S_{\mathcal{H}}(S)$, if there does not exist another solution ψ to $\mathcal{H}$ such that $\mathrm{dom}\,\phi$ is a proper subset of $\mathrm{dom}\,\psi$ and $\phi(t, j) = \psi(t, j)$ for all $(t, j) \in \mathrm{dom}\,\phi$ [11, Definition 2.7]. It is apparent that complete solutions are maximal, but the converse statement is not true.

The following definition of the execution of an open hybrid automaton is adapted from [30, Definition 3.16].

Definition 4. *An execution of an open hybrid automaton with initial condition* $(q_0, x_0) \in Q \times \mathbb{R}^n$ *is a tuple* $\chi := (\mathcal{T}, q, x, u, y)$ *satisfying the following:*

1. $(t, j) \mapsto (q, x, u)(t, j)$ *is a solution to* (2.5) *in the sense of [31, Definition 2.29];*

2. $\mathcal{T} := \{t_j\}_{0 \le j \le J}$ *with* $J := \sup_j \mathrm{dom}(q, x, u)$ *is the collection of jump times, i.e.,* $(t_{j-1}, j-1), (t_{j-1}, j) \in \mathrm{dom}(q, x, u)$ *for each* $j \in \{1, 2, \ldots, J\}$*;*

3. $y(t, j) = h((q, x)(t, j))$ *for each* $(t, j) \in \mathrm{dom}(q, x, u)$*.*

We say that an execution χ *of an open hybrid automaton with initial condition* $(q_0, x_0) \in Q \times \mathbb{R}^n$ *is maximal if* $(t, j) \mapsto (q, x, u)(t, j)$ *is a maximal solution to* (2.5)*. If* $(t, j) \mapsto (q, x, u)(t, j)$ *is complete, then we also say that* χ *is complete.*

2.3 Event-Triggered Control of a OHA

The previous section explained the definition proposed for OHA, computing the dynamics of the hybrid system at hand. Though the definition of the maps (f° and G°) and sets (C° and D°) simulate the flowing and continuous dynamics of the physical system at hand, the proposed solution to the control has a second discrete component, due to the event-triggered methodology. The reassignment of expected input and trajectory values ($u^\star$ and $x^\star$) can be modeled as a discrete component which in itself has a jump map and jump set, depicting the ETC dynamics, and a new flow set, C that takes the previous set

(C°) from (2.6a).

Assumption 2. *Given an open hybrid automaton* $\mathcal{H}_a^\circ := (Q, E, f^\circ, \mathrm{Domain}^\circ, \mathrm{Guard}^\circ, \mathrm{Reset}^\circ, h)$ *with* $h(q,x) = (q,x)$ *for each* $(q,x) \in Q \times \mathbb{R}^n$*, there exists a unique complete execution* χ *to* $\mathcal{H}_a^\circ$*.*

Given an open hybrid automaton satisfying Assumption 2, one may represent the state and input trajectories of a particular execution as functions not only of hybrid time, but also of the initial conditions (q_0, x_0) as follows:

$$t \mapsto \begin{pmatrix} x^\star(t,j;q_0,x_0) \\ u^\star(t,j;q_0,x_0) \end{pmatrix} \qquad \forall (t,j) \in \mathrm{dom}(q^\star, x^\star, u^\star)$$

due to the uniqueness of complete executions to $\mathcal{H}_a^\circ$. Under the same assumption, it follows that

$$\begin{aligned} x^\star(t,j;q_0,x_0) &= x^\star(t-t_j, 0; q(t_j,j), x(t_j,j)) \\ u^\star(t,j;q_0,x_0) &= u^\star(t-t_j, 0; q(t_j,j), x(t_j,j)) \end{aligned}$$

for each $t \in I^j := \{t \geq t_j : (t,j) \in \mathrm{dom}(q,x,u)\}$ and each $j \in \{1,2,\ldots,J\}$. Therefore, in order to follow an execution it is not necessary to keep track of the number of jumps, but merely the time since the last jump $\tau = t - t_j$ and the state of the hybrid automaton at that point $(q,x)(t_j,j)$. For this reason, and with a slight abuse of notation, we represent a segment of desired trajectories ($x^\star$) and desired input ($u^\star$) of a particular execution with initial condition $(q,x) \in Q \times \mathbb{R}^n$ as follows:

$$\tau \mapsto \begin{pmatrix} x^\star(\tau;q,x) \\ u^\star(\tau;q,x) \end{pmatrix}$$

for each $\tau \in [0,T]$ following the convention adopted in [11] where $T \in \mathbb{R}_{\geq 0}$ is the time of the jump.

The proposed controller design is motivated by the goal of reducing the number of transmissions between the controller and the actuator. To achieve this, we propose the transmission of the control signal $u^\star$ as a function of time, to the actuator during initialization and during the following events: when the distance between the desired state trajectory $x^\star$ and the current state of the hybrid automaton exceeds a pre-specified threshold δ; or when a timer variable, which keeps track of the time since the last reset of the hybrid automaton, reaches its timeout.

More formally, the state of the hybrid closed-loop system $\mathcal{H} := (C, F, D, G)$ which implements the logic described above is denoted by $\xi := (q, x, \hat{x}, \tau) \in \Xi := Q \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}_{\geq 0}$, where $(q,x) \in Q \times \mathbb{R}^n$ denotes the operating mode and the state of the hybrid automaton $\mathcal{H}_a^\circ$, $\hat{x} \in \mathbb{R}^n$ is a memory variable that stores the value of the state x at jumps, and $\tau \in \mathbb{R}_{\geq 0}$ is the timer variable.

The flows of the hybrid closed-loop system are described as follows:

$$\dot{\xi} = f(\xi) := \begin{pmatrix} 0 \\ f^{\circ}(q, x, u^{\star}(\tau; q, \hat{x}) \\ 0 \\ 1 \end{pmatrix} \qquad \xi \in C \tag{2.7}$$

where

$$C := \{\xi \in \Xi : \tau \in [0, T(q, \hat{x})],\ |x - x^{\star}(\tau; q, \hat{x})| \leq \delta,\ (q, x, u^{\star}(\tau; q, \hat{x})) \in C^{\circ}\}, \tag{2.8}$$

with continuous $T(q, \hat{x})$. As can be verified in (2.7), the system is allowed to flow according to the dynamics specified by f as long as the state ξ belongs to C, i.e., if the timer τ belongs to $\tau \in [0, T(q, \hat{x})]$, the distance between the current value of the state $x \in \mathbb{R}^n$ and the trajectory $\tau \mapsto x^{\star}(\tau; q, \hat{x})$ does not exceed the parameter $\delta > 0$ and the combination of the values of the state and of the current input belong to the domain of the current operating mode, as specified in the given open hybrid automaton $\mathcal{H}_a^{\circ}$. The variable $\hat{x} \in \mathbb{R}^n$ stores the value of the state $x \in \mathbb{R}^n$ at the start of the trajectory $x^{\star}$. More precisely, $\hat{x}$ should be equal to $x^{\star}(0; q, \hat{x})$ save for poor initialization of the controller variables. The state variable $x \in \mathbb{R}^n$ evolves during flows according to the continuous dynamics specified in $\mathcal{H}_a^{\circ}$ considering that the input signal is $\tau \mapsto u^{\star}(\tau; q, \hat{x})$.

The jumps of the hybrid closed-loop system consist of the union of two distinct jump maps and jump sets as follows

$$\xi^{+} \in G(\xi) := G_1(\xi) \cup G_2(\xi) \qquad \xi \in D := D_1 \cup D_2. \tag{2.9}$$

with the underlying assumption that $G_i(\xi) = \emptyset$ for each $\xi \notin D_i$ and each $i \in \{1, 2\}$. In the sequel, we describe each component of the jump map and jump set in detail.

The pair (G_1, D_1) represents the event-triggered mechanism and it is given by

$$G_1(\xi) := \{(g_q, g, \hat{g}, g_\tau) \in \Xi : g_q = q,\ g = \hat{g} = x,\ g_\tau = 0\} \quad \forall \xi \in D_1 \tag{2.10}$$

where

$$D_1 := \{\xi \in \Xi : \tau \geq T(q, \hat{x}) \text{ or } |x - x^{\star}(\tau; q, \hat{x})| \geq \delta\}. \tag{2.11}$$

When either the timer τ exceed the limit set by $T(q, \hat{x})$ or when the state of the system deviates by an amount δ with respect to $x^{\star}$, the jump dynamics described by (2.10) do the following:

- Update the state of the memory variable $\hat{x} \in \mathbb{R}^n$ to match the current value of the state $x \in \mathbb{R}^n$;
- Keep the current operating mode $q \in Q$ unchanged;

- Reset the timer $\tau \in \mathbb{R}_{\geq 0}$ to 0.

These updates trigger the computation of a new execution $\chi = (\mathcal{T}, q^\star, x^\star, u^\star, y)$ of $\mathcal{H}_a^\circ$ with initial condition (q, x).

The pair (G_2, D_2) represents the jump dynamics of the open hybrid automaton $\mathcal{H}_a^\circ$. This refers to the jump dynamics introduced in Section 2.2, which, in other words, means mode switching, and they are given by:

$$G_2(\xi) := \{(g_q, g, \hat{g}, g_\tau) \in \Xi : (g_q, g) \in G^\circ(q, x, u^\star(\tau; q, \hat{x})),\ \hat{g} = g,\ g_\tau = 0\} \tag{2.12}$$

for each $\xi \in D_2 := \{\xi \in \Xi : (q, x, u^\star(\tau; q, \hat{x})) \in D^\circ\}$, where D° is given by (2.6b).

Before delving into finding the proof of pre-asymptotic stability of a hybrid closed loop, we must define the concept of uniformly pre-attractiveness, strongly forward pre-invariance and stability.

A compact set $\mathcal{A} \subset \mathbb{R}^n$ is said to be uniformly pre-attractive from a set $S \subset \mathbb{R}^n$ if every $\phi \in S_\mathcal{H}(S)$ is bounded and for every $\epsilon > 0$ there exists $\tau > 0$ such that $\mid \phi(t, j) \mid_\mathcal{A} \leq \epsilon$ for every $\phi \in S_\mathcal{H}(S)$ and $(t, j) \in \operatorname{dom} \phi$ with $t + j \geq \tau$ [11, Definition 6.24].

Let us call a set, $S \subset \mathbb{R}^n$, strongly forward pre-invariant if for every $\phi \in S_\mathcal{H}(S)$, the range of ϕ is a subset of S [11, Definition 6.25]. Where rge is defined as the range of a subset

Theorem 1. *If Assumptions 1 and 2 hold, and there exists a compact set $\mathcal{X} \subset \mathbb{R}^n$ such that*

$$\mathcal{A} := \{\xi \in \Xi : \hat{x} \in \mathcal{X},\ \tau \leq T(q, \hat{x}),\ x = x^\star(\tau; q, \hat{x})\}$$

is strongly forward pre-invariant and uniformly pre-attractive for $\mathcal{H}$, then $\mathcal{A}$ is pre-asymptotically stable for $\mathcal{H}$.

Proof. • $\mathcal{H}$ is nominally well-posed, since it satisfies the hybrid basic conditions;

It follows from Assumption 2 and the definition of an execution of an open hybrid automaton that $x^\star$ is continuous. It follows from Assumption 1 that C° is closed, and, since the Euclidean norm and T are continuous, then the flow set C in (2.8) is closed. Using similar arguments, it is also possible to prove that D° is closed due to Assumptions 2 and 1. It follows from continuity of f° in Assumption 1 and the continuity of $u^\star$ that the flow map (2.7) is outer semicontinuous, locally bounded and a singleton, hence convex-valued. The map G_1 in (2.10) is continuous, hence outer semicontinuous and locally bounded. It follows from Assumption 1 that G_2 in (2.12) is outer semicontinuous and locally bounded relative to D_2, hence we conclude that G is outer semicontinuous and locally

bounded relative to D.

- It follows from continuity of $x^\star$ and T that $\mathcal{A}$ is compact;

  To prove that $\mathcal{A}$ is compact, we need to prove that it is both closed and bounded. Closedness of $\mathcal{A}$ follows from the continuity if T and $x^\star$. To see this, let $\gamma(\xi) \coloneqq (T(q,\hat{x}) - \tau, x - x^\star(\tau; q, \hat{x}))$. Then, $\mathcal{A}$ can be rewritten as

  $$\mathcal{A} = \{\xi \in \Xi : \hat{x} \in \mathcal{X}, \gamma^{-1}([0,+\infty) \times \{0\})\}$$

  and $\gamma^{-1}([0,+\infty) \times \{0\})$ is the preimage of a closed set through a continuous map, hence it is closed [32, Lemma 2.7]. Compactness of $\mathcal{X}$ implies that $\mathcal{X}$ is closed, hence $\mathcal{A}$ is the intersection of two closed sets, thus it is also closed. To prove that $\mathcal{A}$ is bounded, notice that Q is a finite discrete set, thus it is also compact. It follows from compactness of $\mathcal{X}$ and continuity of T that $T(Q \times \mathcal{X})$ is compact, thus $\tau < +\infty$ for each $\xi \in \mathcal{A}$. In particular, we have that $\mathcal{S} \coloneqq \{(q,\hat{x},\tau) \in Q \times \mathcal{X} \times \mathbb{R}_{\geq 0} : \tau \leq T(q,\hat{x})\}$ is compact. Since $x^\star$ is continuous, $x^\star(\mathcal{S})$ is compact. Finally, since $x \in x^\star(\mathcal{S})$ for each $\xi \in \Xi$, we conclude that $\mathcal{A}$ is bounded, which concludes the proof that it is compact.

- From the assumptions in the theorem's statement, we prove pre-asymptotic stability of $\mathcal{A}$ using [11, Proposition 7.5].

□

Definition 5. *[11, Definition 3.38] A function $\beta : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a class-$\mathcal{KL}$ function, also written $\beta \in \mathcal{KL}$, if it is nondecreasing in its first argument, nonincreasing in its second argument, $\lim_{r \to 0^+} \beta(r,s) = 0$ for each $s \in \mathbb{R}_{\geq 0}$, and $\lim_{s \to \infty} \beta(r,s) = 0$ for each $r \in \mathbb{R}_{\geq 0}$.*

If each maximal solution to $\mathcal{H}$ is complete, then solutions to $\mathcal{H}$ inherit the properties of the executions of the open hybrid automaton $\mathcal{H}_a^\circ$. For example, taking into consideration Definition 5, if we construct an algorithm that finds complete bounded executions $\chi^\star = (\mathcal{T}, q^\star, x^\star, u^\star, y)$ to $\mathcal{H}_a^\circ$ such that

$$|(q^\star, x^\star)(t,j)|_{\mathcal{A}^\circ} \leq \beta((q^\star, x^\star)(0,0), t+j)$$

for some class-$\mathcal{KL}$ function β, then each maximal solution to $\mathcal{H}$ from $\mathcal{A}$ satisfies

$$|(q,x)(t,j)|_{\mathcal{A}^\circ} \leq \beta((q,x)(0,0), t+j).$$

3

ETC of Modular Aerial Vehicles

Contents

3.1 General Model for an Aerial Vehicle

In this section, the general model for the dynamics of a modular aerial vehicle will be presented and defined as a state space system. An adaptation from the state space representation presented in [33], for a standard UAV where the rotors are fixed-pitch, rigid rotors, and thrust control is obtained through control of the torque to the motors, is defined as

$$\begin{cases} \dot{\xi} = v \\ m\dot{v} = mge_3 + F \\ \dot{R} = R\mathsf{sk}(\Omega) \\ \mathbf{I}\dot{\Omega} = -\Omega \times \mathbf{I}\Omega + \Gamma \end{cases} \tag{3.1}$$

Letting $\xi = (x, y, z)$, the position of the center of mass of the vehicle, in the inertial frame ($\mathcal{I}$) relative to a fixed origin, $v \in \mathcal{I}$ denote the linear velocity expressed in the same inertial frame and $\Omega \in \mathcal{A}$ denote the angular velocity of the airframe expressed in the body fixed frame ($\mathcal{A}$). Let m be the mass of the vehicle, g the gravitational acceleration and e_3 denotes the vertical direction downwards into the earth. The orientation of the rigid body is given by an orthogonal rotation matrix $R : \mathcal{A} \to \mathcal{I}$, mapping the fixed body frame to the inertial frame . The three dimensional vector $F \in \mathcal{I}$ combines the principal non-conservative forces applied to the vehicle, in the body fixed frame. The notation $\mathsf{sk}(\Omega)$ denotes the skew-symmetric matrix such that $\mathsf{sk}(\Omega)v = \Omega \times v$ for the vector cross-product $\times$ and any vector $v \in \mathbb{R}^3$. Let $\mathbf{I} \in \mathbb{R}^{3\times3}$ denote the constant inertia matrix around the centre of mass (expressed in the body fixed frame) and $\Gamma \in \mathcal{A}$ the torque derived from differential rotors thrust.

The general model in (3.1) serves to illustrate the complexity of the full state space system of an aerial vehicle, encompassing six degrees of motion. Given that the primary objective of this thesis is to control the modules in an endeavor to simulate the concept of a modular vehicle, it is in our best interest to propose a simplified version (reducing to three degrees of freedom: motion in x-axis, motion in the y-axis, and yaw rotation) of the state space system. This will allow us to further concentrate on the control problem at hand.

3.2 Two Dimensional Simplification

In the previous section, the general model for the dynamics of an aerial vehicle was presented and defined as a state space system. To simplify the modeling of the vehicles presented at the beginning of this chapter, certain assumptions were made, including:

- Aerodynamic drag forces are negligible;

- All vehicle modules are equivalent, having the same properties such as mass;
- Other external disturbances are neglected;
- The modules, due to their nature, are considered to be at an upright orientation collinear with the z-axis;
- The modules are considered to have three degrees of freedom.

The system under consideration is three-dimensional; however, we have assumed the constant altitude in the z-axis, to be fixed. Consequently, the system can be treated as a two-dimensional system. The assumption of invariance along the z-axis implies that the gravitational force F_g is constantly being canceled by the thrust, assuming a closed-loop control dedicated to maintaining a constant altitude.

Furthermore, the upright orientation, as previously mentioned, further emphasizes the notion of viewing the system as two-dimensional. This is because roll and pitch motions were constrained to zero, and only the angular motion in the yaw direction was taken into account, as it acts in the XY plane.

Having said this, the simplified version of the state space representation is devoid of a body-fixed frame, keeping every two-dimensional variable defined in the inertial frame. Consequently, there is a lack of a rotation matrix. Lastly, the angular positions and velocities are unidimensional variables.

By analyzing the forces and torques acting on each module, we can describe how the dynamics of the system work and define a state space representation of this system. We shall elucidate the fundamental dynamics guiding the physical behavior of the vehicles through an in-depth breakdown of the underlying equations governing the movement of each module.

3.2.1 Definition of the Dynamics of Two Independent Modules

The Figure 3.1 depicts the system diagram illustrating two disconnected modules, in other words, two modules independent of each other. The diagram showcases the two vehicles, along with the position vectors (p_1 and p_2) representing the positions of each module's center of mass.

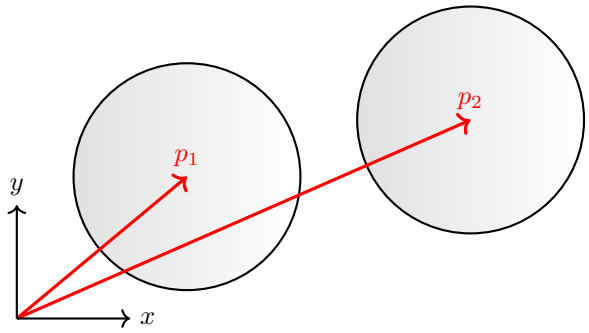

Figure 3.1: System diagram illustrating the position vectors (p_1 and p_2) of the two disconnected modules.

Newton's second law for motion and rotation of each module plays a major role in establishing the dynamics of the system for a simplified depiction of the two modules. These equations describe the connection between the module's motion and rotation and the forces and torques acting on it. According to the motion equation, the module's mass times its acceleration equals the total of all external forces acting on it, in this case the sole force acting is the input force from the motor (independent of the second module). The rotation equation takes into consideration all torques acting on the module, and once again it is independent of the second module's dynamics. These equations offer a thorough knowledge of the dynamics of the first module when taken together. The mentioned equations that define the dynamics of the first module are,

$$m\ddot{p}_1 = F_1 \tag{3.2}$$

$$J\ddot{\theta}_1 = M_1 \tag{3.3}$$

where $m \in \mathbb{R}$ is the mass of each module, $J \in \mathbb{R}$ is the inertia (with respect to the center of mass) of each module, $p_1 \in \mathbb{R}^2$ is the position of the center of mass of the first module, $\theta_1 \in \mathbb{R}$ is the orientation of the first module, $F_1 \in \mathbb{R}^2$ is the force applied to the first module and $M_1 \in \mathbb{R}$ is the torque applied to the first module. The dynamics of the second module are obtained in the same manner, and a state system representation of a two vehicle system can be defined as,

$$\begin{cases} \dot{p}_i = v_i \\ \dot{v}_i = \dfrac{F_i}{m} \\ \dot{\theta}_i = \omega_i \\ \dot{\omega}_i = \dfrac{M_i}{J} \end{cases} \qquad \text{with } i \in \{1, 2\}$$

3.2.2 Definition of the Dynamics of Two Connected Modules

The modular dynamics of the system are defined by the interaction of the modules with each other when these are attached to each other. Contrary to the previously explained system in Section 3.2.1, the dynamics of a module connected to another is not solely dependent on its own inputs, since the actuation and motion of the connected module will interfere. The following expressions will better explain how two connected modules interact, by explaining the contact force between two modules in the x and y coordinates, and its impact on their physical positioning. By analyzing the forces and torques acting on each module, we can describe how they affect the relative displacement and orientation of each, while maintaining the same contact point. The main objective is finding the definition of the contact force that acts in an effort to obtain a physical system of two connected modules that act as one, with double the inputs. We shall model the basic dynamics guiding the physical behavior of the coupled vehicles by an in-depth breakdown of the underlying equations governing the contact force.

Figure 3.2 depicts the system diagram illustrating two interconnected modules. The diagram showcases the contact forces (F_c) acting between the modules, along with the position vectors (p_1 and p_2) representing the positions of each module's center of mass. Additionally, the vectors r_1 and r_2 are illustrated, representing the displacement from the center of mass to the respective contact point.

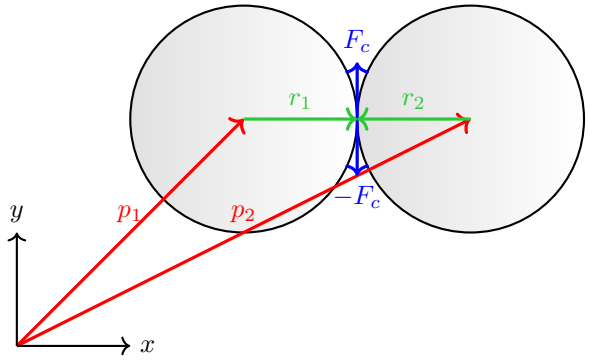

Figure 3.2: System diagram illustrating the contact force (F_c), position vectors (p_1 and p_2) and displacements from body masses to contact point (r_1 and r_2) of the two interconnected modules.

Just like the system outlined in Section 3.2.1, this system is characterized by Newton's second law for both motion and rotation of each module. However, in this case, considering the contact dynamics, we have to factor in the contact force (F_c) and its impact on the system.

Before presenting the new set of differential equations that govern the system's dynamics, it is crucial to clarify that the symbol $\times$ used in any of the equations in this section represents the cross product. Often it is interesting to evaluate the cross product of two vectors assuming that the 2D vectors are extended to 3D by setting their z-coordinate to zero. This is the same as working with 3D vectors on the

xy-plane. Therefore, we will redefine the meaning of this operation as follows:

For any vectors a and b in $\mathbb{R}^2$, where $a = \begin{bmatrix} a_1 & a_2 \end{bmatrix}^T$ and $b = \begin{bmatrix} b_1 & b_2 \end{bmatrix}^T$, we introduce the redefined operation expressed as:

$$a \times b = a_1 b_2 - a_2 b_1 \tag{3.4}$$

With the new $\times$ operation in mind, these new differential equations, similar to (3.2) and (3.3), are as follows:

$$m\ddot{p}_1 = F_1 + F_c \tag{3.5}$$

$$J\ddot{\theta}_1 = M_1 + r_1 \times F_c \tag{3.6}$$

where $F_c \in \mathbb{R}^2$ is the contact force. The dynamics of the second module are obtained in the same manner and defined by:

$$m\ddot{p}_2 = F_2 - F_c \tag{3.7}$$

$$J\ddot{\theta}_2 = M_2 + r_2 \times (-F_c). \tag{3.8}$$

Lastly, let's consider the path of the contact point between the two modules over time, denoted as $p_c(t)$. This path can be obtained by summing the position vector p_1 of the first module's center of mass with the vector r_1 representing the displacement from the first module's center of mass to the contact point. Similarly, we can obtain the path by summing the position vector p_2 of the second module's center of mass with the vector r_2 representing the displacement from the second module's center of mass to the contact point. Equation (3.9) represents the equality of these two paths:

$$p_c(t) = p_1 + r_1 = p_2 + r_2. \tag{3.9}$$

3.2.2.A Definition of the Contact Properties

To ensure continuous contact between the two modules, it is imperative that the contact force F_c satisfies certain constraints throughout their interaction. These constraints stem from the equality in the Equation (3.9) and its corresponding derivatives with respect to time. By examining the equality of this equation and its derivatives, we can deduce the necessary conditions that must be upheld by F_c to maintain the connection between the modules. Therefore, the following set of constraint must be

satisfied in order to define the force of contact (F_c) in time.

$$\frac{d}{dt}(p_1 + r_1 - p_2 - r_2) = 0 \tag{3.10}$$

$$\frac{d^2}{dt^2}(p_1 + r_1 - p_2 - r_2) = 0. \tag{3.11}$$

When observing the differential equations (3.5), (3.6), (3.7), (3.8), and the constraint equations (3.10) and (3.11), it is important to ensure that all the parameters involved are either system inputs, state space variables (or their corresponding derivatives), or can be defined in terms of the aforementioned parameters. With this being said, we must define F_c, r_1 and r_2 as functions of the inputs and state space variables.

The displacement vectors r_1 and r_2 are vectors with magnitude r and angle with respect to the x-axis α_1 and α_2. We know from how the system and the vectors were defined that, $\alpha_1 = \pi + \alpha_2$ and from this knowledge,

$$r_1 = -r_2. \tag{3.12}$$

Having said this, and from (3.9) we can write both vectors as a function of the state space variables p_1 and p_2 as,

$$r_1 = -r_2 = \frac{p_2 - p_1}{2}. \tag{3.13}$$

It will be useful to obtain the parameters $\dot{r_1}$ and $\dot{r_2}$ as a function of the state space variables, and with this in mind we take the previous definition of the displacement vectors,

$$r_i = \|r\| \begin{bmatrix} \cos(\alpha_i) \\ \sin(\alpha_i) \end{bmatrix}$$

and consequently we can obtain $\dot{r_i}$, the derivative in terms of time, as follows:

$$\dot{r_i} = \frac{\partial r_i}{\partial \alpha_i} \frac{d\alpha_i}{dt} = \|r\| \begin{bmatrix} -\sin(\alpha_i) \\ \cos(\alpha_i) \end{bmatrix} \dot{\alpha_i}$$

Taking into consideration that although α_i may not be the same as the state space variable θ_i, because the contact point maintains a constant position relative to the module's fixed body frame, their time derivatives must be the same. Therefore, we have $\dot{\alpha_i} = \dot{\theta_i} = \omega_i$. Considering this, we can derive the

following expression:

$$\begin{aligned}\dot{r_i} &= \|r\| \begin{bmatrix} -\sin(\alpha_i) \\ \cos(\alpha_i) \end{bmatrix} \omega_i \\ &= \begin{bmatrix} 0 & -\omega_i \\ \omega_i & 0 \end{bmatrix} \|r\| \begin{bmatrix} \cos(\alpha_i) \\ \sin(\alpha_i) \end{bmatrix} \\ &= \begin{bmatrix} 0 & -\omega_i \\ \omega_i & 0 \end{bmatrix} r_i \\ &= S(\omega_i) r_i \end{aligned} \tag{3.14}$$

Where the matrix $S(x)$ can be defined as

$$S(x) = \begin{bmatrix} 0 & -x \\ x & 0 \end{bmatrix}, \tag{3.15}$$

and given a vector $k \in \mathbb{R}^2$, the vector $S(1)k$ is a new vector perpendicular to k.

3.2.2.B Definition of the Contact Force

The expression that defines F_c as a function of the state space variables of the system can be obtained from the equations, (3.10) and (3.11). The first equation can be rewritten as,

$$v_1 + \dot{r_1} - v_2 - \dot{r_2} = 0.$$

The parameters $\dot{r_1}$ and $\dot{r_2}$ have been obtained in (3.14) and can be substituted, obtaining the first constraint (3.10) as:

$$v_1 - v_2 + S(\omega_1) r_1 - S(\omega_2) r_2 = 0 \tag{3.16}$$

The second constraint equation (3.11) can be rewritten as the time derivative of the first constraint equation which is given by (3.16), which becomes,

$$\begin{aligned} \dot{v_1} - \dot{v_2} + \frac{d}{dt}\left(S(\omega_1) r_1\right) - \frac{d}{dt}\left(S(\omega_2) r_2\right) = 0 \Leftrightarrow \\ \ddot{p_1} - \ddot{p_2} + S(\dot{\omega}_1) r_1 + S(\omega_1) \dot{r_1} - S(\dot{\omega}_2) r_2 - S(\omega_2) \dot{r_2} = 0 \end{aligned} \tag{3.17}$$

This equation has parameters that are not state space variables, and for that reason in order to obtain the expression of F_c, the following auxiliary substitutions must be done:

- The parameters $\ddot{p_1}$ and $\ddot{p_2}$ are the modules linear acceleration, and can be obtained from (3.5) and

(3.7) that,

$$\ddot{p_1} = \frac{F_1 + F_c}{m} \qquad\qquad \ddot{p_2} = \frac{F_2 - F_c}{m} \tag{3.18}$$

- From the equation (3.14) we can rewrite two of the equation's terms, and obtain:

$$S(\omega_1)\dot{r_1} = S(\omega_1)S(\omega_1)r_1 = -\omega_1^2 r_1 \qquad\qquad S(\omega_2)\dot{r_2} = S(\omega_2)S(\omega_2)r_2 = -\omega_2^2 r_2 \tag{3.19}$$

- Because the parameter $\dot{\omega_i}$ is equal to the second derivative of the angle positioning of the module, $\ddot{\theta}_i$. The equations (3.6) and (3.8) can be used in order to obtain:

$$S(\dot{\omega}_1)r_1 = S\left(\frac{M_1}{J}\right)r_1 + S\left(\frac{r_1 \times F_c}{J}\right)r_1 \qquad S(\dot{\omega}_2)r_2 = S\left(\frac{M_2}{J}\right)r_2 - S\left(\frac{r_2 \times F_c}{J}\right)r_2 \tag{3.20}$$

- Lastly we shall take into consideration the definition of $\times$ and $S(x)$ to reorganize the following,

$$\begin{aligned}
S(a \times b)a &= S(a_1 b_2 - a_2 b_1)a \\
&= \begin{bmatrix} 0 & -a_1 b_2 + a_2 b_1 \\ a_1 b_2 - a_2 b_1 & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \\
&= \begin{bmatrix} -a_1 a_2 b_2 + a_2^2 b_1 \\ a_1^2 b_2 - a_1 a_2 b_1 \end{bmatrix} \\
&= \begin{bmatrix} a_2^2 & -a_1 a_2 \\ -a_1 a_2 & a_1^2 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \\
&= \left(\begin{bmatrix} a_1^2 + a_2^2 & 0 \\ 0 & a_1^2 + a_2^2 \end{bmatrix} - \begin{bmatrix} a_1^2 & a_1 a_2 \\ a_1 a_2 & a_2^2 \end{bmatrix} \right) \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \\
&= \left(\|a\|_2^2 I_2 - a a^T \right) b.
\end{aligned} \tag{3.21}$$

The final expression that defines F_c can be obtained by substituting (3.17) with the equalities derived in (3.18), (3.19), and (3.20), with the assistance of the equality in (3.21):

$$F_c = \left[\Lambda - \frac{r_1 r_1^T}{J} - \frac{r_2 r_2^T}{J} \right]^{-1} \left[\frac{F_2 - F_1}{m} + \frac{S(M_2)r_2 - S(M_1)r_1}{J} + \omega_1^2 r_1 - \omega_2^2 r_2 \right] \tag{3.22}$$

$$\Lambda = \left(\frac{2}{m} + \frac{\|r_1\|^2}{J} + \frac{\|r_2\|^2}{J} \right) I_2$$

where the matrix Λ is a constant-valued diagonal matrix dependent on the system physical parameters.

From the fundamental contact requirement for the positions (3.9), the relationship between the module's positions and the contact point is well established. Taking the first derivative of these equalities, we obtain,

$$v_c = \dot{p_c} = v_1 + S(\omega_1)r_1 = v_2 + S(\omega_2)r_2 \tag{3.23}$$

representing the required relationship between the angular velocities and linear velocities of both modules, and the full connected vehicle. Whenever the modules are connected, these equalities must hold true.

3.2.2.C Systems of Differential Equations

Given the previous deductions and differential equations obtained, the proposed systems of differential equations that govern both modes of operation and define the flow map are formally represented as dependent on the state vector, $x \in \mathbb{R}^{12}$, and the control vector $u \in \mathbb{R}^6$ (for the situation of two modules), as $f^{\circ}(q, x, u)$ for any mode $q \in \{0, 1\}$.

For the sake of simplicity, and for the rest of this thesis, the state vector elements may be referred to as either x_i or their physical meanings, for each $i \in \{1, 2\}$ module, the positions and velocities ($p_i \in \mathbb{R}^2$ and $v_i \in \mathbb{R}^2$) and the angular positions and velocities ($\theta_i \in \mathbb{R}$ and $\omega_i \in \mathbb{R}$) such that,

$$p_1 = [x_1\ x_2]^T,\ v_1 = [x_3\ x_4]^T,\ \theta_1 = x_5,\ \omega_1 = x_6,$$
$$p_2 = [x_7\ x_8]^T,\ v_2 = [x_9\ x_{10}]^T,\ \theta_2 = x_{11},\ \omega_2 = x_{12}.$$

Similarly, the control vector elements may be referred to as either u_i or by their physical meanings. For each $i \in \{1, 2\}$ module, the input forces ($F_i \in \mathbb{R}^2$) and input torques ($M_i \in \mathbb{R}$) are defined such that,

$$F_1 = [u_1\ u_2]^T,\ M_1 = u_3,$$
$$F_2 = [u_4\ u_5]^T,\ M_2 = u_6.$$

Taking this into consideration, from this point forward, this representation will often be used merely in an effort to simplify and enhance the comprehension of many of the expressions shown. Beginning with

the flow map representations for each mode of operation,

$$f^{\circ}(0,x,u)=\begin{cases}\dot{p}_1=v_1\\ \dot{v}_1=\dfrac{F_1}{m}\\ \dot{\theta}_1=\omega_1\\ \dot{\omega}_1=\dfrac{M_1}{J}\\ \dot{p}_2=v_2\\ \dot{v}_2=\dfrac{F_2}{m}\\ \dot{\theta}_2=\omega_2\\ \dot{\omega}_2=\dfrac{M_2}{J}\end{cases}\qquad f^{\circ}(1,x,u)=\begin{cases}\dot{p}_1=v_1\\ \dot{v}_1=\dfrac{F_1+F_c}{m}\\ \dot{\theta}_1=\omega_1\\ \dot{\omega}_1=\dfrac{M_1+r_1\times F_c}{J}\\ \dot{p}_2=v_2\\ \dot{v}_2=\dfrac{F_2-F_c}{m}\\ \dot{\theta}_2=\omega_2\\ \dot{\omega}_2=\dfrac{M_2+r_2\times(-F_c)}{J}\end{cases} \tag{3.24}$$

3.3 Trajectory Generation

Given an isolated module, i, where $(u(t))^{\mathbf{i}}$ is the input vector of the module in time, that guides the module by minimizing the difference between the state vector, $(x(t))^{\mathbf{i}}$, and the reference signal to be followed, $(r(t))^{\mathbf{i}}$. It becomes apparent that this can be scaled to N modules, but in the scope of this thesis, we will delve into defining the OHA for controlling only two modules. With this being said, we consider,

$$x(t)=\begin{bmatrix}(x(t))^{\mathbf{1}}\\ (x(t))^{\mathbf{2}}\end{bmatrix},\quad u(t)=\begin{bmatrix}(u(t))^{\mathbf{1}}\\ (u(t))^{\mathbf{2}}\end{bmatrix},\quad r(t)=\begin{bmatrix}(r(t))^{\mathbf{1}}\\ (r(t))^{\mathbf{2}}\end{bmatrix} \tag{3.25}$$

The notation $[.]^{\mathbf{i}}$ (or $(.)^{\mathbf{i}}$) indicates the values are related to the i-th module.

The concept of trajectory generation relies on the idea of calculating a given continuous function, the desired input, $u^{\star}(\tau)$, that has the main purpose to guide the modules to follow the reference signal, for the following T seconds, according to the current mode they are in, such that $\tau\in[0;T]$. This can be expressed as:

$$u^{\star}(\tau)=\begin{bmatrix}(u^{\star}(\tau))^{\mathbf{1}}\\ (u^{\star}(\tau))^{\mathbf{2}}\end{bmatrix} \tag{3.26}$$

Along with the desired input function, we must also introduce the concept of desired trajectory, $x^{*}(\tau)$. The desired trajectory can be explained as the plant's full state output upon receiving $u^{\star}(\tau)$ as an input.

$$x^{\star}(\tau)=\begin{bmatrix}(x^{\star}(\tau))^{\mathbf{1}}\\ (x^{\star}(\tau))^{\mathbf{2}}\end{bmatrix} \tag{3.27}$$

The functions $x(t)$, $u(t)$ and $r(t)$ are defined in the time variable t giving the state, input and reference values for any instance t, from the beginning to the end of the simulation. On the other hand, the

functions $x^\star(\tau)$ and $u^\star(\tau)$ are functions redefined at any instance of event and for that reason are defined only for the values of τ.

The reference signals have the same structure as the state space vector, i.e.

$$(r(t))^i = \begin{bmatrix} p(t) \\ v(t) \\ \theta(t) \\ \omega(t) \end{bmatrix}^{\mathbf{i}}_{\mathsf{r}}, \tag{3.28}$$

but must not be confused with the concept of the desired trajectory. The desired input, $u^\star(\tau)$, guides the modules in an attempt to follow $r(t)$; the resulting dynamics of inputting $u^\star(\tau)$ yield the desired trajectory $x^\star(\tau)$. The notation $[.]_\mathsf{r}$ (or $(.)_\mathsf{r}$) indicates the variables are related to the reference signal.

As it was mentioned and made clear multiple times now, the modules have two modes of operation. The mode $q = 0$ where the modules are separated and have distinct dynamics, and the mode $q = 1$ where both modules are connected as one single vehicle.

In theory, we defined the reference signals to be distinct for the instances we desire separated modules (mode $q = 0$) and we set the same reference signals ($(r(t))^1 = (r(t))^2$) for the instances we desire the modules to connect (mode $q = 1$).

The diagrams in Figures 3.3 and 3.4 depict the dynamics of each mode in the context of two modules, following their references. The circles describe the position of the modules in time, and the dashed lines represent the position values of the reference signals.

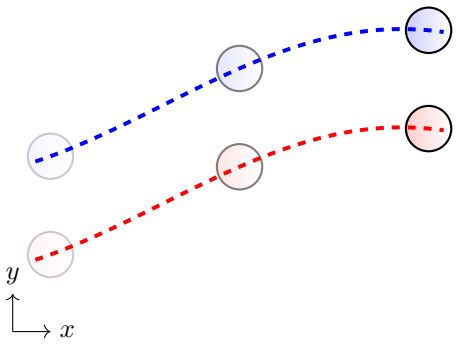

Figure 3.3: Expected dynamics for mode $q = 0$.

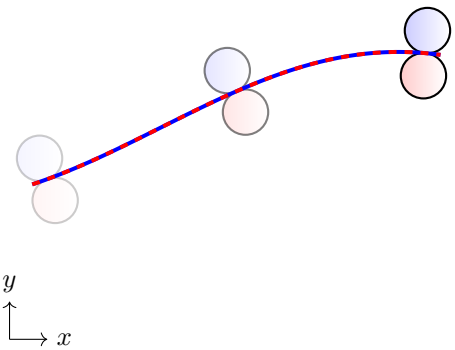

Figure 3.4: Expected dynamics for mode $q = 1$.

We can see the centers of mass of the modules perfectly tracking their respective reference signals for separated modules ($q = 0$). On the other hand, in the case of attached modules ($q = 1$), the two reference signals coincide, and the point accurately following the references is the contact point, i.e., the new center of mass for the modular vehicle. This hints on the idea that for different modes of operation,

the control law calculating $u^\star(\tau)$ will vary.

Let's provide a general explanation of how the pair $(x^\star(\tau), u^\star(\tau))$ is computed in a generic scenario. As mentioned in the preceding sections, the controller design adheres to an ETC scheme, which computes and applies control inputs exclusively when certain events (2.9) occur within the system. With this in mind, the overarching approach from the perspective of "Trajectory Generation" is that, given a specific event instance t_0, the controller unit takes the reference signal $r(t)$ for the proceeding T seconds such that $t \in [t_0; t_0 + T]$. Given the reference signal for the desired interval, and the current OHA state $\xi^+ = G(\xi)$ it computes the pair $(x^\star(\tau), u^\star(\tau))$ for the time interval $\tau \in [0; T]$ that solves the initial value problem,

$$\begin{cases} \dot{x}^\star(\tau) = f^\circ(\hat{q}, x^\star(\tau), u^\star(\tau)) \\ x^\star(0) = \hat{x}. \end{cases} \tag{3.29}$$

With $\hat{x} = x(t_0)$ representing the initial value of $x^\star(\tau)$, used for the calculation of $(x^\star(\tau), u^\star(\tau))$. The initial mode value, $\hat{q}$, defines the current mode of operation and the dynamics of the system to consider ($f^\circ(0, \cdot)$ or $f^\circ(1, \cdot)$). The $u^\star(\tau)$, will determine the input to be applied to the plant for $t \in [t_0; t_0 + T]$, unless a mode switch is triggered (2.6b) or in the event that the deviation between the desired state, $x^\star$, and the actual state, x, exceeds the limit δ (2.11).

As it was previously stated, the solution to the control problem (and consequently to (3.29)), is based on an event-triggered Linear Quadratic Regulator (LQR) based approach. For a comprehensive examination and explanation of the LQR, please consult Appendix A. The two important concepts to take into account when considering the LQR control of the modules are:

- The obtained feedback gain matrix, $K \in \mathbb{R}^{3\times 6}$, was designed to control a single module, defined by its state space variable $(x)^i \in \mathbb{R}^6$, by calculating an input control vector $(u)^i \in \mathbb{R}^3$. For multiple modules, multiple state vectors are calculated, resulting in multiple input vectors. These are concatenated as shown in (3.27) and (3.26).

- The reference signals that the modules are tasked to follow are defined *a priori* and constitute one of the arguments required for the calculation of the control vector. As seen in Figure 3.4, depending on the mode of operation, we may want to track the center of mass of each module to follow their references, or in the case of connected modules, we may wish to have the contact point follow the reference.

Before delving into the definition of the control laws for each mode of operation, it is imperative to thoroughly examine the general LQR control law in order to gain a comprehensive understanding of its underlying principles and ascertain how it can be effectively applied to mode $q = 0$, as well as adapted for $q = 1$.

Let us show the standard LQR control law for calculating the value of the input, u of a given vehicle, given a state vector, x taking into consideration the reference signal r.

$$u = -K\left(x - r\right) = -K\left(\begin{bmatrix} p \\ v \\ \theta \\ \omega \end{bmatrix} - \begin{bmatrix} p \\ v \\ \theta \\ \omega \end{bmatrix}_{\mathsf{r}}\right) = -K\underbrace{\begin{bmatrix} p - (p)_{\mathsf{r}} \\ v - (v)_{\mathsf{r}} \\ \theta - (\theta)_{\mathsf{r}} \\ \omega - (\omega)_{\mathsf{r}} \end{bmatrix}}_{\mathbf{e}} \tag{3.30}$$

The gain matrix is calculated and defined as explained in Appendix A. The expression (3.30) illustrates the relationship between the gain matrix elements and the elements in the error vector $\mathbf{e} \in \mathbb{R}^6$. Put succinctly, K is defined in such a manner that u can be determined if we specify the correct errors in each element of **e**. Ensuring that the elements of **e** satisfy the following conditions allows for correct control law definition.

- The first two elements, $p - (p)_{\mathsf{r}} \in \mathbb{R}^2$, define the positional error between the current position of the vehicle's center of mass and its positional reference;
- The third and fourth elements, $v - (v)_{\mathsf{r}} \in \mathbb{R}^2$, define the velocity error between the current velocity of the vehicle's center of mass and its velocity reference;
- The fifth element, $\theta - (\theta)_{\mathsf{r}} \in \mathbb{R}$, defines the angular positional error between the current angular position of the vehicle and its angular positional reference;
- The sixth element, $\omega - (\omega)_{\mathsf{r}} \in \mathbb{R}$, defines the angular velocity error between the current angular velocity of the vehicle and its angular velocity reference.

In practice, the weight matrices (Q and R) from the cost function (A.2), are defined such that the position (p) and angular position (θ) errors have higher significance when comparing to the ones of the velocity (v) and angular velocity (ω) errors. This is done in order to ensure a more robust control strategy when dealing with position references that are not continuous or have too high of derivatives.

3.3.1 Trajectory for Disconnected Modules

Having a well established definition of the K matrix, composition, and the reference signals, we can assure that the control for any module in mode $q = 0$, can be calculated by using the standard control law (3.30) since the state vector $(x^{\star}(\tau))^{\mathbf{i}}$ depicts the dynamics of the modules in regard to their center of mass and for that reason we should minimize the error between it and its corresponding reference

signal $(r(t_0+\tau))^{\mathbf{i}}$. In this way we obtain for any i-th module,

$$(u^{\star}(\tau))^{\mathbf{i}} = -K\left[(x^{\star}(\tau))^{\mathbf{i}} - (r(t_0+\tau))^{\mathbf{i}}\right], \tag{3.31}$$

and concatenating the desired trajectory and desired input vectors, we obtain the solution to the initial value problem,

$$u^{\star}(\tau) = \begin{bmatrix}(u^{\star}(\tau))^{\mathbf{1}} \\ (u^{\star}(\tau))^{\mathbf{2}}\end{bmatrix} \quad x^{\star}(\tau) = \begin{bmatrix}(x^{\star}(\tau))^{\mathbf{1}} \\ (x^{\star}(\tau))^{\mathbf{2}}\end{bmatrix} \tag{3.32}$$

3.3.2 Trajectory for Connected Modules

The control of the two attached modules becomes more complex and less orthodox in comparison to the usual control of a singular module. The objective behind the attachment of two or more modules is to create a singular, more able vehicle composed by many modules. Considering the expected dynamics depicted by Figure 3.4, some conclusions can be drawn.

The reference signals for each module are equal for the instances we desired connected modules, and because of this we know for a fact that it is impossible for both modules to have their center of mass to follow the reference signal. The same can be said for the velocity, angular position and angular velocity. We solve this by defining carefully the differences composing $\mathbf{e}$ with respect to the full modular vehicle, i.e. the state values to be considered are related to the modular vehicle and not the individual modules. Let us start by defining the error vector in regard to the first module, $(\mathbf{e})^{\mathbf{1}}$.

The position tracking is referenced to the modular vehicle's center of mass, also known as contact point, p_c. This value can be obtained from (3.9) and we know that the positioning error depicts the deviation between p_c and the reference positioning, $(p)^{\mathbf{1}}_r$.

$$\begin{bmatrix}e_1 \\ e_2\end{bmatrix}^{\mathbf{1}} = p_c - (p)^{\mathbf{1}}_r \tag{3.33}$$

Similarly to the positioning error, the velocity error is in regard to the velocity at the contact point, v_c. This variable can be obtained from (3.23) and we know that the velocity error depicts the deviation between v_c and the reference velocity, $(v)^{\mathbf{1}}_r$.

$$\begin{bmatrix}e_3 \\ e_4\end{bmatrix}^{\mathbf{1}} = v_c - (v)^{\mathbf{1}}_r \tag{3.34}$$

When considering the angular positioning error, there are some aspects to be considered. Firstly, we must define the orientation that we desire, for the modules. From Figure 3.5 we establish what we

consider should be the orientation, θ of the modular vehicle.

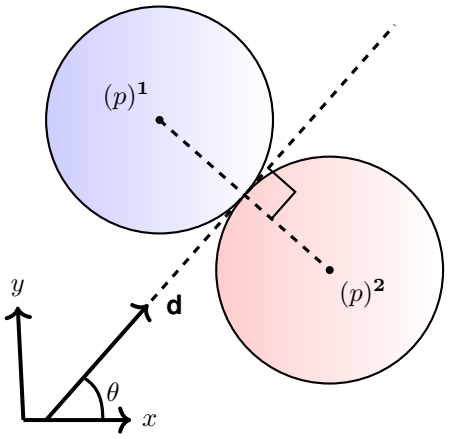

Figure 3.5: Depiction of the modules for mode $q = 1$. The vector $\mathbf{d}$ is perpendicular to $\left((p)^{\mathbf{2}} - (p)^{\mathbf{1}}\right)$ and the angle θ is the angle $\mathbf{d}$ makes with the x-axis. This diagram is not to scale.

Having said this, we can define the modular vehicles' orientation in state space variables, taking the direction vector,

$$\mathbf{d} = S(1)\left((p)^{\mathbf{2}} - (p)^{\mathbf{1}}\right) \tag{3.35}$$

such that $\mathbf{d} \perp \left((p)^{\mathbf{2}} - (p)^{\mathbf{1}}\right)$. Finally, by obtaining the angle between this vector and the x-axis, we find θ.

Let us assume that the reference orientation, $(\theta)^{\mathbf{1}}_r$, is defined to be the angle of the vector tangent to the reference position at any instance. We can assume this because the reference signals are designed to better suit the objective, hence being defined to obtain the dynamics depicted in Figure 3.4. Finalizing the definition of the fifth element of $(\mathbf{e})^{\mathbf{1}}$ as,

$$(e_5)^{\mathbf{1}} = \theta - (\theta)^{\mathbf{1}}_r. \tag{3.36}$$

This difference has to be reframed to the interval $[-\pi, \pi)$, because, given two difference error values (γ) and $(\gamma + 2\pi n)$, with $n \in \mathbb{N}$, both errors should result in the same control values.

The last element of the error vector, e_6, is defined by the difference between the modular vehicle's angular velocity, ω, and the reference angular velocity, $(\omega)^{\mathbf{1}}_r$. Given the attachment configuration of the modules, where both modules are connected to a fixed point, then it is undeniable that both modules and the full vehicle all share the same angular velocity, $(\omega)^{\mathbf{1}} = (\omega)^{\mathbf{2}} = \omega$. With this being said, we obtain,

$$(e_6)^{\mathbf{1}} = \omega - (\omega)^{\mathbf{1}}_r. \tag{3.37}$$

Having defined the error vector, we can rewrite (3.30) for any i module in mode $q = 1$, such that:

$$(u^\star(\tau))^{\mathbf{i}} = -K \left(\begin{bmatrix} p_c(\tau) \\ v_c(\tau) \\ \theta(\tau) \\ \omega(\tau) \end{bmatrix} - \begin{bmatrix} p(t_0+\tau) \\ v(t_0+\tau) \\ \theta(t_0+\tau) \\ \omega(t_0+\tau) \end{bmatrix}_r^{\mathbf{i}} \right) = -K \left((x(\tau))^{\mathbf{v}} - (r(t_0+\tau))^{\mathbf{i}} \right) = -K\mathbf{e} \tag{3.38}$$

where $(x(t))^{\mathbf{v}}$ depicts an adapted state vector regarding the modular vehicle, such that $\mathbf{e}$ is defined as explained. The elements of $(x(t))^{\mathbf{v}}$ are calculated from the state space variables ($x^\star$), in order to solve the initial value problem. Concatenating the desired trajectory and desired input vectors, we obtain the solution to the initial value problem,

$$u^\star(\tau) = \begin{bmatrix} (u^\star(\tau))^{\mathbf{1}} \\ (u^\star(\tau))^{\mathbf{2}} \end{bmatrix} \quad x^\star(\tau) = \begin{bmatrix} (x^\star(\tau))^{\mathbf{1}} \\ (x^\star(\tau))^{\mathbf{2}} \end{bmatrix} \tag{3.39}$$

From the definition of the desired control input calculation, if we ensure f° is continuous between events, we guarantee continuous desired input $u^\star(\tau)$ for any jump, as found necessary from Theorem 1. Assuming that the mechanics of the real system closely resemble the approximation (3.29), the control input obtained will adequately operate the modules to follow the reference signal.

3.4 Closed-Loop Hybrid System

In the Section 2.2, the general definition for an OHA was presented. Now, an overview of the multi-modal aspect of the dynamical system will be provided. The multiple modes define the operation of the system, exhibiting both flowing dynamics and jumping dynamics, defining the state's dynamics by

$$\mathcal{H}_a^\circ := (Q, E, f^\circ, \mathrm{Domain}^\circ, \mathrm{Guard}^\circ, \mathrm{Reset}^\circ, h).$$

Firstly, we acknowledge that there are two modes available and for that reason we define the directed graph $G = (Q, E)$ that defines each operating mode, as follows,

$$\begin{aligned} Q &:= \{0, 1\} \\ E &:= \{(0,1), (1,0)\}. \end{aligned} \tag{3.40}$$

Having all the possible modes of operation defined, we acknowledge that, when operating in mode q, the available modes to switch to are at best the remaining modes, $Q \setminus q$ since there are no edges, E,

directing a node to itself,

$$\forall q \in Q, \quad \{(q,q)\} \notin E, \tag{3.41}$$

and the only possible directions to be followed for this specific case are from 0 to 1 and 1 to 0.

3.4.1 Flow Map

As mentioned, the hybrid system in question can be explained by its continuous flowing dynamics and its discrete jumping dynamics. The continuous mechanics of the system are defined by a function f° and a flow set C°, as seen in (2.5) and (2.6). The flow map, f°, is dependent on the mode of operation and was previously defined in (3.24)

As previously explained, the flow of mode $q = 0$, defined by $f^\circ(0, \cdot)$, is the separated mode and for that reason each input contributes solely for one of the vehicles. In this mode, each module is individually actuated and capable of following their distinct path independent of the other module. On the other hand, the flow of mode $q = 1$, defined by $f^\circ(1, \cdot)$, is the connected mode where the actuation of both modules contributes in some manner to the dynamics of each module.

3.4.2 Flow Set

To complete the definition of the flowing dynamics, we must have a well-defined flow set, C°. This set defines the values that the state x and u can take for both modes of operation. From (2.6) we know that the set is dependent on the definition of $\mathrm{Domain}^\circ(q)$.

Upon analyzing the system, it becomes clear that the working values of (x, u) that define the mode $q = 0$ are for any given point where the modules are found detached from each other. Consequently, for the mode $q = 1$, we consider positions of modules where they are attached to each other. Assuming circular modules with a diameter of R, and assuming that the state position values are referencing the center of the vehicle's body, we can determine that when two modules are in contact with each other, the distance between the modules is R.

$$\mathrm{Domain}^\circ(q) = \begin{cases} \{(x,u) \in \mathbb{R}^6 \times \mathbb{R}^3 : \|p_1 - p_2\|_2 \geq R\} \text{ if } q = 0 \\ \{(x,u) \in \mathbb{R}^6 \times \mathbb{R}^3 : \|p_1 - p_2\|_2 \leq R\} \text{ if } q = 1. \end{cases} .$$

It is important to note that when $q = 1$, the set includes physically impossible states where the distance between modules is less than the diameter. However, despite never actually taking this state values, these sets must be closed sets, in order to keep the hybrid basic conditions. By taking into account the possibility of bad position referencing or module deformation, the presented domains make the system more robust to unforeseen circumstances.

3.4.3 Jump Set

The discrete dynamics rely on a well-defined jump set, D°. As shown in equation (2.6), the set is defined by $\mathrm{Guard}^{\circ}(q, q')$, which depicts the values of (x, u) that, given the current mode of operation q, the system will jump to the mode q'.

Firstly, for the situation defined by the edge $(0, 1)$, we must define the correct set of values that make this transition possible. To do so, we must define a set of conditions such that if it true, the state is in the guard set. The instant, where this jump occurs, is defined by two conditioning factors:

1. The two modules must come into direct contact with each other.

2. We must also ensure that the two connecting modules aren't distancing each other.

The first point can be defined as $(x, u) \in \mathrm{Domain}^{\circ}(1)$. The second condition requires a deeper understanding of what is being asked.

Let us consider the situation where two modules are moving independently. Having p_1 and p_2 determine the positions of each module, and v_1 and v_2 determine the velocities of each module . If we want to find the condition that ensures the modules are not distancing each other, we rely on the idea of non-increasing distance over time. Defining d as the function in time that describes the distance between the two modules, as follows:

$$d = \|p_1 - p_2\|_2.$$

To guarantee that the distance between the two modules doesn't increase over time, we must ensure that its time derivative, $\dot{d}(t)$, is non-positive, so that the modules are not moving away from each other. We can express this condition as follows:

$$\dot{d} \leq 0.$$

The previous condition can be defined in values of the state variables, by using the chain rule [34] to decompose the expression, as

$$\begin{aligned} \frac{p_1 - p_2}{\|p_1 - p_2\|_2} \cdot v_1 - \frac{p_1 - p_2}{\|p_1 - p_2\|_2} \cdot v_2 \leq 0 \Leftrightarrow \\ \frac{1}{\|p_1 - p_2\|_2}(p_1 - p_2) \cdot (v_1 - v_2) \leq 0. \end{aligned}$$

Taking into consideration the operation $\|\cdot\|_2$ is always non-negative, the final condition ensuring that the vehicles are getting closer is:

$$(p_1 - p_2) \cdot (v_1 - v_2) \leq 0 \tag{3.42}$$

This provides the formalization of the set for the edge $(0, 1)$ as,

$$\text{Guard}^\circ(0,1) = \{(x,u) \in \text{Domain}^\circ(1) : (p_1 - p_2) \cdot (v_1 - v_2) \leq 0\}.$$

To finalize the definition of the guard set for the remaining jumping situation, while operating in $q = 1$ and switching to mode $q = 0$, we must reiterate the prior steps. However, this time, we must envision the separation of two modules that are already connected to each other at this point. To gain a deeper understanding of how detachment operates, we refer back to Section 3.2.2.B, where it was established that the contact between modules is maintained by a mechanism exerting a connecting force, F_c. With this in mind, it becomes evident that the nature of the connection will determine the maximum strength of this attachment methodology. Thus, detachment occurs when a contact force with a separation component (the direction of separation of the vehicles, $(p_2 - p_1)/\|p_2 - p_1\|_2$) exceeds the given maximum (positive) attachment force, σ.

$$F_c^T \cdot (p_2 - p_1)/\|p_2 - p_1\|_2 \geq \sigma.$$

The inequality not only ensures that the forces exerted by the vehicles on each other are in the separation direction ($F_c^T \cdot (p_2 - p_1)/\|p_2 - p_1\|_2 > 0$) but also of sufficient magnitude to provoke detachment.

$$\text{Guard}^\circ(1,0) = \{(x,u) \in \text{Domain}^\circ(1) : F_c^T \cdot (p_2 - p_1)/\|p_2 - p_1\|_2 \geq \sigma\}.$$

The final complete definition of the jump set is, as follows,

$$\text{Guard}^\circ(q,q') = \begin{cases} \{(x,u) \in \text{Domain}^\circ(1) : (p_1 - p_2) \cdot (v_1 - v_2) \leq 0\} & \text{if } (q,q') = (0,1) \\ \{(x,u) \in \text{Domain}^\circ(1) : F_c^T \cdot (p_2 - p_1)/\|p_2 - p_1\|_2 \geq \sigma\} & \text{if } (q,q') = (1,0) \end{cases}.$$

3.4.4 Jump Map

Now that we have defined the Guard° set and the jump set, D° (from Equation (2.6)), we need to explain how the state reacts to each different type of jump. For instance, in the first scenario ($(q,q') = (0,1)$) where two modules collide with each other, we must take into account a discontinuous dynamics. Assuming two vehicles with independent dynamics and no net external forces, when a collision occurs between these two modules, we get from Newton's third law:

$$F_{12} = -F_{21}$$

where F_{12} is the force that the first module exerts on the second module, and F_{21} is the force that the second module exerts on the first module. According to Newton's second law:

$$m_1 a_1 = -m_2 a_2.$$

Where a_i is the acceleration of the i-th module. Since the acceleration of any module can be described as the time derivative of its velocity, we can rewrite the equation as:

$$\begin{aligned} m_1 \frac{d}{dt} v_1 &= - m_2 \frac{d}{dt} v_2 \\ \frac{d}{dt}(m_1 v_1) &= - \frac{d}{dt}(m_2 v_2) \\ \frac{d}{dt}(P_1) &= - \frac{d}{dt}(P_2) \\ \frac{d}{dt}(P_1 + P_2) &= 0 \end{aligned}$$

which gives us the conservation of linear momentum:

$$P_1 + P_2 = P^+. \tag{3.43}$$

Where P_1 and P_2 are the linear momenta of each module before the collision and P^+ is the linear momentum immediately after, of the full system. To obtain the state space after the jump, x^+, as a function of the state before the jump, x, we rewrite (3.43) to obtain,

$$\begin{aligned} P_1 + P_2 &= P^+ \Leftrightarrow \\ m_1 v_1 + m_2 v_2 &= (m_1 + m_2) v^+ \end{aligned}$$

After the collision, the two modules are connected and thus, the center of mass of the connected modules, p_c, must have v^+ velocity.

$$\begin{aligned} m_1 v_1 + m_2 v_2 &= (m_1 + m_2) v^+ \\ v^+ &= \frac{m_1 v_1 + m_2 v_2}{m_1 + m_2} \end{aligned}$$

Finally, for the sake of simplicity, assuming that both modules have the same mass m, the generalized equation that expresses v^+ as a function of v is:

$$\begin{aligned} v^+ &= \frac{m v_1 + m v_2}{2m} \\ v^+ &= \frac{v_1 + v_2}{2} \end{aligned} \tag{3.44}$$

The equation (3.44) implies that the resulting speed of the connected modules' center of mass after a collision is the average speed between the two modules before connecting, with their mass as

weight.

Similarly to the conservation of linear momentum, the conservation of angular momentum must also be ensured, which will result in the post-jump angular velocity values. The angular momentum of a body is given by,

$$L = J \cdot \omega$$

where $J \in \mathbb{R}$ depicts the moment of inertia, and ω the angular velocity. Let us define

$$J = \frac{1}{2} m r^2$$

the moment of inertia [35] of a single detached circular module of mass m and radius r. The system of two detached modules before connecting has angular momentum given by,

$$\begin{aligned} L_1 + L_2 &= J \cdot \omega_1 + J \cdot \omega_2 \\ &= J(\omega_1 + \omega_2). \end{aligned}$$

After the two modules connect, i.e. post jump, the system's overall angular momentum is represented by,

$$L^+ = J^+ \cdot \omega^+$$

Analogously to the linear momentum, after the collision, the full system's speed (in this case angular), is given by ω^+. The main difference, to take into consideration, is that, contrary to the linear momentum where the connected devices have double the mass of a single vehicle, when considering the angular dynamics, the system's moment of inertia when connected, J^+, is not double the original J.

When connected, each module has a new moment of inertia, due to the new axis of rotation at the connection point. From [35], this new moment of inertia J^* can be calculated as,

$$J^* = J + md^2 = \frac{3}{2} m r^2.$$

with d being the distance from the module's center of mass to the new rotational axis, which means that $d = r$. Because there are two identical modules, with the same J^*, the total angular mass of the post-jump connected vehicle,

$$J^+ = 2 \cdot J^* = 3mr^2 = 6J$$

Finally, the relationship between the angular velocities of the modules before connecting (ω_1 and ω_2)

and the angular velocity of the modular vehicle after connection (ω^+) can be expressed as:

$$\begin{aligned} J^+\omega^+ &= J \cdot (\omega_1 + \omega_2) \\ \omega^+ &= \frac{J}{J^+}(\omega_1 + \omega_2) \\ \omega^+ &= \frac{\omega_1 + \omega_2}{6}. \end{aligned} \tag{3.45}$$

Both the resulting linear velocity v^+ and angular velocity ω^+ are related to the full connected system as a single vehicle. This means that both velocities are with respect to the center of mass of the connected modules, p_c. The angular velocity of the full connected vehicle at any given moment, is the same as the angular velocity of each module, since they are connected. On the other hand, the linear velocity of the full connected vehicle and each module is not the same. The calculated v^+ expresses the speed at the connecting point, v_c. Because there might exist a non-null rotation speed, each module has a linear speed different from the connecting point's v_c. The singular module's linear velocities, v_1 and v_2, depend on the full system's linear velocity, v^+, the angular velocity of the system, ω^+, and the distances between the connection point, p_c, and each module's center of mass, p_1 and p_2, as explained in, (3.23).

The last state values to be reassigned for this type of jump is for the linear velocities of each module, v_1^+ and v_2^+. Since the conservation of linear momentum, defined the linear speed at the contact point, v^+ and the conservation of angular momentum defined the angular velocities, ω_1 and ω_2, which are the same, for each module and for the full connected vehicle as ω^+. The linear velocities of each module are derived from (3.23) obtaining the expressions,

$$\begin{aligned} v_1^+ &= v^+ + S(\omega^+)r_1 \\ v_2^+ &= v^+ + S(\omega^+)r_2. \end{aligned} \tag{3.46}$$

From the described ETC dynamics in Section 2.3, it is clear that a mode change necessitates the recalculation of the desired state ($x^\star$) and input ($u^\star$) vectors. Moreover, as per (3.29), at any given instant $t \in [t_j + t_j + T]$, the control input is given by, $u(t) = u^\star(\tau)$. Where t_j is the instance of the j-th event and $u^\star(\tau)$ is the desired input calculated for $\tau \in [0; T]$.

Taking all of this into consideration, along with the expressions in (3.45), and (3.46), the reset map

for edge $(0,1)$ is defined as,

$$\mathrm{Reset}^{\circ}(0,1,x,u) = \left(\begin{bmatrix} p_1 \\ v_1^+ \\ \theta_1 \\ \omega^+ \\ p_2 \\ v_2^+ \\ \theta_2 \\ \omega^+ \end{bmatrix}, u^{\star}(0) \right) \text{ if } (q,q',x,u) \in \{0\} \times \{1\} \times \mathrm{Guard}^{\circ}(0,1). \tag{3.47}$$

Lastly, to define the second type of mode switch, which occurs when $(q,q') = (1,0)$, the Reset indicates that there are no discrete dynamics with respect to the state x. Only the new control input values ($u^{\star}(\tau)$) are reassigned. Thus, $\mathrm{Reset}(1,0,\cdot)$ can be summarized as:

$$\mathrm{Reset}^{\circ}(1,0,x,u) = (x, u^{\star}(0)) \text{ if } (q,q',x,u) \in \{1\} \times \{0\} \times \mathrm{Guard}^{\circ}(1,0). \tag{3.48}$$

Finally, the OHA is established and defined in accordance with the system at hand. Coupling this $\mathcal{H}_a^{\circ}$ along with the model for the event-triggering hybrid system from section 2.3, and the control law shown in Section 3.3. The closed-loop hybrid system is finished and capable of following trajectories. The defined automaton is able to transition between modes of operation, controlling the modules using the established ETC mechanism.

4

Simulation Results

This chapter presents the results obtained from simulations employing Event-Triggered Control on the Open Hybrid Automaton, utilizing the Hybrid Equation Toolbox in *MATLAB*. These simulations provide a detailed insight into the system's behavior under various conditions, forming the foundation for subsequent discussions and conclusions in this study.

The following graphs fall into two categories: the 2-dimensional position graphs and the time domain representation of variables. The first type of graph provides a visual representation of the 2-dimensional positions of two modules over time. The first module's position is represented by a dashed blue line, while the second module's position is depicted with a dashed red line. The desired reference positions for the modules are shown as thicker continuous lines in lighter colors. Additionally, the graphs include representations of the circular modules at each instant where the trajectories are calculated (the start of the simulation, and all ETC events). This visualization offers insights into both position and time domain. It is important to note that the direction of movement in the examples is, generally, from left to right. In the case of the time domain graphs, these are self-explanatory, and the vertical dashed lines coincide with the instances of any given ETC event.

The various tests and simulations shown were conducted over a total simulation period of 6 seconds, with a maximum interval (T) of 1 second between events. Given this, it is anticipated that in cases where there is no change in modes and no disturbances to the system, the entire simulation involves six instances of trajectory calculation: the initial calculation at the commencement of the simulation, followed by five subsequent ETC events occurring at one-second intervals. It must also be noted that for every 2-dimensional position graph in this chapter, there is an equivalent graph (in Appendix B) representing the desired positions instead of the reference positions.

The Figure 4.1 illustrates the dynamics of two initially separated modules ($q = 0$), commencing from positions uncorrelated with their references.

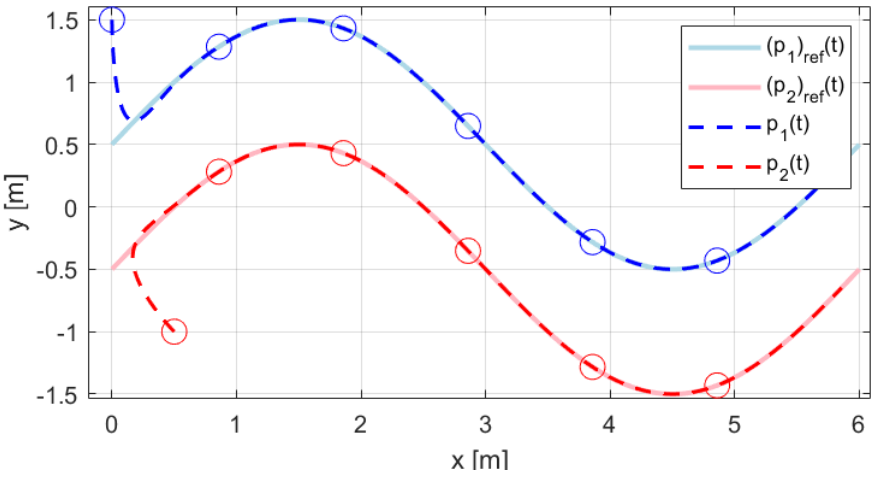

Figure 4.1: Plot of the positions ($p_1(t)$ and $p_2(t)$) and corresponding references ($(p_1)_{ref}(t)$ and $(p_2)_{ref}(t)$), for the case of constant mode, q=0. See Figure B.1 for the corresponding desired positions ($p_1^\star(t)$ and $p_2^\star(t)$).

These modules promptly converge towards their respective designated references, demonstrating the system's capability to control each individual vehicle.

The control of interconnected modules constitutes one of the primary objectives of this work. Figure 4.2 portrays the dynamics of two modules, initially linked ($q = 1$) and tracking a single common reference.

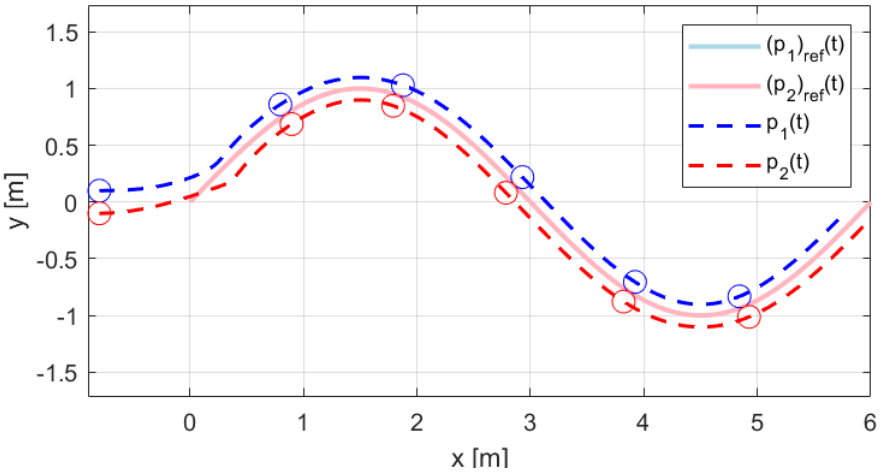

Figure 4.2: Plot of the positions ($p_1(t)$ and $p_2(t)$) and corresponding references ($(p_1)_{ref}(t)$ and $(p_2)_{ref}(t)$), for the case of constant mode, q=1. See Figure B.2 for the corresponding desired positions ($p_1^\star(t)$ and $p_2^\star(t)$).

During the moments when the vehicles are attached to each other, they share the same reference (hence the overlapping reference signals). On these occasions, the contact point between the modules should track the reference for proper control of the system. The displayed Figure provides evidence of accurate control of modules for the mode $q = 1$.

Controlling each operational module of the system is necessary but not sufficient for achieving the desired outcome of a system capable of following any trajectory. The following figures were generated based on tests aimed at transitioning between modes of operation.

In Figure 4.3, the system begins with two separated modules ($q = 0$), and at a certain point, the reference signals overlap, resulting in the attachment of the two modules ($q = 1$).

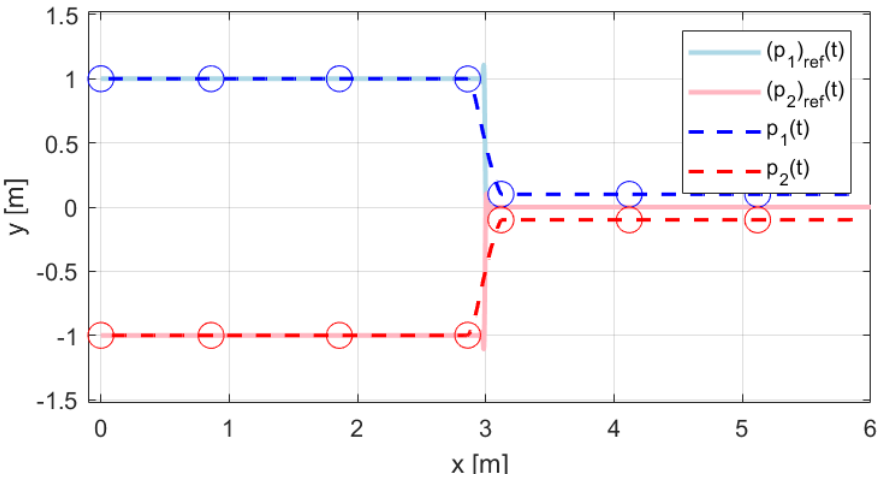

Figure 4.3: Plot of the positions ($p_1(t)$ and $p_2(t)$) and corresponding references ($(p_1)_{ref}(t)$ and $(p_2)_{ref}(t)$), for the case of two modules joining, starting separated. See Figure B.3 for the corresponding desired positions ($p_1^\star(t)$ and $p_2^\star(t)$).

The primary observations drawn from the obtained results indicate that the simulation effectively illustrated the transition between modes of operation. Specifically, the moment of contact is represented by the leftmost connected pair of circles in the figure, signifying that the contact moment leads to the reevaluation of the desired trajectories.

Figure 4.4 portrays the same simulation as depicted in Figure 4.3, but it displays the current mode of operation of the vehicles throughout the entire simulation duration.

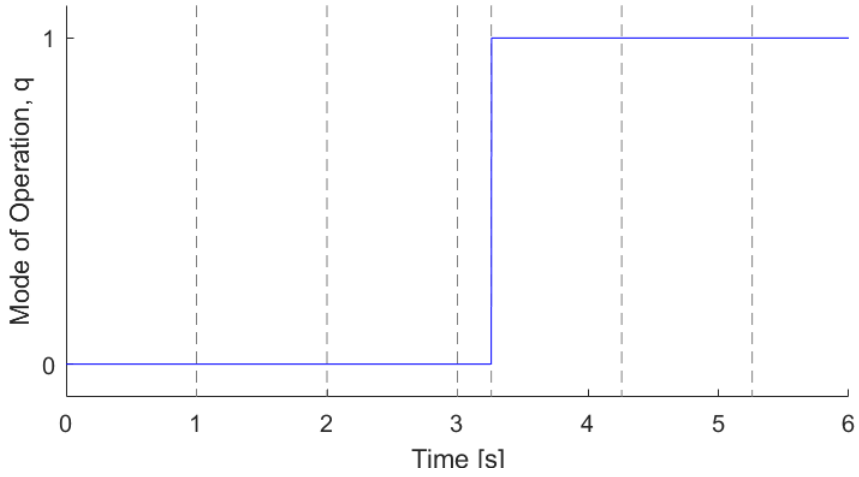

Figure 4.4: Plot of the mode of operation of the modules, for the case of figure 4.3.

The pair of Figures 4.5 and 4.6, much like the preceding two (Figures 4.3 and 4.4), illustrate the positions and the modes of operation for a mode transition. However, in this case, the starting point is two connected modules ($q = 1$) with the intention of separation ($q = 0$).

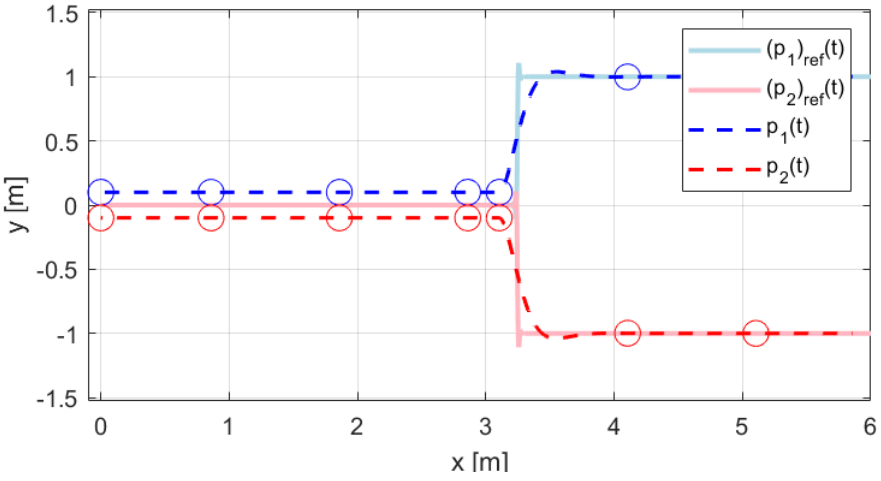

Figure 4.5: Plot of the positions ($p_1(t)$ and $p_2(t)$) and corresponding references ($(p_1)_{ref}(t)$ and $(p_2)_{ref}(t)$), for the case of two modules separating, starting together. See Figure B.4 for the corresponding desired positions ($p_1^\star(t)$ and $p_2^\star(t)$).

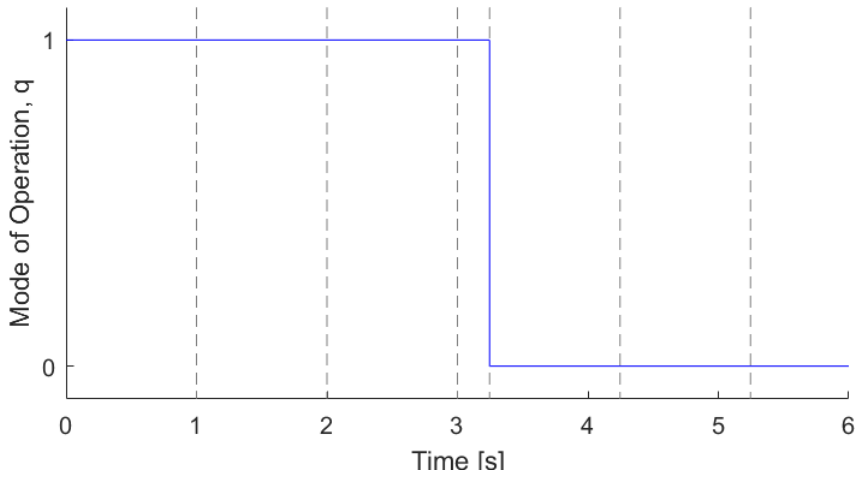

Figure 4.6: Plot of the mode of operation of the modules, for the case of Figure 4.5.

Similar to the previous simulation involving module attachment, the mode transition was effectively demonstrated and simulated. The moment of module separation (mode change) is represented in Figure 4.5 by a pair of modules, and in Figure 4.6 by the fourth vertical dashed line.

After testing the system for operation in both modes and ensuring the correct transitions between them, the next step is to demonstrate the full capabilities of the control system by following more complex, multimode reference signals. Figures 4.7 and 4.8 showcase the dynamics and modes of operation (respectively) of the two modules for a more complete reference signal.

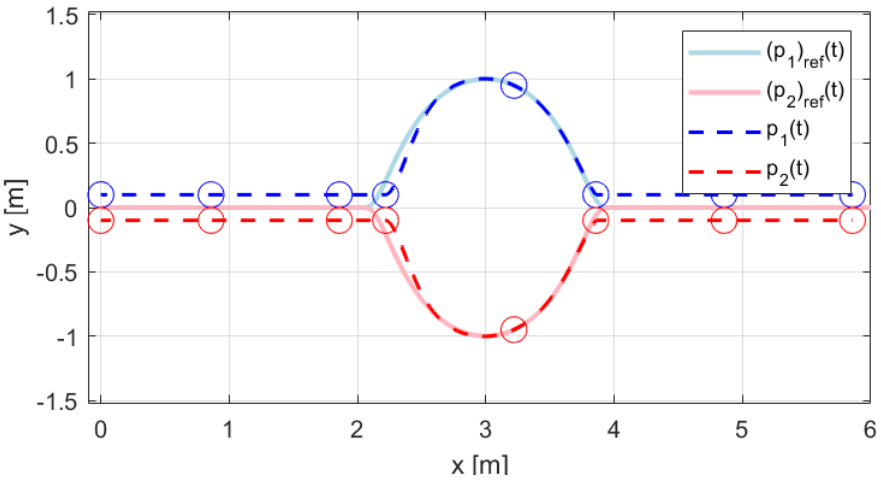

Figure 4.7: Plot of the positions ($p_1(t)$ and $p_2(t)$) and corresponding references ($(p_1)_{ref}(t)$ and $(p_2)_{ref}(t)$), for the case of two modules separating and joining. See Figure B.5 for the corresponding desired positions ($p_1^\star(t)$ and $p_2^\star(t)$).

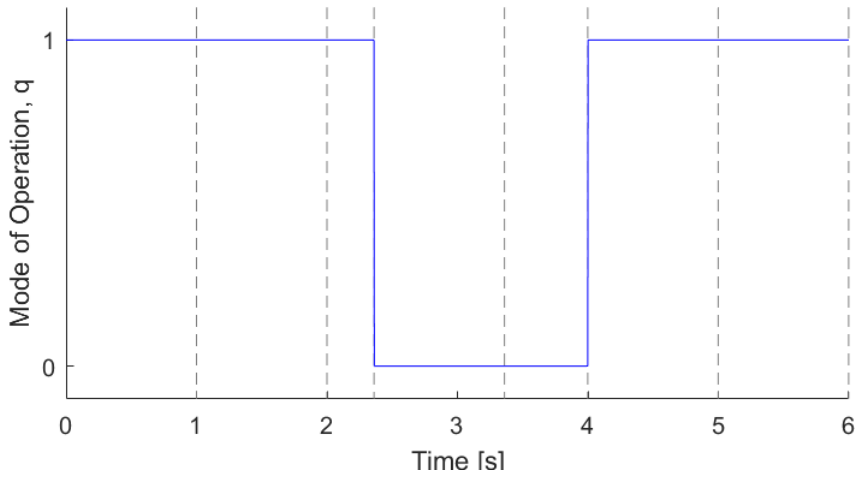

Figure 4.8: Plot of the mode of operation of the modules, for the case of Figure 4.7.

Figure 4.9 illustrates the O_y direction component of the forces actuating on both vehicles for the same simulation (Figure 4.7). Initially, the vehicles maintain a constant y position, and non-zero actuation only occurs when separation is required.

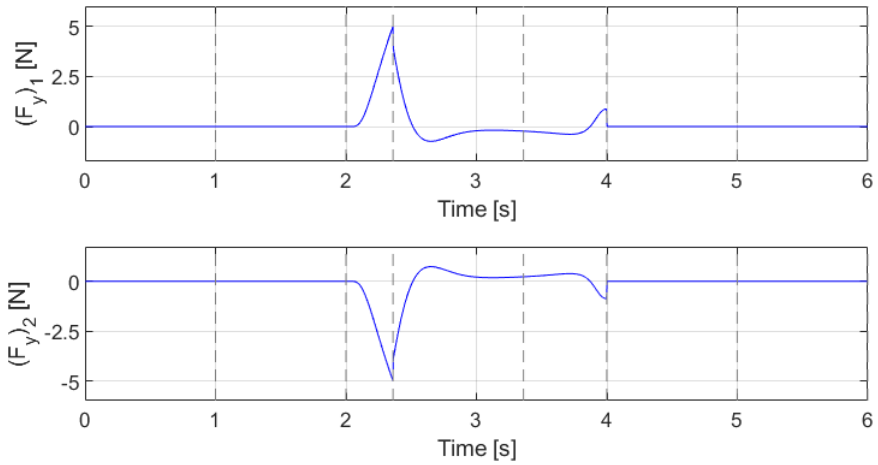

Figure 4.9: Plot of the force actuation in the separation direction O_y in both vehicles, for the case of Figure 4.7.

The forces were selected, specifically the component in O_y, as this aligns with the direction of module separation. It is noticeable that the force of separation increases in both modules, and when these forces are deemed strong enough to break the attachment method [28], the vehicles separate and the mode changes, as indicated by the third vertical dashed line.

To conclude the presentation of the various reference signals tested to ensure the robustness of the control strategy and system definition, Figures 4.10 and 4.11 demonstrate the system's ability to switch between the required modes, enabling it to achieve any desired trajectory.

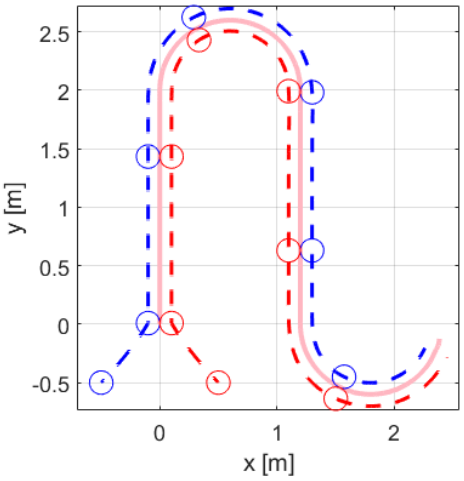

Figure 4.10: "Lawnmower" reference.
See Figure B.6.

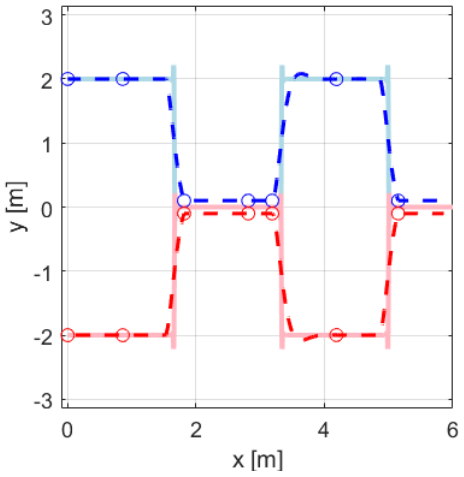

Figure 4.11: Square reference.
See Figure B.7.

The reference signal, depicted in the left plot, was dubbed "lawnmower" due to its distinctive shape. Its purpose is to demonstrate the system's capability to track more complex trajectories with varying rotations and positions, even when the modules are attached. This ensures precise control of the connected vehicles. The second figure features two symmetric square reference signals, further affirming the modules' robust response in swiftly transitioning between modes of operation without difficulty in adhering to the given task.

Despite the promising results observed in the previous simulations under these "ideal" conditions, these initial findings do not adequately attest to the robustness of the system's control or the resilience of the plant in the face of state disturbances. In reality, one of the pivotal aspects of the system remains unconfirmed: the system's capacity to recalibrate its control input ($u^\star$) in scenarios where the system state deviates from the intended target ($x^\star$).

The simulations previously presented showed accurate tracking, since the forces and torques applied to the modules were precisely as designed. However, in reality, there exist external forces that are unaccounted for and unpredictable. Moreover, the communication between the controller unit, responsible for calculating the desired input values ($u^\star$), and the actuator may face compromises. Due to this, the inputs may not be adequate to achieve the intended outcome. To illustrate this, let us consider an example where the desired input value is calculated as usual, but the actual actuation is compromised,

resulting in,

$$F_1(t) = F_1^\star(\tau) - \begin{bmatrix} \mu \\ \mu \end{bmatrix} sin(10t)$$
$$F_2(t) = F_2^\star(\tau) - \begin{bmatrix} \mu \\ \mu \end{bmatrix} sin(10t)$$

Where F_1 and $F_1^\star$ (as well as F_2 and $F_2^\star$) represent the first and second (fourth and fifth) elements of the control vectors $u(t)$ and $u^*(t)$. The parameter μ quantifies the magnitude of the disturbance being considered.

The ensuing three simulations were conducted employing simple sinusoidal references, and all parameters were held constant, except for the disturbance coefficient μ. In these experiments, the ETC error threshold (2.11) was maintained at a constant value of $\delta = 0.2$. The uniform simulation parameters across the three scenarios (except for μ) were chosen to showcase the specific impacts of varying disturbance intensities.

The first scenario, depicted by Figures 4.12 and 4.13, serves as a baseline simulation, as it entails no disturbance in effect ($\mu = 0$).

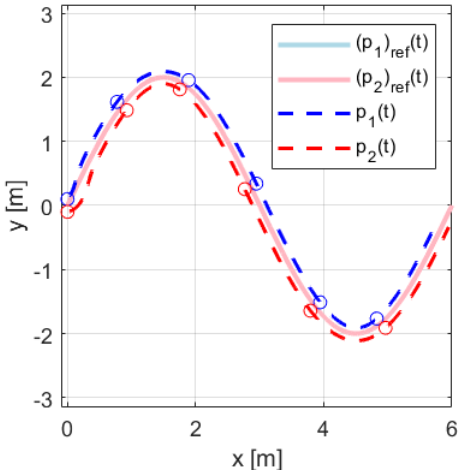

Figure 4.12: Dynamics for $\delta = 0.2$ and $\mu = 0$.

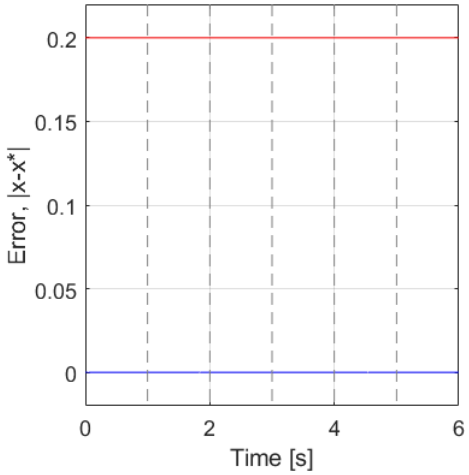

Figure 4.13: Error for $\delta = 0.2$ and $\mu = 0$.

Just as expected, the trajectories are followed perfectly and the error plot displays a consistently null value, along with equidistant ETC instances.

The second situation, depicted in Figures 4.14 and 4.15, illustrates the outcome of the control system in the presence of a minor disturbance coefficient ($\mu = 0.05$).

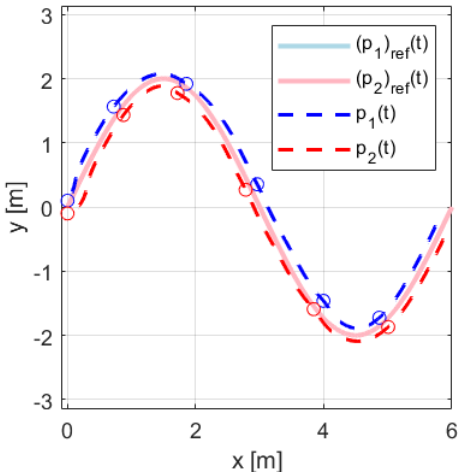

Figure 4.14: Dynamics for $\delta = 0.2$ and $\mu = 0.05$.

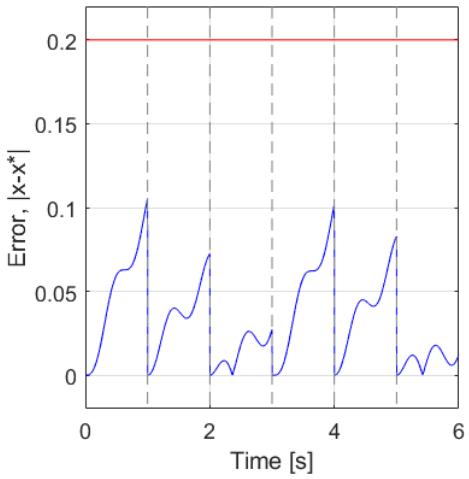

Figure 4.15: Error for $\delta = 0.2$ and $\mu = 0.05$.

The small dimension of μ is insufficient to yield significant deviations from the desired trajectory. While there are some minor visible deviations, it is evident from the error plot that the deviation is not significant enough to surpass the threshold values. Furthermore, the event instances remain equidistant, separated by one second.

The final situation depicted in Figures 4.16 and 4.17 now exhibits a substantial disturbance, providing a compelling demonstration of the system's stability.

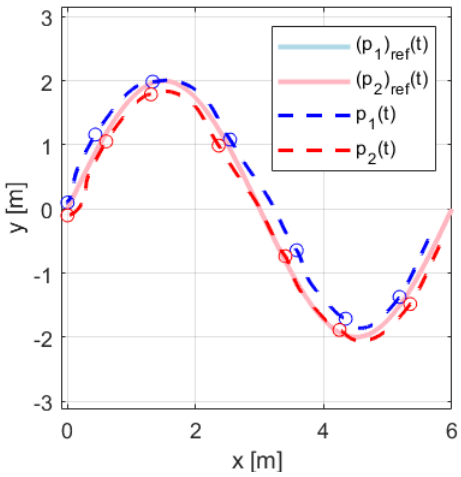

Figure 4.16: Dynamics for $\delta = 0.2$ and $\mu = 0.15$.

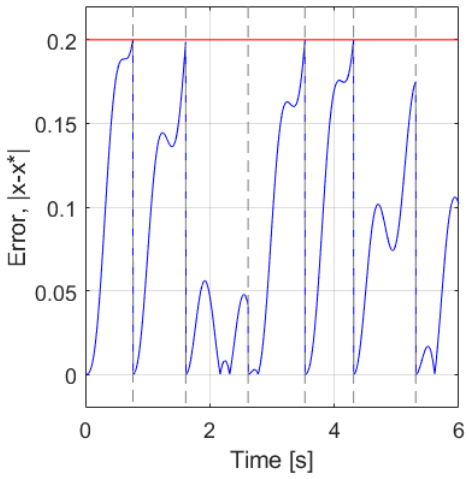

Figure 4.17: Error for $\delta = 0.2$ and $\mu = 0.15$.

From the error plot, it becomes apparent that the disturbance has become significant to the point where the maximum threshold (δ) of the error has been surpassed. For this third simulation, there can be seen six vertical dashed lines, indicating six events of recalculation of the control input, as opposed to the five in the previous two examples. This demonstrates that the disturbance was substantial enough to necessitate recalculation, although from the positioning plot, the error is not as apparent. It is visibly noticeable that the tracking of the reference signal isn't accurate, but it is this ETC mechanism that maintains the system in a controlled state, and limits divergence from the reference.

The sensitivity of the system to the deviations can be managed by adjusting the values of the threshold (δ), resulting in a more accurately referenced path, though at the cost of multiple unexpected recalculations of the control input. Figures 4.18 and 4.19 depict the position and error plots, for the situation where $\delta = 0.1$ and $\mu = 0.15$.

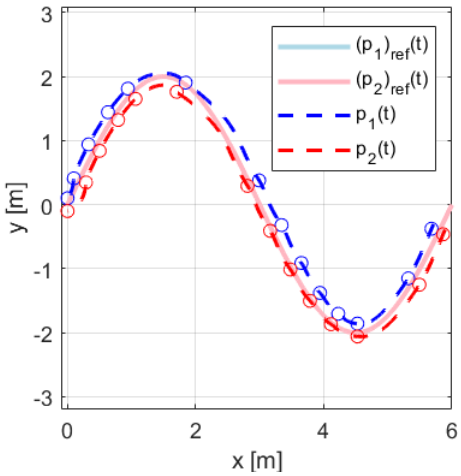

Figure 4.18: Dynamics for $\delta = 0,1$ and $\mu = 0.15$.

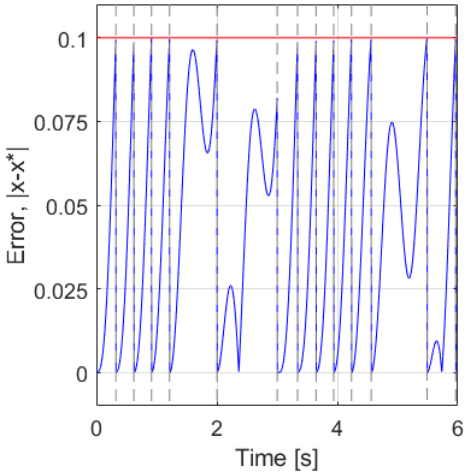

Figure 4.19: Error for $\delta = 0,1$ and $\mu = 0.15$.

Even though, in this situation, the disturbance (μ) was kept the same as in the previous simulation (Figures 4.16 and 4.17), due to the decreased maximum threshold, the event-triggered controller becomes more sensitive to smaller deviations. This leads to more frequent recalculations of $(x^\star, u^\star)$, resulting in better tracking of the reference signal.

In Figures 4.20 and 4.21, the opposite situation was depicted. This time we kept the same disturbance $\mu = 0.15$ but changed the threshold to be larger ($\delta = 0.3$), allowing for greater divergence from the objective.

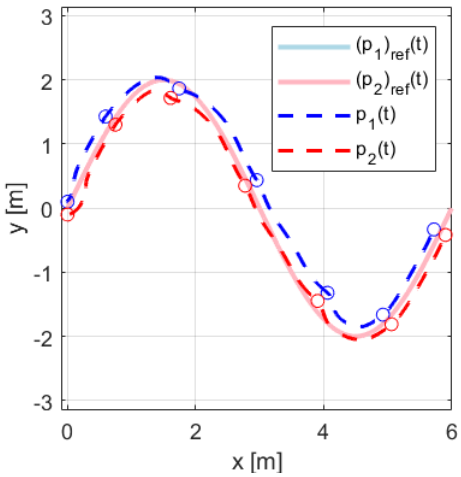

Figure 4.20: Dynamics for $\delta = 0,3$ and $\mu = 0.15$.

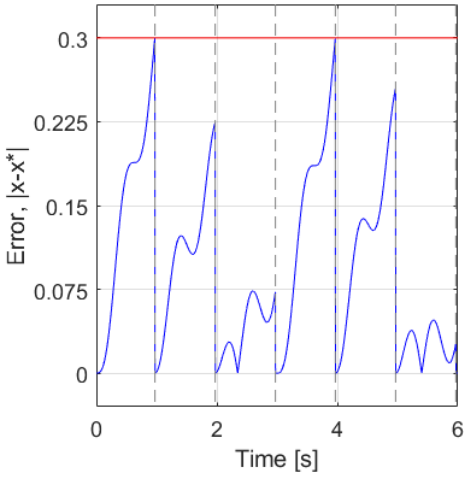

Figure 4.21: Error for $\delta = 0,3$ and $\mu = 0.15$.

The results from this simulation demonstrate less accurate tracking of the reference signals when compared to Figures 4.16 and 4.17. In this case, a divergence error greater than 0.2 between the state value (x) and the desired value ($x^{\star}$) is permitted without triggering a recalculation.

5

Conclusion

In this thesis, we introduced the concepts of event-triggered control of multiple modules of aerial vehicles along with the paradigm of hybrid systems and how it can be applied to the system under study.

The work was divided into an initial research on the state of the art and the current applications of the topic at hand. A formal proposition of the problem, defined as an Open Hybrid Automaton, and the respective theoretical proof of stability, motivated the final definition of the plant system and the corresponding event-triggered control strategy.

The state-space equations describing the dynamics of the plant, both for two separated modules, as well as for the second mode where the two modules are connected and operate as a single vehicle, were derived. The final step involved the physical evaluation of how the properties of the vehicles would respond in real-world conditions when attaching and detaching from each other. Following a thorough analysis of the module's physical properties, a definitive proposal for the complete hybrid system was formulated, encompassing the multi-mode dynamics in conjunction with the event-triggered control paradigm.

Finishing it all off by conducting simulations using the software MATLAB, which illustrated the correct functioning of the closed-loop system and demonstrated the simulated dynamics of the modules following more complex reference trajectories. Having demonstrated that even under adverse situations and disturbances, the system maintained its course and adhered to the main objective.

As part of future research, intriguing open questions arise regarding the influence of scaling the system to N modules on its dynamics and outcomes. Regrettably, due to the time constraints in completing this thesis, delving into this advanced perspective was not feasible. Nonetheless, it sparks curiosity about the numerous applications that could be derived from these multiple N modules, capable of achieving far more than a single UAV. Having said this, the contribution of the work in this thesis, particularly the Event-Triggered Control paradigm, proves more beneficial in alleviating computational and communication strain from a real-time control strategy, especially when dealing with N modules and more complex systems. This is achieved by confining the computational and communicative load to event instances.

Bibliography

[1] D. Saldana, B. Gabrich, G. Li, M. Yim, and V. Kumar, "Modquad: The flying modular structure that self-assembles in midair," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 691–698.

[2] E. Barmpounakis and N. Geroliminis, "On the new era of urban traffic monitoring with massive drone data: The pneuma large-scale field experiment," *Transportation research part C: emerging technologies*, vol. 111, pp. 50–71, 2020.

[3] CONSORTIQ, https://consortiq.com/, accessed: 01-11-2022.

[4] M. Abdelrahim, R. Postoyan, J. Daafouz, and D. Nesic, "Co-design of output feedback laws and event-triggering conditions for linear systems," *Proceedings of the IEEE Conference on Decision and Control*, vol. 2015, 08 2014.

[5] D. Saldana, P. M. Gupta, and V. Kumar, "Design and control of aerial modules for inflight self-disassembly," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3410–3417, 2019.

[6] J. Paulos, B. Caraher, and M. Yim, "Emulating a fully actuated aerial vehicle using two actuators," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 7011–7016.

[7] G. E. Setyawan, W. Kurniawan, and A. C. L. Gaol, "Linear quadratic regulator controller (lqr) for ar. drone's safe landing," *2019 International Conference on Sustainable Information Engineering and Technology (SIET)*, pp. 228–233, 2019.

[8] N. S. Labib, M. R. Brust, G. Danoy, and P. Bouvry, "The rise of drones in internet of things: A survey on the evolution, prospects and challenges of unmanned aerial vehicles," *IEEE Access*, vol. 9, pp. 115 466–115 487, 2021.

[9] W. P. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *2012 ieee 51st ieee conference on decision and control (cdc)*. IEEE, 2012, pp. 3270–3285.

[10] M. Okasha, J. Kralev, and M. Islam, "Design and experimental comparison of pid, lqr and mpc stabilizing controllers for parrot mambo mini-drone," *Aerospace*, vol. 9, no. 6, p. 298, 2022.

[11] R. Goedel, R. G. Sanfelice, and A. R. Teel, *Hybrid dynamical systems: modeling stability, and robustness*. Princeton University Press, Princeton, NJ, 2012.

[12] D. Saldana, B. Gabrich, G. Li, M. Yim, and V. Kumar, "Modquad: The flying modular structure that self-assembles in midair," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 691–698.

[13] G. Del Serrone, G. Cantisani, and P. Peluso, "Speed data collection methods: a review," *Transportation research procedia*, vol. 69, pp. 512–519, 2023.

[14] E. Balestrieri, P. Daponte, L. De Vito, and F. Lamonaca, "Sensors and measurements for unmanned systems: An overview," *Sensors*, vol. 21, no. 4, p. 1518, 2021.

[15] FRONTEX, https://frontex.europa.eu/, accessed: 19-09-2023.

[16] ——, https://tinyurl.com/frontex-thematic-industry-days, accessed: 19-09-2023.

[17] C. Mattupalli, C. A. Moffet, K. N. Shah, and C. A. Young, "Supervised classification of rgb aerial imagery to evaluate the impact of a root rot disease," *Remote sensing*, vol. 10, no. 6, p. 917, 2018.

[18] E. Vrochidou, V. N. Tsakalidou, I. Kalathas, T. Gkrimpizis, T. Pachidis, and V. G. Kaburlasos, "An overview of end effectors in agricultural robotic harvesting systems," *Agriculture*, vol. 12, no. 8, p. 1240, 2022.

[19] W. Tong, W. Jie, and T. Bailing, "Periodic event-triggered formation control for multi-uav systems with collision avoidance," *Chinese Journal of Aeronautics*, vol. 35, no. 8, pp. 193–203, 2022.

[20] M. Araki, "Pid control," *Control Systems, Robotics and Automation: System Analysis and Control: Classical Approaches II*, pp. 58–79, 2009.

[21] J. E. Dentler, "Real-time model predictive control for aerial manipulation," Ph.D. dissertation, University of Luxembourg, Luxembourg, Luxembourg, 2018.

[22] P. Ellis, "Extension of phase plane analysis to quantized systems," *IRE Transactions on Automatic*

Control, vol. 4, no. 2, pp. 43–54, 1959.

[23] M. T. Andrén, "A brief history of event-based control," University Lecture, 2016.

[24] G. Chen, D. Yao, H. Li, Q. Zhou, and R. Lu, "Saturated threshold event-triggered control for multiagent systems under sensor attacks and its application to uavs," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 69, no. 2, pp. 884–895, 2021.

[25] M. S. Mahmoud and Y. Xia, *Networked control systems: cloud control and secure control*. Butterworth-Heinemann, 2019.

[26] N. Gandhi, D. Saldana, V. Kumar, and L. T. X. Phan, "Self-reconfiguration in response to faults in modular aerial systems," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2522–2529, 2020.

[27] HookandLoop.com, "Duragrip® and velcro® brand hook and loop products," https://www.hookandloop.com/products, accessed: 2023-10-09.

[28] ——, "Velcro® brand sew on hook and loop fasteners," https://www.hookandloop.com/hook-and-loop-brands/velcro/velcror-brand-sew-on-hook-and-loop-fasteners, accessed: 2023-10-09.

[29] B. Lafferriere, G. Lafferriere, and M. N. Nguyen, *Introduction to Mathematical Analysis I*. Portland State University Library, 2022.

[30] J. Lygeros, C. Tomlin, and S. Sastry, "Hybrid systems: modeling, analysis and control," *Electronic Research Laboratory, University of California, Berkeley, CA, Tech. Rep. UCB/ERL M*, vol. 99, p. 6, 2008.

[31] R. Sanfelice, *Hybrid Feedback Control*, ser. Princeton Series in Applied Mathematics Series. Princeton University Press, 2021.

[32] J. Lee, *Introduction to topological manifolds*. Springer Science & Business Media, 2010, vol. 202.

[33] T. Hamel, R. Mahony, R. Lozano, and J. Ostrowski, "Dynamic modelling and configuration stabilization for an x4-flyer." *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 217–222, 2002.

[34] T. F. Banchoff, "Multivariable calculus," August 2008.

[35] M. Spiegel, *SCHAUM'S OUTLINE SERIES ON THEORETICAL MECHANICS*. McGraw Hill Education, 1982.

[36] S. Zacher, *Closed Loop Control and Management: Introduction to Feedback Control Theory with Data Stream Managers*. Springer Nature, 2023.

[37] R. Tedrake, *Underactuated Robotics*. MIT, 2023. [Online]. Available: https://underactuated.csail.mit.edu

[38] L. Lessard, "Lecture notes in linear systems," Fall 2019/20.

[39] J. P. Hespanha, *Linear systems theory*. Princeton university press, 2018.

[40] C. Hajiyev and S. Y. Vural, "Lqr controller with kalman estimator applied to uav longitudinal dynamics," *Scientific Research Publishing*, 2013.

A

Linear Quadratic Regulator

This appendix provides a comprehensive explanation of the control algorithm known as LQR. When addressing a control problem, the primary consideration is which algorithm will be employed to derive the desired control law, leading to the most optimal system response.

The initial concept to consider before explaining any type of controller is the distinction between open-loop and closed-loop control.

An open-loop system, also known as a non-feedback system, is a type of control system in which the output has no influence on the input signal. This means that the controller unit has no knowledge of the current state at any given time during execution. The disadvantage of open-loop control is the requirement that the behavior of the plant should be precisely known beforehand, as there is no feedback to recognize and correct possible errors [36].

A closed-loop system, also known as a feedback system, is a type of control system in which the output is “fed back” to the input, thereby altering the input and influencing the system’s performance.

The ability to access the current state is what makes this control methodology a powerful tool, capable of handling unforeseen errors, disturbances, obstacles, etc. The main downside of this approach, in comparison to the open-loop system, is that the added flexibility and robustness of the controller come at a greater cost in terms of implementation complexity.

Having clarified the two main branches of control algorithms, it becomes apparent that, for the case study, using a closed-loop strategy is mandatory. Without a feedback control system, the UAVs are unable to follow desired trajectories or make corrections for disturbances or tracking mistakes. Additionally, there are various closed-loop control algorithms, many of which will be discussed and compared in Section 1.3. In this section, we will provide a general explanation of the chosen algorithm, the LQR. A comprehensive, detailed explanation of the algorithm is beyond the scope of this introduction, but can be found in [37, Chapter 8].

When controlling a system using LQR, the linear feedback control law defining the optimal control vector u is expressed as,

$$u = -Kx \tag{A.1}$$

with $x \in \mathbb{R}^n$ representing the state vector, the notation used in this section was selected to enhance the readability of the equations. It should be noted, however, that both the state and control vectors should be understood as functions of time, t.

In order to optimize the state variables and control input, it is found the u (and x from (A.1)) that minimizes the following cost function,

$$J = \int_0^{+\infty} (x^T Q x + u^T R u)\, dt, \tag{A.2}$$

where $Q \in \mathbb{R}^{n \times n}$ is a positive semidefinite matrix and $R \in \mathbb{R}^{m \times m}$ is a positive definite matrix. The weight matrices Q and R are chosen to control each state, effectively prioritizing different aspects of the response. For instance, adjusting the values of Q and R can result in faster or slower responses. To prioritize state response, we can increase the values of Q in relation to the values of R, resulting in a faster response at the expense of a higher demand on the control input.

The Figure A.1 is a block diagram of a closed-loop LQR control system. From the diagram, the logic behind this control algorithm becomes apparent. Defining the correct feedback gain, K, allows the system to follow a desired reference signal.

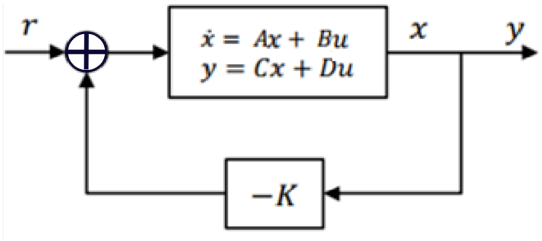

Figure A.1: Block diagram of an LQR controller. Image from [7].

Some pre-conceived knowledge to take into consideration for the comprehension of this section is that the LQR requires a linear system in order to find the optimal feedback gain K. In the case of a non-linear system (such as most real world systems) the LQR can be used in a linearization of the original system around a equilibrium point.

The main contribution the LQR algorithm offers is in the definition of the feedback gain K. The control problem hinges on finding the most suitable values for K. Due to the linear properties of the state system and the quadratic cost function, a solution can be derived for the optimization problem. The controller methodology explanation draws significantly from the lecture [38, Lecture 20], with [39] as the primary literature source.

$$\begin{aligned} \min_{u \in \mathbb{R}^m} \quad & \int_0^{+\infty} (x^T Q x + u^T R u)\, dt \\ \text{s.t.} \quad & \dot{x} = Ax + Bu \end{aligned} \tag{A.3}$$

To obtain the solution to the optimization problem, we begin by rewriting the cost function. This is done without altering its value, but rather by introducing a positive scalar through addition and subtraction, such that,

$$x_0^T P x_0 - x_0^T P x_0 + \int_0^{+\infty} (x^T Q x + u^T R u)\, dt \tag{A.4}$$

Where $P \in \mathbb{R}^{n \times n}$ is a positive semidefinite matrix, where from the linear plant equation $\dot{x} = Ax + Bu$ we can show,

$$\begin{aligned} -x_0^T P x_0 &= \int_0^{+\infty} \frac{d}{dt} \left(x^T P x\right)\, dt \\ &= \int_0^{+\infty} \dot{x}^T P x + x^T P \dot{x}\, dt \\ &= \int_0^{+\infty} (Ax + Bu)^T P x + x^T P (Ax + Bu)\, dt \end{aligned}$$

By inputting this equality into the expression (A.4) we can further simplify it as,

$$\begin{aligned} & x_0^T P x_0 + \int_0^{+\infty} \left[(Ax+Bu)^T P x + x^T P (Ax+Bu) + x^T Q x + u^T R u \right] \, dt \\ = & x_0^T P x_0 + \int_0^{+\infty} \left[x^T (A^T P + PA + Q) x + u^T R u + x^T P B u + u^T B^T P x \right] \, dt \end{aligned}$$

To finish the simplification of the cost function (J) that we aim to minimize we take the following equality

$$u^T R u + x^T P B u + u^T B^T P x = (u + R^{-1} B^T P x)^T R (u + R^{-1} B^T P x) - x^T (P B R^{-1} B^T P) x$$

and end up with a final expression equal to the original cost function (A.2), as follows,

$$x_0^T P x_0 + \int_0^{+\infty} \left[x^T (A^T P + PA + Q - P B R^{-1} B^T P) x + (u + R^{-1} B^T P x)^T R (u + R^{-1} B^T P x) \right] \, dt \quad \text{(A.5)}$$

Due to the nature of the matrices Q and R, the integrand of the cost function in Equation (A.2) is non-negative. Consequently, the optimal state and control vectors will result in a non-negative J. The first term of the expression (A.5) does not depend on the control vector, and for this reason, it can be ignored when considering the minimization (A.3). The first term of the integrand needs to be minimized; in this case, the optimal possibility would be for it to be equal to zero. The same applies to the second term of the integrand. In this case, we can refer to Equation (A.1) and obtain,

$$(u + R^{-1} B^T P x)^T R (u + R^{-1} B^T P x) = 0 \Longrightarrow K = R^{-1} B^T P$$

In order to minimize the first term of the integrand we do the same and find that the solution to the optimization problem is finding the matrix P that satisfies the Algebraic Riccati Equation (ARE),

$$A^T P + PA + Q - P B R^{-1} B^T P = 0 \quad \text{(A.6)}$$

Solving the ARE results in multiple solutions, i.e. multiple possibilities of P that solve (A.6). Consequently, this results in multiple possibilities for the feedback gain. The last step into finding the optimal K is by evaluating the stability of the closed loop system.

$$\dot{x} = Ax + Bu = (A - BK)x = A_{CL} x$$

In other words, the final step is finding K such that the eigenvalues of A_{CL} are all negative.

To summarize the complete step-by-step process for defining the LQR controller:

1. Define the weight matrices Q and R.

2. Solve the ARE to obtain all possible values of P.

3. Compute all possible values of K.

4. Finally, evaluate the closed-loop systems to find the K that yields stability.

In practice, for the results obtained in Chapter 4, the feedback gain was calculated using the `lqr` function in MATLAB [40].

B

Additional Figures

This appendix provides additional plot figures, offering further insight into the simulations previously presented in Chapter 4, along with some supplementary simulations mentioned in the same chapter.

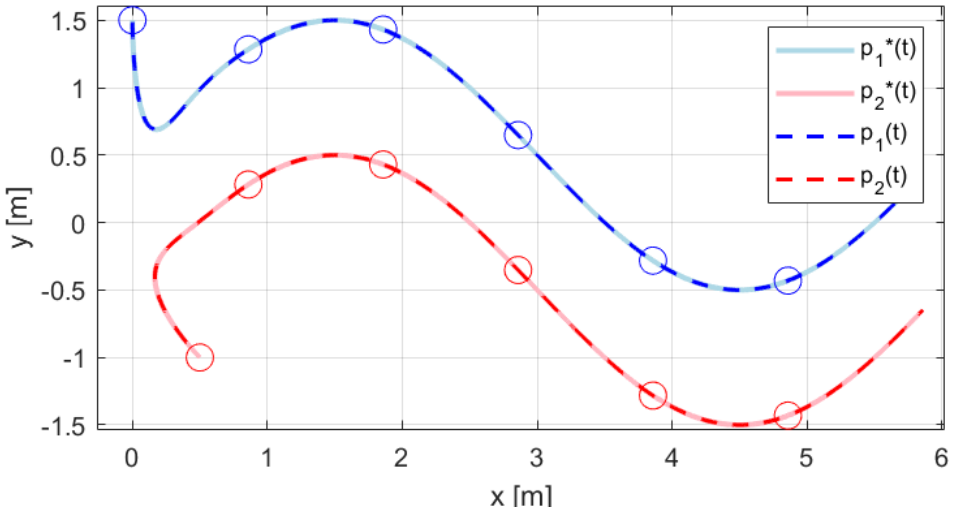

Figure B.1: Plot of the positions ($p_1(t)$ and $p_2(t)$) and corresponding desired positions ($p_1^\star(t)$ and $p_2^\star(t)$), for the case of Figure 4.1.

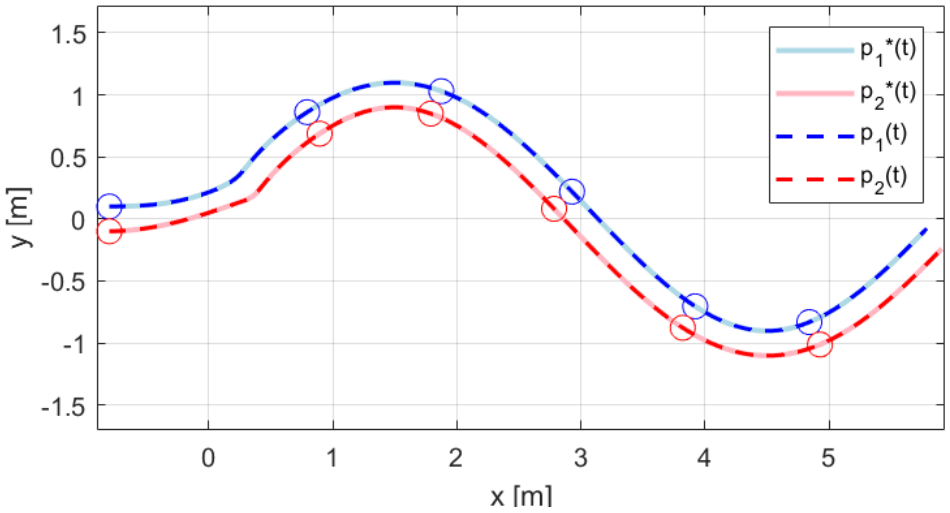

Figure B.2: Plot of the positions ($p_1(t)$ and $p_2(t)$) and corresponding desired positions ($p_1^\star(t)$ and $p_2^\star(t)$), for the case of Figure 4.2.

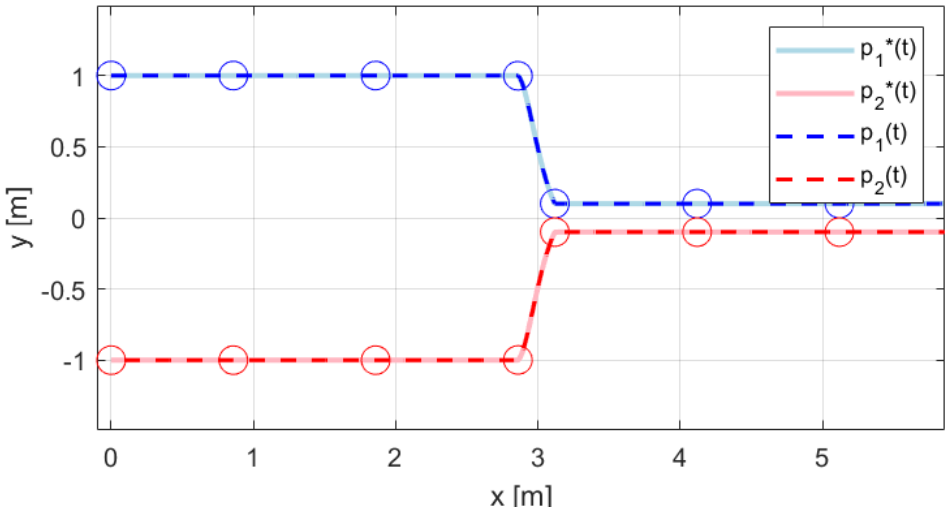

Figure B.3: Plot of the positions ($p_1(t)$ and $p_2(t)$) and corresponding desired positions ($p_1^\star(t)$ and $p_2^\star(t)$), for the case of Figure 4.3.

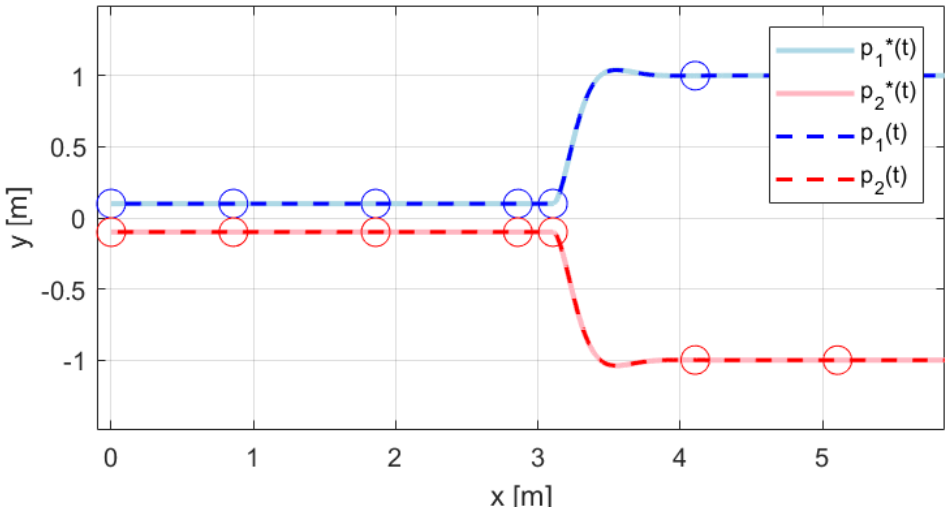

Figure B.4: Plot of the positions ($p_1(t)$ and $p_2(t)$) and corresponding desired positions ($p_1^\star(t)$ and $p_2^\star(t)$), for the case of Figure 4.5.

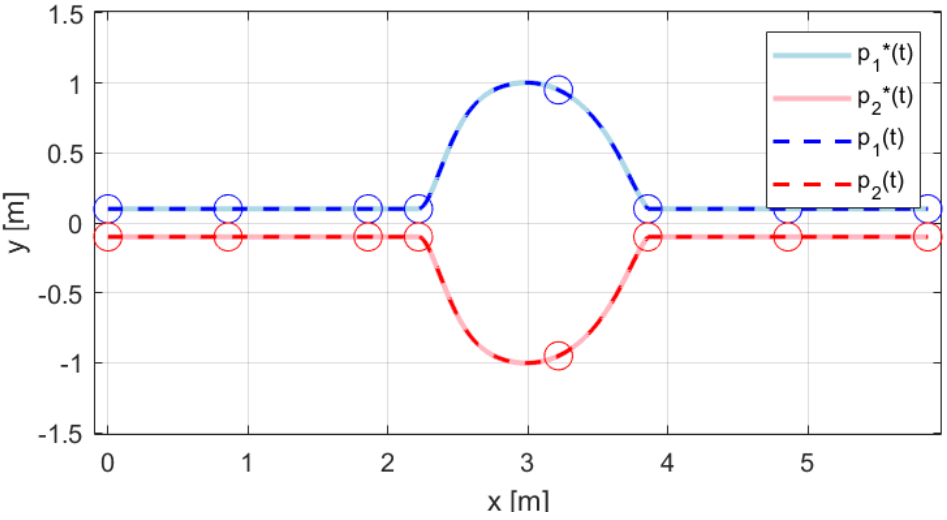

Figure B.5: Plot of the positions ($p_1(t)$ and $p_2(t)$) and corresponding desired positions ($p_1^\star(t)$ and $p_2^\star(t)$), for the case of Figure 4.7.

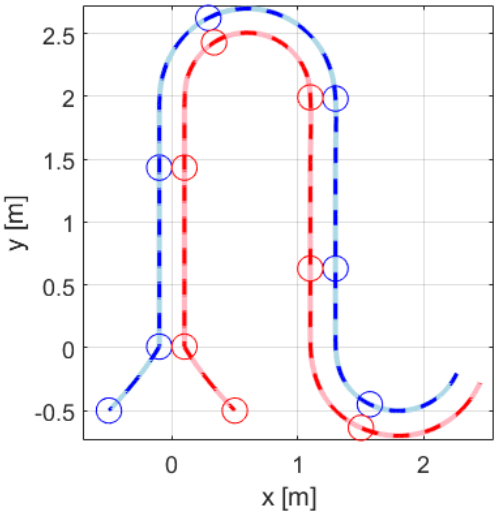

Figure B.6: Plot of the positions ($p_1(t)$ and $p_2(t)$) and corresponding desired positions ($p_1^\star(t)$ and $p_2^\star(t)$), for the case of Figure 4.10.

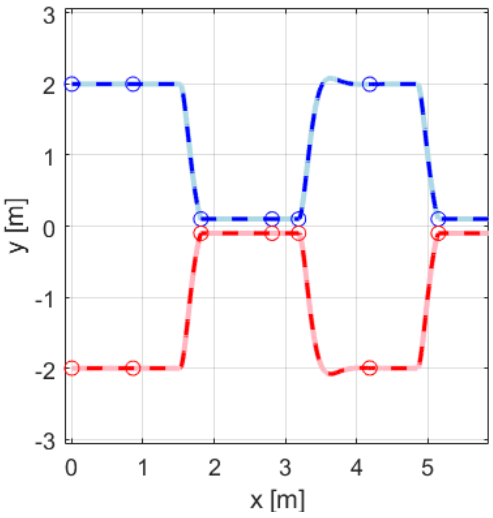

Figure B.7: Plot of the positions ($p_1(t)$ and $p_2(t)$) and corresponding desired positions ($p_1^\star(t)$ and $p_2^\star(t)$), for the case of Figure 4.11.

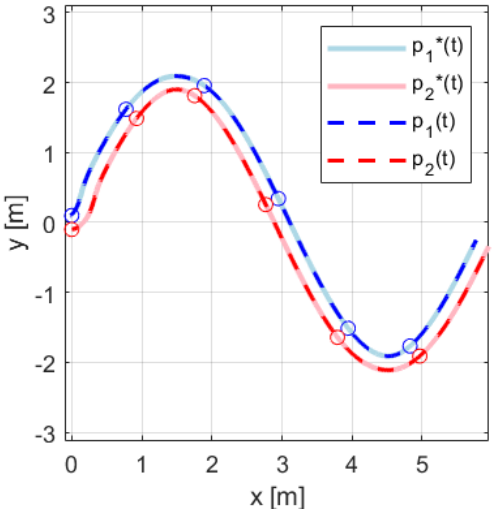

Figure B.8: Plot of the positions ($p_1(t)$ and $p_2(t)$) and corresponding desired positions ($p_1^\star(t)$ and $p_2^\star(t)$), for the case of Figure 4.12.

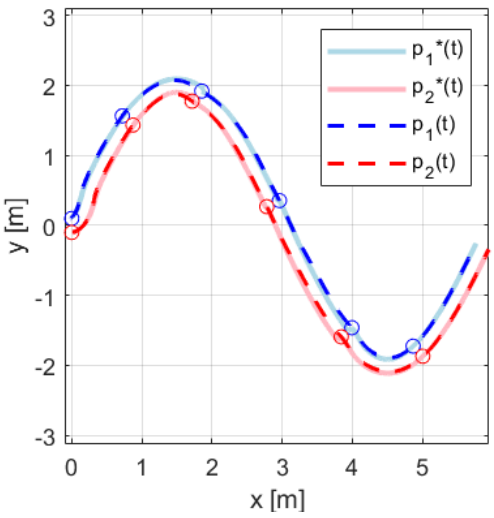

Figure B.9: Plot of the positions ($p_1(t)$ and $p_2(t)$) and corresponding desired positions ($p_1^\star(t)$ and $p_2^\star(t)$), for the case of Figure 4.14.

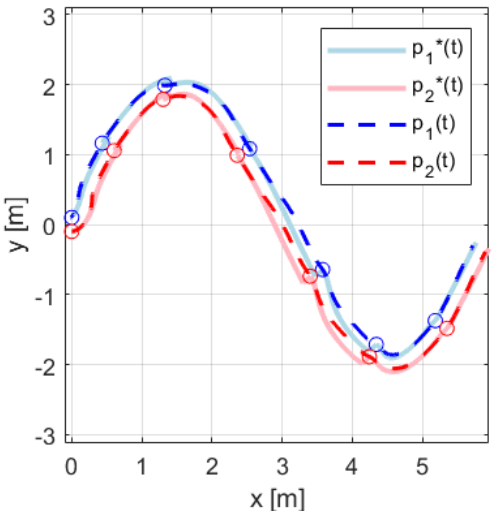

Figure B.10: Plot of the positions ($p_1(t)$ and $p_2(t)$) and corresponding desired positions ($p_1^\star(t)$ and $p_2^\star(t)$), for the case of Figure 4.16.

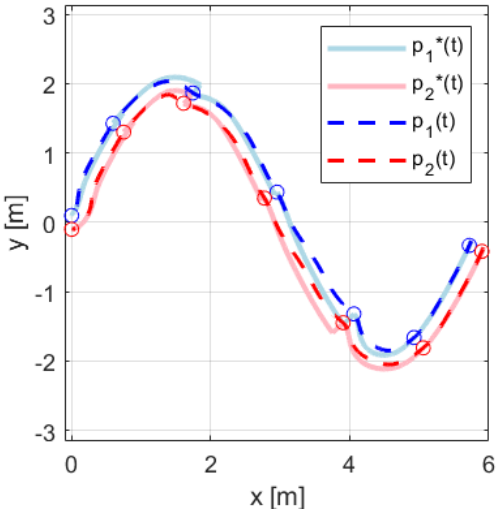

Figure B.12: Plot of the positions ($p_1(t)$ and $p_2(t)$) and corresponding desired positions ($p_1^\star(t)$ and $p_2^\star(t)$), for the case of Figure 4.20.

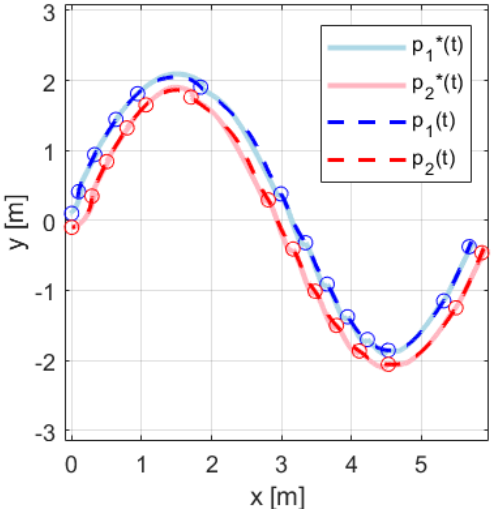

Figure B.11: Plot of the positions ($p_1(t)$ and $p_2(t)$) and corresponding desired positions ($p_1^\star(t)$ and $p_2^\star(t)$), for the case of Figure 4.18.