Università
degli Studi
di Padova

Dipartimento
di Ingegneria
dell'Informazione

Control of Tethered Aerial Robots in presence of disturbances and sampled communication: theory and experiments

Ph.D. candidate
Enrica Rossi

Advisor
prof. Luca Schenato

Director & Coordinator
prof. Andrea Neviani

Ph.D. School in
Information Engineering

Department of
Information Engineering
University of Padova
2020

University of Padova
Department of Information Engineering

Università
degli Studi
di Padova

Ph.D. course in: Information Engineering
Curriculum: Information Science and Technology

Control of Tethered Aerial Robots in presence of disturbances and sampled communication: theory and experiments

Director: Prof. Andrea Neviani
Advisor: Prof. Luca Schenato

Ph.D. candidate: Enrica Rossi
Cycle: 33nd (2017-2020)

Year: 2020

List of Figures

List of Tables

Abstract

In this work we address the problem of advance control of tethered aerial robots in the context of load transportation and landing.

Firstly, we design and implement a two layer hierarchical controller for a tethered aerial vehicle. In particular, a challenging problem of smoothly landing on a inclined plane is considered. Such problem is characterized by nonlinear control objective and constraints which require advanced control techniques like Nonlinear Model Predictive Control (NMPC), that is known to be computationally demanding. We propose a control architecture composed by two layers: an NMPC operating at 100Hz is employed to generate feasible trajectories on-line and a simplified local controller, working at 500Hz, is used to track them. The effectiveness of this hierarchical control structure is demonstrated by a closed-loop simulation. Moreover, the rejection of external disturbances, such as the wind, is a peculiar characteristic of the proposed control strategy.

Secondly, we propose a feedback-based motion planner for a class of multi-agent systems with a sparse kinematics structure. In other words, the agents are coupled together only by a common object, that can be a load to be transported (in this case there is a physical link among each agent and the object), a reference point in the space or a robot to be escorted (in these cases a constraint is defined and allows to express the agents configurations with respect to the object one). The goal is to steer this object into a desired configuration. We show that the proposed strategy can be applied to diverse types of systems: both aerial and ground ones, with or without physical connections among the agents. In this chapter we study *square* systems, whose differential kinematics is characterized by a square Jacobian matrix: this assumption helps to simplify our analysis. We suppose that a global motion planner generates a sequence of desired configurations that satisfy constraints as obstacles and singularities avoidance. Then, a local planner receives these references and generates the desired agents velocities, which are converted into force inputs for the vehicles. We focus on the local planner design both in the case of continuously available measurements and when they are transmitted to the agents via sampled communication. For the latter problem, we propose two strategies. The first is the discretization of the continuous-time strategy that preserves stability and guarantees exponential convergence regardless of the sampling period. In this case, the planner gain is static and computed off-line. The second strategy requires to collect the measurements from all sensors and to solve online a set of differential equations at each sampling period. However, it has the advantage to provide doubly exponential convergence. Numerical simulations of these strategies are provided for the cooperative aerial manipulation of a cable-suspended load. In addition, we show experimental results obtained by applying our strategies on a real system.

We are also interested to extend the analysis of the problem just introduced, where we developed two strategies to implement point-stabilization control for a class of systems with a sparse kinematic structure. In particular, we considered the scenario where measurements are transmitted via wireless. Notice that in the previous case we assumed that the Jacobian matrix of the kinematic model was square, in order to simplify the analysis. In this work we still consider a class of multi-agent systems with a sparse kinematic structure in the scenario where measurements are available at sampling instants,

but we extend the analysis to systems characterized by a rectangular Jacobian matrix. In practice, this means that the number of robots is redundant, i.e. a desired system configuration can be obtained by different configurations of the robots. We propose a solution to this problem: if unfeasible velocities are given as references to the agents, they are projected on a space of feasible values. We firstly propose a kinematic model which is consistent with the overall systems dynamics under high-gain control and then we provide sufficient conditions for the exponential stability and monotonic decrease of the configuration error under different norms. Then, we test the proposed controllers on the full dynamical system showing the benefit of local communication. Finally, instead of simplifying the wireless transmission with sampled communication, we exploit a simulator for real-time control systems, where a wireless network can be simulated, by setting all the parameters as in a real transmission.

The last contribution is an inverse-kinematics controller for a class of multi-robot systems in case of sampled communication. The scenario is similar to the one described in the previous lines, but this time we are interested to trajectory tracking instead of point-stabilization control, where a sequence of reference velocities was *planned* for the system in order to reach a constant desired configuration. Thus, we develop a Sampled Communication aware Inverse Kinematic *controller* (SIKM) specific for trajectory tracking that, given a motion of reference (in configuration space), receives the sampled measurements from the system and generates the desired robots velocities along the whole trajectory. We provide two strategies to choose the controller gain. In one case it is computed off-line: once the sampling time is assigned, we propose a method to compute the controller gain that guarantees stability and asymptotic zero-tracking error. In this case the controller is implementable in a distributed way. In the second scenario the gain is updated on-line and only a centralized implementation can be achieved: in this scenario we practically show by means of simulations that the proposed control method provides a faster convergence rate while tracking the assigned trajectory. Notice that both strategies work with no communication among robots. Numerical simulations of the two techniques are provided for the cooperative aerial manipulation of a cable-suspended load. We also compare our strategies to other possible solutions, showing better performance and stability guarantees.

Sommario

In questa tesi proponiamo diverse soluzioni per controllare i robot aerei vincolati. In primo luogo, progettiamo e implementiamo un controllore gerarchico a due livelli per un veicolo aereo vincolato. In particolare, viene considerato un problema di atterraggio su un piano inclinato. Tale problema è caratterizzato da una funzione obiettivo non lineare e vincoli che richiedono tecniche di controllo avanzate come Nonlinear Model Predictive Control (NMPC), noto per essere computazionalmente impegnativo. Proponiamo un'architettura di controllo composta da due livelli: un NMPC che opera a 100 Hz viene utilizzato per generare traiettorie ammissibili in tempo reale e un controllore locale semplificato, che lavora a 500 Hz, viene utilizzato per inseguirle. L'efficacia di questa struttura di controllo gerarchica è dimostrata in simulazione dove si può vedere che il sistema controllato riesce a gestire disturbi esterni, come il vento, ma anche incertezze di modello.

In secondo luogo, proponiamo un pianificatore di movimento basato sulla retroazione per una classe di sistemi multi-agente con una struttura cinematica sparsa. In altre parole, gli agenti sono accoppiati tra loro da un oggetto comune, che può essere un carico da trasportare (in questo caso c'è un collegamento fisico tra ogni agente e l'oggetto), un punto di riferimento nello spazio o un robot da scortare (in questi casi viene definito un vincolo che consente di esprimere le configurazioni degli agenti rispetto a quella dell'oggetto). L'obiettivo è quello di guidare questo oggetto nella configurazione desiderata. Mostriamo poi che la strategia proposta può essere applicata a diversi tipi di sistemi: sia aerei che terrestri e con o senza connessioni fisiche tra gli agenti. In questo capitolo studiamo i sistemi quadrati, la cui cinematica differenziale è caratterizzata da una matrice Jacobiana quadrata: questa ipotesi aiuta a semplificare la nostra analisi. Supponiamo che un pianificatore di movimento globale generi una sequenza di configurazioni desiderate che soddisfino vincoli come evitare ostacoli e singolarità. Poi un pianificatore locale riceve questi riferimenti e genera le velocità di riferimento per gli agenti: queste vengono in seguito convertite in valori di forza che i veicoli devono esercitare. Ci concentriamo sulla progettazione del pianificatore locale sia nel caso in cui le misure sullo stato del sistema siano disponibili nel continuo sia quando vengono trasmesse agli agenti tramite comunicazione campionata. Per quest'ultimo problema, proponiamo due soluzioni. La prima è la discretizzazione della strategia a tempo continuo che preserva la stabilità e garantisce la convergenza esponenziale indipendentemente dal periodo di campionamento. In questo caso, il guadagno del pianificatore è statico e calcolato off-line. La seconda strategia richiede di conoscere le misure dell'intero stato del sistema e di risolvere in tempo reale una serie di equazioni differenziali ad ogni periodo di campionamento. Tuttavia, ha il vantaggio di fornire una convergenza doppiamente esponenziale. Mostriamo poi i risultati di alcune simulazioni numeriche dove queste strategie vengono utilizzate per la manipolazione aerea cooperativa di un carico sospeso tramite cavi. Inoltre, mostriamo i risultati sperimentali ottenuti applicando le nostre soluzioni su un sistema reale.

In seguito viene sviluppata un'interessante estensione del problema appena introdotto. In particolare consideriamo uno scenario più generico: una classe di sistemi multi-agente con una struttura cinematica sparsa deve spostarsi da una configurazione a un'altra e le misure sullo stato del sistema sono disponibili ad istanti campionati. La differenza sostanziale è che ora assumiamo che la matrice Jacobiana del modello cinematico sia

rettangolare. Questo aspetto complica l'analisi sulla stabilità poiché ciò significa che il numero di robot è ridondante, ossia una velocità di configurazione desiderata può essere ottenuta da diverse configurazioni dei robot. Proponiamo quindi una soluzione a questo problema: se delle velocità non ammissibili sono date come riferimento agli agenti, esse vengono proiettate su uno spazio di valori ammissibili. Sotto l'ipotesi di separazione delle scali temporali, proponiamo un modello cinematico che sia coerente con la dinamica complessiva del sistema sotto l'ipotesi di guadagno elevato e poi forniamo condizioni sufficienti per la stabilità esponenziale e la convergenza a zero dell'errore di configurazione, considerando diverse norme. Quindi, testiamo la tecnica proposta sul sistema dinamico completo, evidenziando i vantaggi della comunicazione locale. Infine, invece di semplificare la trasmissione wireless con comunicazione campionata, sfruttiamo un simulatore per sistemi di controllo in real-time, dove è possibile simulare una rete wireless, impostando tutti i parametri come in una trasmissione reale.

Infine sviluppiamo un controllore basato sulla cinematica inversa per una classe di sistemi multi-robot in caso di comunicazione campionata. Lo scenario è simile a quello descritto precedentemente, ma ora siamo interessati all'inseguimento di una traiettoria invece che al raggiungimento di una configurazione costante. Pertanto, sviluppiamo un controllore che permetta l'inseguimento della traiettoria di riferimento tenendo conto della comunicazione campionata: in pratica, data una traiettoria (nello spazio di configurazione), riceve le misure campionate dal sistema e genera le velocità desiderate dei robot lungo l'intera traiettoria. Anche in questo caso sviluppiamo due strategie per scegliere il guadagno del controllore. In un caso viene calcolato off-line: una volta assegnato il tempo di campionamento, proponiamo un metodo per calcolare il guadagno del controllore che garantisca stabilità e convergenza asintotica dell'errore di tracking a zero. In questo caso il controllore è implementabile in modo distribuito. Nel secondo scenario il guadagno viene aggiornato in real-time e si può ottenere solo un'implementazione centralizzata: d'altra parte, come si evince da simulazioni numeriche, il metodo di controllo proposto fornisce una velocità di convergenza maggiore. In ogni caso entrambe le strategie funzionano senza comunicazione tra i robot. Come accennato, mostriamo i risultati di simulazioni numeriche delle due tecniche per la manipolazione aerea cooperativa di un carico sospeso via cavo. Confrontiamo inoltre le nostre strategie con altre possibili soluzioni, mostrando che possiamo ottenere migliori prestazioni e garanzie di stabilità.

Acknowledgments

I would like to thank my advisor prof. Luca Schenato for his guidance and support during my PhD. I care to thank him for his patience and giving me inspiration to deal with challenging and complicated problems.

I would also like to deeply thank prof. Ruggero Carli for cooperating with Luca to help me in developing my thesis and spending time to discuss about interesting solutions, also by creating collaborations with colleagues to exploit at best our knowledge and experience.

I would also like to express my gratitude to Antonio Franchi and Marco Tognon who supervised me during my staying at LAAS-CNRS in Toulouse, France, after hosting me to develop my master thesis in 2015-2016. They inspired me with new approaches and made me see my work from new perspectives. They made me feel like at home during all the months I spent there. I want also to thank Dario Sanalitro with whom I collaborated at LAAS-CNRS: he helped me to set-up the experiments I needed for my work and he revealed to be a very gentle person, always available.

Another special thank goes to all the colleagues and friends that I met at the Department of Information Engineering in Padova: since my arrival they made me feel welcome and they always helped me if I needed some advice, both in academic and personal life.

A thank goes to each of the professors and researchers in the University of Padova, since, during these eight years as an academic student they all have forged my mind for the worst case scenarios and contributed to grow me as a cultured scholar, aiming for an advanced education.

I would like to express an other special thank to all the friends who I met before and during the last three years: I spent a really great time with them; in particular, I would like to thank Valeria Rubin and Marta Nardo who have been great friends since I were very young, Davide Bicego, Selma Zamoum and Henri Prifti with whom I shared my best and funniest moments in Toulouse.

A very important thank goes to my family with whom I was always free to talk about everything and who care a lot about my happiness. They supported my during my whole academic career and personal life. They taught me that working hard gives you more possibilities and choices and, most of all, satisfaction.

Finally, I would like to thank the person who knows me the best and supported me during the half of my life, my fiancé Simone Prataviera. He always understood when I needed to stay at home to study and work instead of going out and having fun. He has always been there for me both in the easiest and more difficult moments. I really appreciate everything he has done for me and I wish we will continue to share our lives as we have done until now.

Contents

1

Introduction

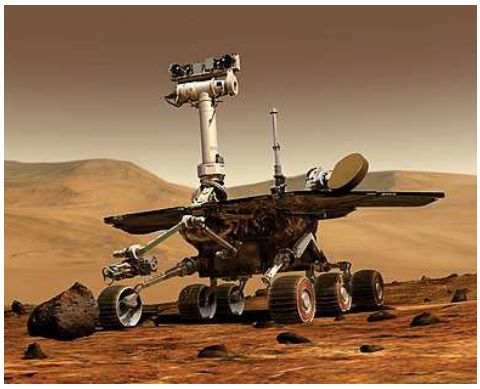
Contents

Nowadays, robotics is part of our lives, in different scenarios; the most common environment is the industrial one, where different categories of robots are employed, depending on the specific application. They allow to reduce human effort, to reduce the time required to accomplish an assigned task, but also to avoid the employees to perform repetitive, stressful and boring operations. Moreover, they can be exploited to perform maneuvers which would result dangerous for human beings. The most frequently encountered robot are the manipulators which are of a great interest both in research and in industry. This class of robots is mainly used for pick-and-place tasks since they are characterized by high accelerations, repeatability, and accuracy. Autonomous robot vehicles are also common and can be used on land, in the water and in the air. The land-based mobile robots are mostly applied in artificial environments, like apartments, hospitals, museums, etc. but also on highways and even pathless grounds. Most mobile robots are nevertheless used on flat ground where they move thanks to the wheels they are endowed with, which can be designed to enable omnidirectional movements. Robot vehicles examples are vacuum cleaners, autonomous lawn mowers, intelligent guides through department stores or museums, attendants in clinical centers, space rovers, or autonomous cars. Some examples of the mentioned systems are reported in Fig.1.1.

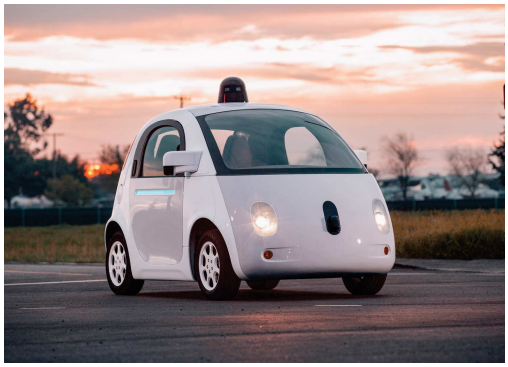
A very special class of robotic vehicles is the one of Unmanned Aerial Vehicles (UAVs) which are aerial vehicles that can work fully or partially autonomously. Research on aerial robotics has attracted an increasing interest over the last decade. This is mainly motivated by recent technological progresses and scaling of the hardware that have made aerial robots usable in a wide range of applications, e.g., search and rescue, patrolling, environmental monitoring, providing assistance in hostile environments, etc.

The first studies appeared in the 1900s: the first UAV was built by the Americans Lawrence and Sperry in 1916. They gave birth to the first attitude control by developing a gyroscope to stabilize the body. The device was called *aviation torpedo* and it could fly for about 50 kilometers. However, it seems that the UAVs were not technically mature to be exploited during the Second World War. Thus, their actual development and study began at the end of the 1950s and continued during the Vietnam War until the 1970s. After the War, other countries like Israel and the U.S. began to develop smaller

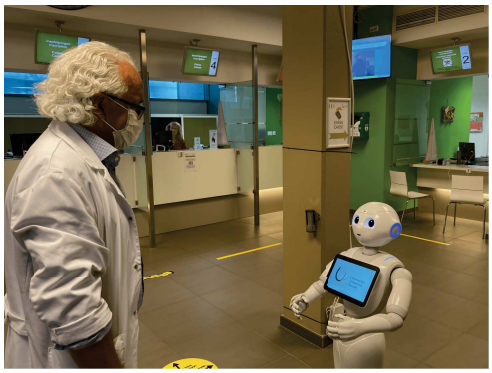
(a) *Dalmec industrial manipulator https://www.whitech-it.com/.*



(b) *Opportunity rover https://mars.nasa.gov/mer/mission/rover/.*

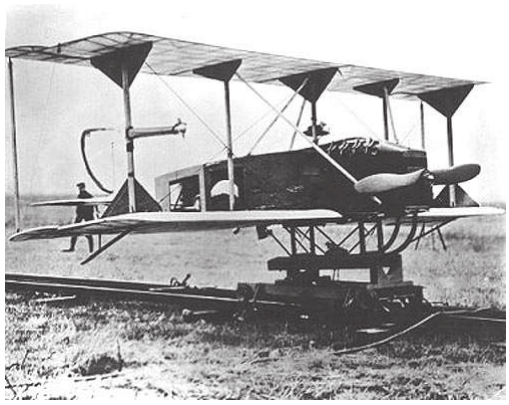


(c) *Google autonomous car https://waymo.com/.*

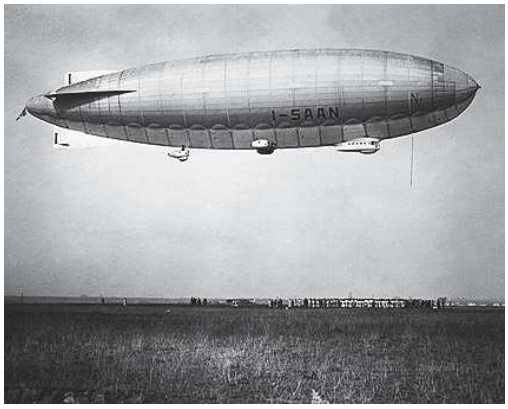


(d) *Autonomous robots check patients in at Belgium hospitals.*

Figure 1.1. Examples of robotics in our lives.

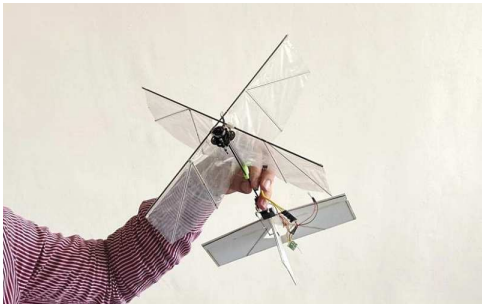


(a) *First UAV in the world, 1916.*



(b) *I-SAAN, a semi-rigid Italian-built airship.*

(c) *Applied Aeronautics Albatross Fixed Wing UAV.*



(d) *Example of flapping-wing UAV from National Chiao Tung University.*

Figure 1.2. Examples of Unmanned Aerial Vehicles.

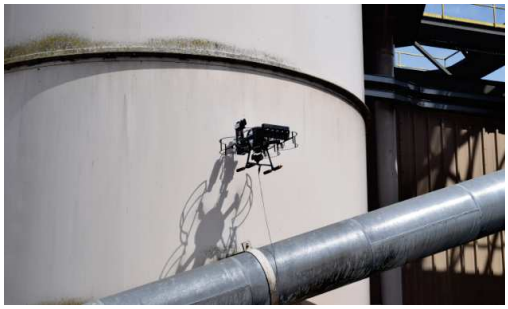
and cheaper versions of these vehicles. They also carried video-cameras which sent images to a human operator. Since then, the study and development of the UAVs was related to the military field. The main goal was to increase the flight endurance and the admitted payload. Depending on the vehicle configuration and performance, four main categories of Unmanned Aerial Vehicles have been defined: fixed-wing UAVs which have high endurance, can fly at high cruising speeds but need a runaway to take-off and land; then, the rotary-wing UAVs, also called Vertical Take-off and Landing UAVs (VTOL), which allow high maneuverability and hovering. An other category are the Blimps which are lighter than air and are characterized by lower flight speed but long endurance; finally, flapping-wing UAVs have flexible small wings whose movements are inspired by birds. Figure 1.2 represents some examples of the mentioned aerial vehicles. A more detailed survey on the history of such systems can be found in Kendoul (2012). Even though the improvement of these vehicles performance in terms of endurance, maneuverability and sensing was strictly related to military purposes, anyway their application began to extend to the civil area: for example they were used as remote sensors to inspect buildings and terrains, to map some areas of interest or even to surveil road traffic and coastal borders; in addition, they began to be exploited as autonomous systems able to interact with the surrounding environment, for instance to manipulate and transport objects Maza, Kondak, Bernard, and Ollero (2010a); Michael, Fink, and Kumar (2011) or to search and rescue during disasters. The field of application of the UAVs spread very quickly due to the improved sensing and computing equipment and to the lower cost and

Figure 1.3. The first British fixed-wing VTOL aircraft, 1958.

lighter weight. These characteristics, indeed, made these vehicle accessible to a wider community, both in the industrial and research environment.

An interesting category among the ones cited before is of course the VTOL vehicles, Figure 1.3: they are endowed with high maneuverability and can hover in place and they do not need any runway to take-off and land. These aspects are fundamental when such vehicles have to fly in environments which are small or with many obstacles. Besides these useful properties, standard VTOL vehicles generate a thrust force along a fixed direction with respect to the body frame; for this reason they belong to the class of under-actuated vehicles, that is the state cannot be fully controlled. Indeed, the attitude and the position are not independent one from the other. This is evident when the vehicle needs to move horizontally: it firstly needs to rotate such that a component of the generated thrust is horizontal, hence an acceleration along the same direction of this component allows the robot to translate toward the desired direction. The same happens when it needs to stop moving forward: in this case an inverse rotation would restore the thrust direction (vertically this time) such that the horizontal acceleration becomes zero. Several works studied VTOL because their characteristic (under-actuation) makes it difficult to develop a controller able to immediately reject external disturbances while stabilizing the system. To this aim, in Bicego (2019) the authors propose the possibility of orienting the aerial vehicle actuators in a different manner, in order to take advantage of the Multi-Directional Thrust (MDT) force capability: this allows the vehicles to perform maneuvers where rotation and translation are completely independent. As a consequence, novel control strategies must be designed to drive these aerial robotic systems.

Among all the challenges related to aerial vehicles, some attention has been devoted to tethered flying solutions (see Maza, Kondak, Bernard, and Ollero (2010b)Pinkney, Hampel, and DiPierro (1996), Muttin (2011)), like the case of one vehicle tethered to a ground station via a taut cable. An example is reported in Figure 1.4a. Tethered aerial vehicles can be exploited for reconnaissance capabilities, surveillance and high-bandwidth communications. In such situation, the presence of the cable has multiple purposes: firstly, to be used as a high-bandwidth channel: the cable ensures a more stable, immune to interference and secured communication w.r.t. wireless connection hence real-time high-speed data transmission can be achieved, allowing for accident detection, emergency communication and coordination of rescue operations, see Figure 1.4b. In the telecommunications field tethered UAVs can be used as airborne relays to



(a) *Example of a tethered drone used for inspection.*

(b) *Tethered Station for Unlimited Endurance Flights.*

Figure 1.4. Additional examples of Unmanned Aerial Vehicles.

create temporary networks thanks to their high-speed data transfer. In this case, cellular coverage can be extended in crowded environments and connectivity can be improved in remote areas. Secondly, the cable can be used to provide energy: it allows to power the robots for how long it is needed Eli, without time limitations which instead characterize batteries usage. Thus long surveillance operations are possible by day and night, allowing to monitor wide areas. An other useful application of tethered aerial vehicles is to make them wash or paint buildings at high altitudes: of course this is very helpful because reduces a lot the risk for the human operators. In addition, being the vehicles autonomous, their employment reduces time and costs w.r.t. the scenario where some people have to perform the task. Notice that the cable that tethers the robot to the ground or to a moving station is useful both to send the necessary fluids and to power the robot. Finally, tethered solutions improve hover stability in presence of wind or during dangerous maneuvers, e.g., take off and landing from moving platforms as a ship when the sea is rough. A solution to this problem is proposed in Oh, Pathak, Agrawal, Pota, and Garratt (2006), where the authors face the problem of autonomous landing of a helicopter on a ship deck using a tether. In this scenario, the landing maneuver on a sloped (not flat) surface is a challenging problem for a VTOL (Vertical Take-Off and Landing) vehicle due to its intrinsic underactuation. Indeed, since the aerial vehicle needs to be oriented parallel to the surface when it is close to it, both attitude and position require to be controlled. So, tethered vehicles represent an interesting solution to make landing operations efficient and safe also in presence of non-flat terrains. As mentioned before, a cable improves the stability of the system in presence of disturbances: in Sandino, Bejar, Kondak, and Ollero (2014), for instance, the wind effect on a tethered unmanned helicopter is considered to improve hovering performance. Besides the disturbance rejection issue, hovering and trajectory tracking are interesting tasks for a robot: for example, in Lupashin and D'Andrea (2013), the authors develop a control strategy to stabilize the robot at a given constant value, while in Tognon and Franchi (2015) and Tognon, Dash, and Franchi (2016b), they make the quadrotor follow a desired trajectory assigned a-priori. In the mentioned works, the desired configurations or paths are generated offline. However, in real cases, it may happen that external, unpredictable forces are exerted on the system or an obstacle could be found in the path. To handle such events, the original trajectory, that did not predict these disturbances, must be recomputed taking into account the new information. This is faced in Alexis, Nikolakopoulos, and Tzes (2012) and Alexis,

Nikolakopoulos, and Tzes (2012), where a path tracking problem is solved by using the MPC strategy for an unmanned quadrotor subject to atmospheric disturbances: anyway, in neither case tethered vehicles are considered. Notice that in the previous lines we discussed about single-agent systems, where a task is assigned to a single aerial robot. Besides the interesting applications which can be implemented in this case, some scenarios require to exploit multiple robots to fulfill a common task. The tasks where single-agent systems are usually employed require strict specifications in terms of accuracy and speed of completion, which in some cases can be impossible. Moreover, the risk of mission failure is significantly high due to possible critical sensor or actuation faults. Instead, the use of multiple vehicles may significantly speed up the task completion, improve the perception, and enhance the fault tolerance capabilities of the overall multi-robot system. Typical fields of applications of aerial multi-robot systems are: mapping, localization and navigation; distributed cooperative perception and active sensing; decentralized coordination and cooperation; transporting big objects, surveillance of vast areas. Thanks to decentralised nature of multi-robot systems, the robots apply spatially distributed forces at different locations around objects. The independent actions of different agents can potentially generate a group skill that is very difficult to be achieved by a single robot. This property is particularly important in cooperative transport tasks, where the independent exertion of multiple pushing/pulling forces in different points of an object allow the group to generate precise translation/rotation maneuvers in order to avoid obstacles during transport. Recently, aerial manipulators or physical links like cables, has become a very popular topic. They can be used for aerial manipulation of large objects, for which cooperative approaches are usually applied to overcome the limited payload of a single platform, hence allowing to lift larger and heavier loads Kumar and Michael (2012); Maza, Kondak, Bernard, and Ollero (2010c). This solution is safer and less expensive with respect to the use of a single more powerful aerial vehicle. Different strategies were proposed in literature to tackle this problem. For instance, in Ritz and D'Andrea (2013) the authors face the problem of cooperative aerial transportation of an elastic object. In Caccavale, Giglio, Muscio, and Pierri (2015) multiple flying arms are exploited for a similar problem. In Baizid, Caccavale, Chiaverini, Giglio, and Pierri (2014); Yang and Lee (2015) the authors face the problem by taking into consideration aerial vehicles with a robotic arm. Notice that in order to reduce the coupling between the transported load and the aerial vehicles attitude dynamics, a solution is to perform aerial manipulation using cables instead of manipulators, see Manubens, Devaurs, Ros, and Cortés (2013); Masone, Bülthoff, and Stegagno (2016); Sreenath and Kumar (2013b). The mentioned examples exploit a centralized control. Differently, a decentralized algorithm Mellinger, Shomin, Michael, and Kumar (2010), is more robust and scalable with respect to the number of robots. Notice that in all these scenarios the robots or do not exchange any information, adopting a distributed approach, or they exploit an ideal, continuous-time communication. In the first case, communication is performed implicitly by the forces exchanged through the load. In Figure 1.5 an example of cooperative system where four ground robots manipulate a common object is depicted: in this case, being the robots rigid structures, force feedback can be exploited to control the load pose by installing force or torque sensors. However, when dealing with aerial manipulation, the presence of manipulators attached to the aerial robots would increase a lot the complexity of the

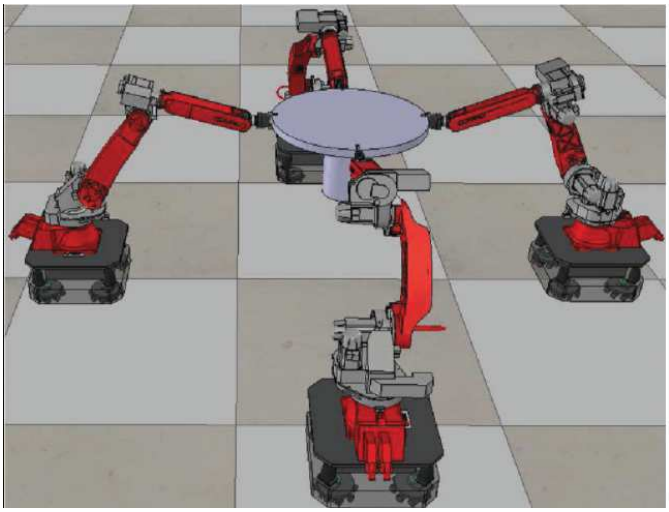


Figure 1.5. Scenario introduced in Marino (2018), where four ground robots cooperatively transport a common load.

overall dynamical model, hence of the task. In this case, force feedback is not sufficient if a precise tracking is required because the system is affected by a lot of external disturbances which cannot be neglected and/or compensated. As a consequence, a pose feedback from the load is needed. A solution to simplify the analysis, in the aerial scenario, is to tether the robots to the load via cables and to exchange information with the common object. In this way, each robot does not need to estimate the force exerted by the other robots on the common load in order to retrieve its pose. It is interesting to notice that even in those cases where force-feedback can be implemented, anyway the force exerted on the common object could break or damage it. Thus a communication-based solution can still be a good alternative.

1.0.1 Contribution

In this work we address the problem of advance control of tethered aerial robots in the context of load transportation and landing.

Firstly, we propose a **two-layers strategy** to make a **tethered quad-rotor** land on an inclined surface: the use of a cable is necessary because of the vehicle under-actuation, that would not allow it to perform such an operation. The two-layer strategy has two main purposes: the higher level is an on-line trajectory generator based on **Nonlinear Model Predictive Control** (NMPC) that generates the desired inputs for the system in order to fulfill the task while satisfying non standard constraints and avoid obstacles and singularities. The low-level controller working at higher frequency allows to handle external disturbances and model uncertainties while tracking the desired trajectory. It is based on-feedback linearization and we show that this multi-layer structure allows to obtain better performance with respect to the use of only NMPC to handle disturbances. Notice that, even though a similar problem has been already faced in Tognon, Testa, Rossi, and Franchi (2016c) where an off-line trajectory is generated to be tracked by the system, in this scenario we aim at proposing a real-time application: indeed, the reference trajectory is computed online and it takes care of eventual external disturbances (like

the wind) and model uncertaintes which affect the task performance:these un-modeled quantities can be handled thanks to the use of the two-layer strategy. Moreover, we impose a set of non-standard and non-linear constraints to be satisfied during the maneuver: this allows to land and take-off smoothly and to avoid the risk of breaking parts of the system.

Then, in the second part of the thesis, we still consider tethered aerial vehicles which are linked to a common object via cables: the goal is to make the whole system reach a desired configuration or track a desired trajectory. To do that, we propose a **multi-layer architecture** that allows to perform the task even in presence of **sampled communication**. Indeed, we are interested in realistic robotics applications where usually communication or is not used at all or is assumed to be ideal, i.e. fast enough to be considered as continuous. A typical example is the case where a group of ground robots which are endowed with robotic arms and use them to transport a payload; in this case they retrieve the payload position and orientation exploiting force feedback. No information is exchanged among the robots in order to increase the flexibility of the system: on the other hand, if the object is delicate, then it could break because of the external forces exerted by the arms. Moreover, if a precise value of position and orientation are required, then force feedback is not enough. A possible solution is the use of communication between each robot and the object in order to know exactly the payload position and orientation. In this case, it is possible to tether a group of aerial robots to the payload via some cables which are attached through passive joints: this allows to avoid the use of grippers and robotic arms. Notice that the use of cables induces the decoupling between the rotational dynamics of the vehicles and the one of the load. Full pose control of the load is possible when the robotic system can attain a six-dimensional wrench on the load by controlling the orientation of the cables and the force applied by them. The minimum setup that possesses such ability is a system with three vehicles connected to three non-collinear points (this system is called Fly-crane in the following). Actually, as mentioned before, when dealing with real systems which communicate via wireless, measurements are sent at sampling times and they may be affected by packets losses and asynchronous behavior. As a consequence, to guarantee a low percentage of packets losses, the rate used to transmit information can not be to large. To our knowledge, for the first time we face such a problem and we propose a control strategy to perform the task in these conditions. We also prove theoretical results which guarantee the stability of the system during the maneuver and we show by means of simulations that our strategy works well on a dynamical system. Finally, we provide the results of experiments on a real system.

1.0.2 Structure of the Thesis

In Chapter 2 we describe an on-line Model Predictive Control for tethered Unmanned Aerial Vehicles to perform a safe and constrained maneuver. We propose a multi-layer architecture to generate on-line an optimal trajectory to allow a tethered aerial robot land on an inclined surface. In Chapter 3 we design a local planner in order to make a class of multi-agent systems reach a desired configuration which is assumed to be generated by a global planner, that takes care of singularities and obstacles avoidance.

The proposed strategy is analysed in the context of sampled communication. Chapter 4 considers the previous problem in a more general scenario, where the multi-agent system is over-actuated: in this case we propose two techniques to implement point-stabilization control, as before, such that the controlled system can handle unfeasible reference to be tracked by the system. Finally, in Chapter 5 we propose a solution to make a group of robots track a trajectory when measurements from the system are available at sampling instants.

2

Online Nonlinear Model Predictive Control for tethered UAVs to perform a safe and constrained maneuver

Contents

This section focuses on the design and implementation of a two layer hierarchical controller for a tethered aerial vehicle. In particular, a challenging problem of smoothly landing on a inclined plane is considered. Such problem yields nonlinear control objective and constraints which require advanced control techniques like Nonlinear Model Predictive Control (NMPC), that is known to be computationally demanding. A control architecture composed by two layers is proposed: an NMPC operating at 100Hz is employed to generate feasible trajectories on-line and a simplified local controller, working at 500Hz, is used to track them. The effectiveness of this hierarchical control structure is demonstrated by a closed-loop simulation. Moreover, the rejection of external disturbances, such as the wind, is a peculiar characteristic of the proposed control strategy.

2.1 Introduction

This section is based on the work Rossi, Bruschetta, Carli, Chen, and Farina (2019): in Section 1 we already named some motivations that make tethered UAVs an interesting solution in many scenarios: the presence of the cable guarantees a secure communication, improves the stability of the robot in presence of external disturbances and allows maneuvers which would be otherwise impossible due to the intrinsic under-actuation of vehicles like quad-rotors. The work presented in this section is a further development of Tognon, Testa, Rossi, and Franchi (2016a), where an optimal trajectory is generated to make a tethered UAV (Unmanned Aerial Vehicle) perform safe landing and takeoff maneuvers. A hierarchical nonlinear controller is exploited to track the position and orientation of the vehicle. Nevertheless, the optimal path is generated offline and is not recomputed if the system state changes, hence loosing its optimality properties.
As in Tognon et al. (2016a), we consider the case of landing on a inclined plane, but using a simplified 2D model. In particular, our goal is to develop a two-layers strategy: the first layer is an *on-line* reference generator based on Nonlinear Model Preditive Control (NMPC) that minimizes a nonlinear functional cost under *non standard* constraints and to generate feasible trajectories, that adapt on-line to disturbances or modeling errors; the second layer is a simplified feedback linearization controller that allows the system to track the trajectory generated by the first layer. Since the constraints are

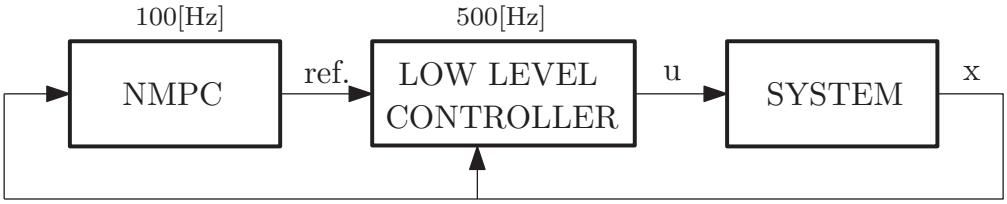


Figure 2.1. Block diagram of the two-layer control architecture

computationally demanding, the NMPC problem cannot be solved at high frequency. This is why the additional low-level controller is used: working at high frequency, it can handle disturbances and model uncertainties while tracking the reference trajectory. In Figure 2.1 the mentioned architecture is depicted. In particular, the first sample of the state sequence generated from the NMPC is interpolated and used as reference for the low level controller, that works at a higher frequency. With this control architecture, since NMPC is updated with state measurements at each time step, then the robots are able to adaptively reconfigure when the environment where they are operating suddenly changes. At the same time, their behavior satisfies a set of highly non linear constraints, which are studied for the particular task that is considered. We show that the proposed control strategy makes the system reject non modeled external disturbances (in this case the wind) better than using only NMPC. Thus, a controlled landing maneuver can be performed, adjusting the orientation and the velocity of the UAV when it approaches the ground. The simulations show that, despite the complexity of such restrictions, it is still possible to reach high performance, by using such a two-layers structure. A fast NMPC implementation is used to meet the real time requirement of this application, exploiting

the so-called Real Time Iteration (RTI) scheme Diehl, Bock, Schlöder, Findeisen, Nagy, and Allgöwer (2002) implemented in the package *MATMPC* Chen, Bruschetta, Picotti, and Beghi (2019); Yutao (2017). As a consequence, techniques that reduce the model or the number of prediction points (Abdolhosseini, Zhang, and Rabbath (2013)) are not required.
We can resume the contributions of this chapter in the following points:

- We derive the 2D dynamical model of a quad-rotor, tethered to the ground through a cable, motivating the choice of such system
- We define an optimization problem to make the robot land on an inclined surface by imposing a set of constraints in order to guarantee that the maneuver is safe and smooth
- Based on the optimization problem, we design a NMPC framework in order to generate on-line the inputs for the system
- We define a two-layer architecture, where the output of the NMPC, a desired trajectory, is given to a low-level local controller: it converts the trajectory into force and torque values for the robot
- We validate our strategy on the dynamical model of the system, by adding external disturbances (the wind) and including model uncertaintes

2.1.1 Overview

The chapter is organized as follows. In Section 2.2 we describe the model of the system that has to be controlled and we formulate the problem we aim at solving. In Section 2.3 we illustrate the on-line reference generator we developed, discussing in detail the optimization problem we solve within the NMPC implementation. In Section 2.5 the low level controller is introduced. In Section 2.6 we report the numerical results obtained and we show that the low-level controller improves the performance when external disturbances affect the system.

2.2 Modeling and problem formulation

Consider an unmanned aerial vehicle (UAV) tethered to the ground through a taut cable with fixed length l (see Figure 2.2). The robot body (B) has mass $m_B \in \mathbb{R}_{>0}$ and inertia $J_B \in \mathbb{R}_{>0}$, whereas the cable mass and inertia are neglected in order to simplify the analysis: indeed, for a non rigid connection, they are usually much smaller with respect to the other components mass and inertia. Let us denote the world frame by $\mathcal{F}_W$, with origin O_W and axes $\{\mathbf{x}_W, \mathbf{y}_W, \mathbf{z}_W\}$ where $\mathbf{z}_W$ is opposite to the gravity vector. Then let us introduce the frame $\mathcal{F}_B$ rigidly attached to the UAV, with axes $\{\mathbf{x}_B, \mathbf{y}_B, \mathbf{z}_B\}$ and origin O_B, set on the center of mass (CoM) of the body. Observe that the axes $\mathbf{y}_B$ and $\mathbf{y}_W$ are parallel to each other and both perpendicular to the plane where the robot moves; in this work, indeed, we assume that the UAV can move only in the 2-dimensional

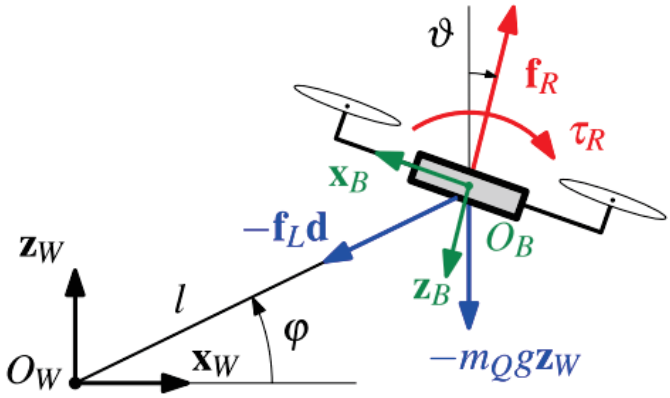


Figure 2.2. Representation of the system and its main variables.

$\mathbf{x}_W, \mathbf{z}_W$ - plane. In particular, the CoM of the UAV is constrained to move on the circle centered at O_W with radius l. Thus, position and orientation can be completely described by the generalized coordinates $\mathbf{q} = (\varphi, \vartheta)$ where φ represents the elevation of the UAV with respect to the ground, while ϑ is the UAV attitude; in particular, the position of the UAV is described on $\mathcal{F}_W$ as $\mathbf{p}_B = [\mathbf{x}_B, \mathbf{y}_B, \mathbf{z}_B]^T = [l\cos\varphi, 0, l\sin\varphi]^T$ while the angular velocity as $\boldsymbol{\omega}_B = [0, \dot{\vartheta}_B, 0]^T$. We assume the UAV is endowed with two propellers, both situated at distance d from the CoM (see Figure 2.2). When rotating these propellers, the forces $\mathbf{f}_1$, $\mathbf{f}_2$ are generated, where $\mathbf{f}_i = b\omega_i^2$, being b a constant depending on physical characteristic and ω_i^2 the angular speed of propeller i. The forces $\mathbf{f}_1$, $\mathbf{f}_2$ actuate the system with the thrust $f_R \in \mathbb{R}$ and torque $\tau_R \in \mathbb{R}$ (both depicted in red in Figure 2.2) such that $\mathbf{f}_R = -f_R \mathbf{z}_B$ and $\boldsymbol{\tau}_R = \tau_R \mathbf{y}_B$. The relation between $\mathbf{f}_R$, $\boldsymbol{\tau}_R$ and $\mathbf{f}_1$, $\mathbf{f}_2$ is uniquely determined by

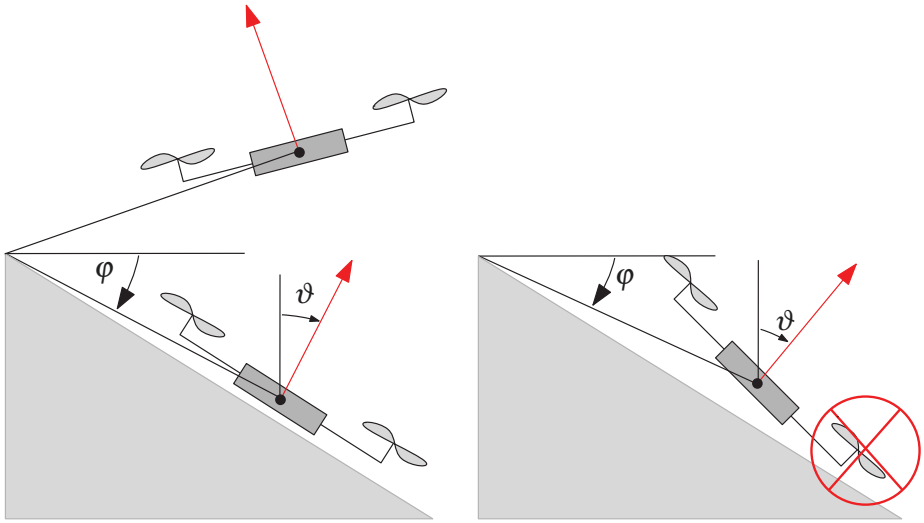
$$\begin{bmatrix} \mathbf{f}_R \\ \boldsymbol{\tau}_R \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ -d & d \end{bmatrix} \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{bmatrix} \tag{2.1}$$

In the following we will assume $\mathbf{f}_R$, τ_R to be the inputs of the UAV. In addition, the extremities of the cable are anchored to a fixed point, O_W, and to O_B that moves on the 2D plane. The dynamic model of the system is derived using the Euler-Lagrangian formulation, computing the kinetic and potential energies K and U, the Lagrangian function $\mathcal{L} = K - U$, the generalized forces Q and solving $\frac{d}{dt}\frac{dL}{d\dot{\mathbf{q}}} - \frac{d\mathcal{L}}{d\mathbf{q}} = Q$, where $\mathbf{q} = [\varphi \ \ \vartheta]^T$; the following model is thus obtained

$$\begin{aligned} m_B l \ddot{\varphi} &= -m_B g \cos\varphi + f_R \cos\varphi + \vartheta \\ J_B \ddot{\vartheta} &= \tau_B. \end{aligned} \tag{2.2}$$

The derivation of (2.2) is explained in Tognon and Franchi (2015). It is convenient to rewrite the model in state-space form. To this aim, let us introduce the state vector $\mathbf{x} = [x_1, x_2, x_3, x_4]^T = [\varphi, \dot{\varphi}, \vartheta, \dot{\vartheta}]^T$ and the input vector $\mathbf{u} = [u_1, u_2]^T = [f_R, \tau_R]^T$; then we have

$$\begin{aligned} &\dot{x}_1 = x_2. \quad \dot{x}_2 = a_1 \cos x_1 + a_2 \cos(x_1 + x_3) u_1 \\ &\dot{x}_3 = x_4, \quad \dot{x}_4 = a_3 u_2 \end{aligned} \tag{2.3}$$



(a) *Correct landing* **(b)** *Wrong landing*

Figure 2.3. Feasible and unfeasible landing

where $a_1 = -g/l \quad a_2 = 1/(m_B l) \quad a_3 = 1/J_B$.
An important quantity in our framework is the stress f_L acting on the cable; it is derived by projecting along the cable all the forces acting on the system, namely, the gravitational and centrifugal forces and the total thrust f_R, leading to

$$f_L = \frac{1}{a_2} x_2^2 + \frac{a_1}{a_2} \sin(x_1) + \sin(x_1 + x_3) f_R \tag{2.4}$$

In order to have a taut cable, f_L must be greater than zero.

2.3 On-line Reference Generator

In this section, we describe the guidelines used to design the optimization problem to be solved into the NMPC framework we consider. This allows to generate on-line and in a closed-loop fashion the trajectory to be followed The design of the on-line reference generator is based on an NMPC formulation, which allows to define complex control objectives and constraints. Before describing the optimization problem, we address those conditions that play an important role in the on-line solution of the control problem. specifically:
Feasibility condition: to guarantee a safe maneuver, the propellers must touch the ground with zero velocity. Hence, the UAV must decelerate and be approximately oriented as the inclined plane while approaching it, in order to avoid situations like the one in Figure 2.3b and prefer instead the scenario shown in Figure 2.3a.
Stress: it is the internal force acting along the cable and must be $f_L \geq 0$, otherwise the cable is not taut and the dynamic model is no more valid. However, an upper-bound is imposed by the internal elasticity properties of the cable.
Thrust, torque: the maximum value ω_{max} attained by the speeds of the propellers ω_i, $i = 1,2$ reflects on the maximum admissible thrust and torque values which actuate the

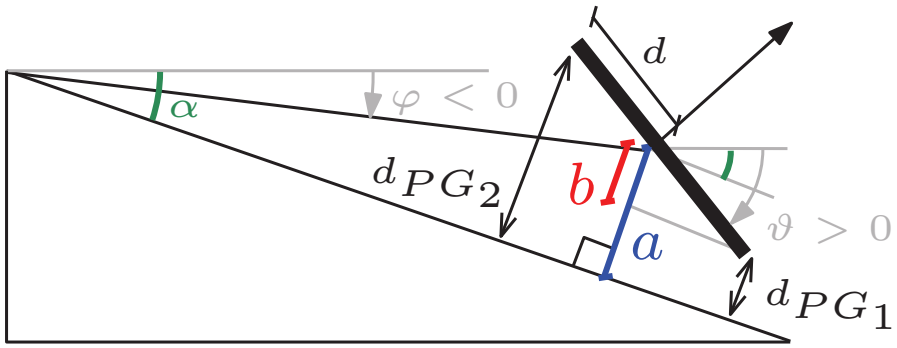


Figure 2.4. The robot is clockwise rotated with respect to the sloped surface with $\vartheta \geq \alpha$; in this case $b \geq 0$ hence both (2.7) are satisfied.

UAV.

Smoothness of the inputs: discontinuities of the inputs or states are not feasible for a real system, hence smoothness has to be enforced. In this regard, it is convenient to modify the dynamic model in (2.3), assuming as new input the time-derivatives of the thrust and of the torque, that is, $\dot{f}_R$, $\dot{\tau}_R$. Introducing the state $\mathbf{x} = [\varphi, \dot{\varphi}, \vartheta, \dot{\vartheta}, f_R, \tau_R]^T$ and the input $\mathbf{u} = [\dot{f}_R, \dot{\tau}_R]^T$, the new dynamics can be written as

$$\begin{aligned} \dot{x}_1 &= x_2, \quad \dot{x}_2 = a_1 \cos x_1 + a_2 \cos(x_1 + x_3) x_5 \\ \dot{x}_3 &= x_4, \quad \dot{x}_4 = a_3 x_6, \quad \dot{x}_5 = u_1, \quad \dot{x}_6 = u_2 \end{aligned} \tag{2.5}$$

Next section formally describes the introduced constraints.

2.3.1 Constraints

We start by formulating the constraints associated to the physical limits of the system, that is,

$$0 \leq f_L \leq f_{L_{max}}, \quad 0 \leq f_R \leq f_{R_{max}}, \quad |\tau_R| \leq \tau_{R_{max}} \tag{2.6}$$

where $f_{L_{max}}$ is the maximum stress that can be exerted on the cable, and $f_{R_{max}}$, $\tau_{R_{max}}$ are, respectively, the maximum thrust and torque that can be generated by the propellers. Observe that, in real scenarios, when landing, the propellers can not touch the ground before the legs of the UAV. In our simplified model we impose that they can not reach the inclined surface before the system CoM. Let α be the slope of the inclined plane. Then, from geometrical considerations, we want that the distance between each propeller and the ground (d_{PG_1}, d_{PG_2}) is always greater than zero:

$$d_{PG_1} := a - b \geq 0, \qquad d_{PG_2} := a + b \geq 0 \tag{2.7}$$

where $a = l \sin(\varphi + \alpha) \geq 0$ and $b = d \sin(\vartheta - \alpha)$. The meaning of (2.7) when $\vartheta \geq \alpha$ is represented in Figure 2.4.

To avoid feasibility issues, such constraints are not suitable to be used as hard constraints in the NMPC formulation. The formulation with soft constraints is then introduced

adding the slack variables s_1 and s_2 and (2.7) become

$$a-b\geq s_1,\quad a+b\geq s_2,\quad s_1\geq 0,\quad s_2\geq 0 \tag{2.8}$$

In our problem, we consider constraints in (2.8) as soft constraints and we incorporate them into the functional cost. In Tab.2.1 the list of constraints and their values are reported.

	min	max
$f_R\,[N]$	0	15
$\tau_R\,[N\cdot m]$	-1.2	1.2

	min	max
$f_L\,[N]$	0	10
$s_{1,2}\,[m]$	0	∞

Table 2.1. The bounding values of the thrust, the torque and the stress. The slack variables must be greater than zero in order to properly formulate soft constraints.

2.3.2 Functional cost

In this subsection we focus on the functional cost to be minimized: it includes different contributions and formally is described as

$$J=\int_0^T\sum_{j=1}^{9}\left\|\eta^j\right\|_{w_j}^2 d\tau \tag{2.9}$$

where

$$\begin{array}{ll}\eta^1=\dot{f}_R & \eta^2=\dot{\tau}_R\\ \eta^3=h(\varphi)\,(\vartheta-\vartheta^*) & \eta^4=h(\varphi)\,\dot{\varphi}\\ \eta^5=h(\varphi)\,\dot{\vartheta} & \eta^6=s_1,\,\eta^7=s_2\\ \eta_8=\varphi-\varphi^* & \eta_9=\vartheta-\vartheta^*.\end{array} \tag{2.10}$$

and w_j are the weights, respectively labeled as $k_{f_R},k_{\tau_R},k_\vartheta,k_{\dot{\varphi}},k_{\dot{\vartheta}},k_s,k_{\Delta_\varphi},k_{\Delta_\vartheta}$. Moreover, $\vartheta^*=\alpha$ and $\varphi^*=-\alpha$ denote the desired attitude and elevation. Observe that η^1 and η^2 penalize the rate of variation of the thrust f_R and of the torque τ_R. The terms η^3 , η^4 and η^5 force the system to slow down when approaching the inclined surface and become parallel to it before landing. Notice that the contributions $(\vartheta-\vartheta^*)$, $\dot{\varphi}$ and $\dot{\vartheta}$ in η^3, η^4 and η^5, are weighted by the sigmoid function

$$h(\varphi)=\frac{1}{1+e^{\gamma(\varphi-\varphi_{lim})}} \tag{2.11}$$

where γ and φ_{lim} are two parameters to be tuned; in particular γ determines the smoothness of the function, while φ_{lim} is such that $h(\varphi_{lim})=\frac{1}{2}$. Notice that, since $\varphi^*=-\alpha$, then the closer the UAV to the final configuration ($\varphi<\varphi_{lim}$), the larger is $h(\varphi)$ and, in turn, the greater the values of η^3, η^4 and η^5 in the functional cost. Figure 2.5 shows an example of a sigmoid function. The terms η^6 and η^7 are associated to the slack variables, introduced to generate the relaxed constraints. Note that they are introduced in (2.10) in a quadratic form instead of the more common linear form: this allows an easier tuning procedure for the relative weights and a smoother function for the optimizer.

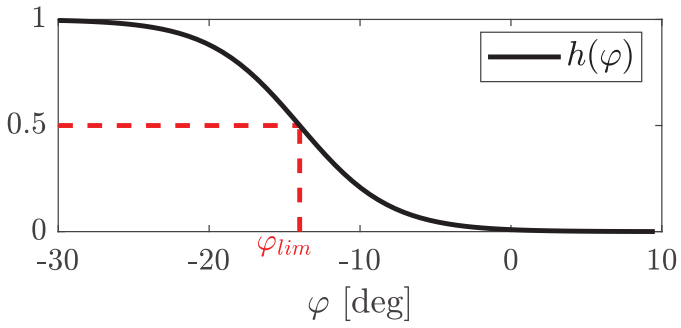


Figure 2.5. Representation of the sigmoid function with $\gamma = 19$ and $\varphi_{lim} = -0.24[rad]$: observe that for $\varphi < \varphi_{lim}$ $h(\varphi) \to 1$.

Finally, the terms η^8 and η^9 are used to set the desired final elevation and attitude for the UAV: to this aim, $k_{\Delta_\varphi}, k_{\Delta_\vartheta}$ weight the distance between the current configuration and the desired one.

Based on the above discussion, we need to solve within the NMPC framework the following optimization problem

$$\begin{aligned} J_t^* = & \min_{x(\,\cdot\,),u(\,\cdot\,),p,T} J[x(\,\cdot\,),u(\,\cdot\,),s(\,\cdot\,),p,T] \\ & \text{subject to:} \\ & \forall t \in [t_0,T] : f(t,\dot{x}(t),x(t),u(t),p,T) = 0 \\ & \qquad\qquad r(t,x(t),u(t),s(t),p,T) \leq 0 \end{aligned} \tag{2.12}$$

where p is a vector of parameters, T the total duration, x the state, u the input, s the slack variables and J the functional cost introduced in this subsection. Moreover the equality constraints refers to the dynamics of the system in (2.5), while the inequality constraints account for the constraints discussed in subsection 2.3.1.

2.4 NMPC for Reference Generation

To generate a feasible trajectory for the low level controller, problem (2.12) should be solved on-line in real-time. Firstly, (2.12) is discretized using multiple shooting Bock and Plitt (1984) yielding the following nonlinear programming problem (NLP):

$$\begin{aligned} \min_{\mathbf{x},\mathbf{u}} \quad & \frac{1}{2}\sum_{k=0}^{N-1} \|\eta_k(x_k,u_k)\|_W^2 + \frac{1}{2}\|\eta_N(x_N)\|_{W_N}^2 \\ s.t. \quad & x_0 = \hat{x}_0, \\ & x_{k+1} = F(x_k,u_k),\ k = 0,1,\ldots,N-1 \\ & r_k(x_k,u_k) \leq 0,\ k = 0,1,\ldots,N-1, \\ & r_N(x_N) \leq 0, \end{aligned} \tag{2.13}$$

where $\mathbf{x} = [x_0^\top, x_1^\top, \ldots, x_N^\top]^\top$ and $\mathbf{u} = [u_0^\top, u_1^\top, \ldots, u_{N-1}^\top]^\top$ are state and control vectors defined in N shooting nodes and subintervals over the prediction horizon, respectively.

The cost function has the form of $\eta_k = [\eta_k^1, \ldots, \eta_k^7]^\top$ and the corresponding weights are set in the matrix W . The state x_0 is enforced to be equal to the current state measurement $\hat{x}_0$, essentialy leading to a closed-loop scheme. The equality constraint represents the coupling relation between states from two shooting nodes and system dynamics are discretized using an integrator, denoted by F. To reduce the time for solving (2.13), the Real-Time Iteration (RTI) scheme Diehl et al. (2002) is employed. RTI scheme performs a single Sequential Quadratic Programming (SQP) iteration to solve (2.13). Notice that if the sampling time used for discretization is too large, then undesired behaviors may arise between two consecutive samples. Thus, the problem can be neglected by choosing the sampling time small enough (we chose $T_s = 0.01\,[s]$). Moreover, if a complete SQP is exploited to solve the NLP then the constraints are satisfied; on the other hand, when using the RTI scheme, the problem can still be neglected only if the KKT conditions, which are the derivatives of the Lagrangian gradient, are smaller than the chosen threshold.

In this work, instead of applying $\mathbf{u}^*$ to control the quadrotor, the state vector at the first shooting node is used to generate a reference for the low level controller for a period of 10 ms in the future. An interpolator that ensures $\mathcal{C}^2$ condition is implemented to handle the sampling rate transition between NMPC (working at 100Hz) and low level controller (500Hz) and to enforce smoothness on the generated trajectory.

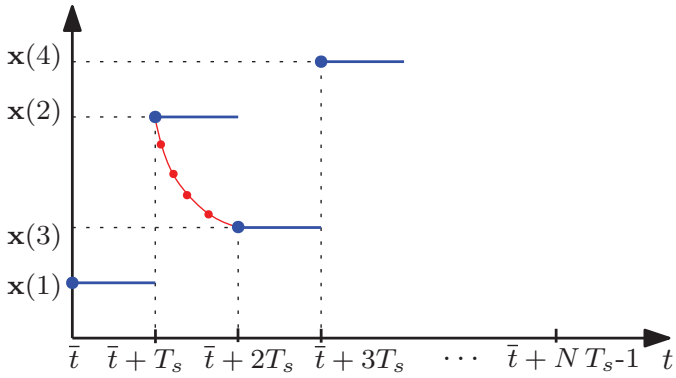


Figure 2.6. Interpolation: at time $\bar{t}$ a sequence of N reference states is generated by solving (2.13); usually the first output is applied and, at $\bar{t} + T_s$, where $T_s = 10\,[ms]$ is the shooting time, a new sequence is computed. In this scenario, instead of sending to the low-level controller a constant reference for T_s seconds, we send the interpolation with the next state in order to anticipate the system behavior.

2.4.1 Implementation Details

The NMPC trajectory generator is implemented in C code and compiled in MATLAB using the software package MATMPC Chen, Bruschetta, Cuccato, and Beghi (2018); Chen et al. (2019); Chen, Cuccato, Bruschetta, and Beghi (2017), in order to be used together with the local controller in Simulink. A fixed step 4^{th} order explicit Runge-Kutta integrator is used to discretize system dynamics over a prediction horizon $T = 0.3$s with $N = 30$ shooting nodes. The resulting dense QP is solved by qpOASES Ferreau, Kirches, Potschka, Bock, and Diehl (2014), which employs on-line active-set method with warm-start strategy.

2.5 Low Level Local Controller Design

The output of the NMPC is a desired trajectory, defined in position and velocity, while the desired acceleration is obtained taking the derivative of the interpolated velocity. So $\mathbf{x}^{(d)}$, $\dot{\mathbf{x}}^{(d)}$, $\ddot{\mathbf{x}}^{(d)}$, are the references to be tracked by system in (2.3). This task is accomplished by a low level controller that we design resorting to a feedback linearization approach. Let us introduce the auxiliary control variables $\bar{u}_1$ and $\bar{u}_2$ defined as

$$\begin{aligned} \bar{u}_1 &:= a_1 \cos x_1 + a_2 \cos(x_1 + x_3) u_1, \\ \bar{u}_2 &:= a_3 u_2 \end{aligned} \tag{2.14}$$

Based on the variables, the dynamics in (2.3) is rewritten as

$$\begin{aligned} \ddot{x}_1 &= \bar{u}_1, \\ \ddot{x}_3 &= \bar{u}_2 \end{aligned} \tag{2.15}$$

namely, it reduces to a two-dimensional system of decoupled double integrators. System in (2.15) can be stabilized by employing a standard PID controller with a feedforward term, e.g.,

$$\begin{aligned} \bar{u}_1 &= \ddot{x}_1^{(d)} - K_P e_1 - K_D \dot{e}_1 - K_I \int e_1(\tau) d\tau \\ \bar{u}_2 &= \ddot{x}_2^{(d)} - K_P e_2 - K_D \dot{e}_2 - K_I \int e_2(\tau) d\tau \end{aligned}$$

where $e_1 := x_1 - x_1^{(d)}$ and $e_2 := x_2 - x_2^{(d)}$ are the position errors.

The values of u_1 and u_2 can be obtained from $\bar{u}_1$ and $\bar{u}_2$ by inverting relations in (2.14), i.e.,

$$\begin{aligned} u_1 &= (\bar{u}_1 - a_1 \cos x_1)/(a_2 \cos(x_1 + x_3)), \\ u_2 &= \bar{u}_2 / a_3 \end{aligned} \tag{2.16}$$

Observe that in the first equation $\cos(x_1 + x_3)$ must be different from zero, i.e., $\vartheta + \varphi \neq \pi/2$. In the considered scenario we are far from this critical condition, that, however, might be taken into account adding a further constraint.

2.6 Simulation

The performance of the proposed control strategy are evaluated simulating the reference generator, the controller and the process within a MatLab-Simulink environment. The controller architecture is as presented in Figure 2.1, i.e a series of an NMPC controller used as reference generator working at 100Hz and a feedback-linearized low level controller working at 500Hz with feed forward action. This high frequency is motivated by an

eventual real-time implementation. The NMPC controller is tuned to exploit at best the propellers, emulating a minimum time maneuver, hence heavily stressing the control performance close to the ground. In Tab. 2.2 the cost function weights are reported. As

k_{f_R}, k_{τ_R}	k_ϑ	$k_{\dot{\varphi}}, k_{\dot{\vartheta}}$	k_s	$k_{\Delta_\varphi}, k_{\Delta_\vartheta}$
10^{-4}	1	1	10^6	$2 \cdot 10^2$

Table 2.2. List of parameters and their values. k_{f_R}, k_{τ_R}, k_ϑ, $k_{\dot{\varphi}}$, $k_{\dot{\vartheta}}$, k_s, k_{Δ_φ}, k_{Δ_ϑ} are the cost function weights.

far as the low level controller is concerned, the gains are tuned manually.

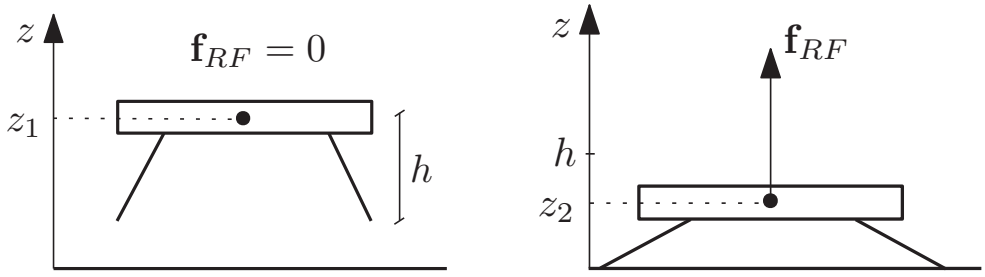
2.6.1 Model for simulation

To obtain more realistic results, the simulations are performed adopting two different models: one for control, i.e. (2.5), and one for simulation. In particular, the simulation model is obtained extending the model described in (2.5) by including the following non-idealities.

Firstly, the robot is assumed to have two **landers**, which can be compressed when they touch the ground. Their effect is obtained by introducing a reaction force, modelled as a dumped elastic force, that counteracts the one exerted by the robot on the ground (see Figure 2.7):

$$\mathbf{f}_{RF} = \begin{cases} k_{el}(h - z) - \xi\dot{\varphi} & \text{if } z \leq h \\ 0 & \text{otherwise} \end{cases} \tag{2.17}$$

where $k_{el} \geq 0$ is the elastic constant, h the landers height, z the distance between the robot CoM and the ground and $\xi \geq 0$ is the dumping factor that multiplies the robot velocity. The second non-ideality is an external **disturbance**: the wind model described



(a) *$z_1 > h$: the landers do not touch the ground.*

(b) *$z_2 \leq h$: the landers are compressed.*

Figure 2.7. The reaction force (see (2.17)) is active if $z \leq h$.

in Sandino et al. (2014) is adopted on the longitudinal direction ($\mathbf{x}_W$). A pulse force at $t = 0\,[s]$ and a sinusoidal force starting at $t = 0.1\,[s]$ are applied at the robot CoM, making the dynamics of θ independent of this disturbance.

In Section 2.6.2, we show how the two-layer strategy guarantees high performance even if the system used in simulation is different from the one used to generate the optimal trajectory.

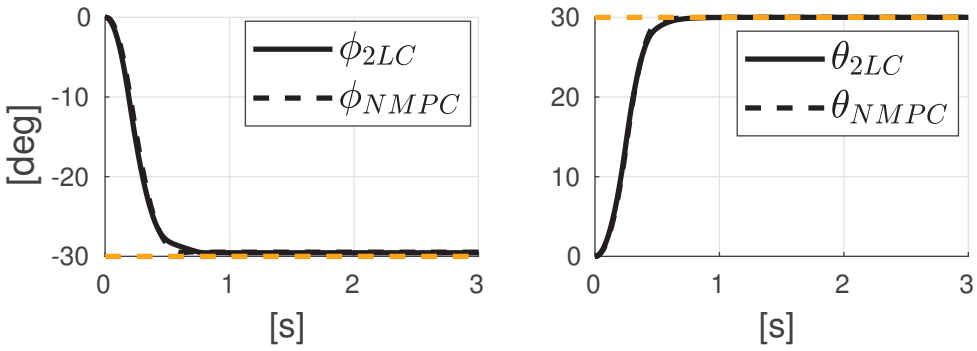


Figure 2.8. Attitude and elevation angles using only NMPC or the combination of NMPC and the low level controller (2LC). The yellow line represents the desired value. Observe that the attitude is already almost parallel to the ground, before the landing is completed.

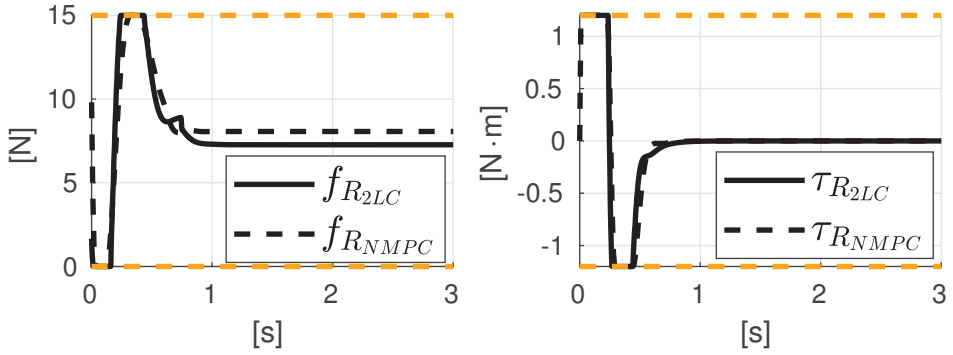


Figure 2.9. Inputs using only NMPC or the combination of NMPC and the low level controller (2LC). The yellow line represents the lower and upper bounds of the thrust and torque. Input saturation emphasizes the fact that the input forces are exploited at best.

2.6.2 Simulation results

The simulation is performed assuming that the initial conditions are set at 0 degrees both for ϑ and φ, whereas the desired values are set to $\vartheta^* = 30° = -\varphi^*$. In all the figures, two cases are compared: the dashed lines refer to the signals generated by using only the NMPC, while the continuous lines describe the signal obtained by the combination of the NMPC and the feedback linearization, that is by the two-layers control strategy (the related subscript in the plots is $2LC$). Firstly, the simulation is performed without the wind action. In Figures 2.8 and 2.9 the attitude and the elevation angles and their rates are reported together with τ_R and f_R. Available input forces are exploited at best, reaching maximum velocities greater than 100 deg/s and moving the UAV from the initial position to the final one in almost 0.7 seconds. Figure 2.9 shows that 2LC requires smaller inputs values and the reason will be more clear in the following lines.

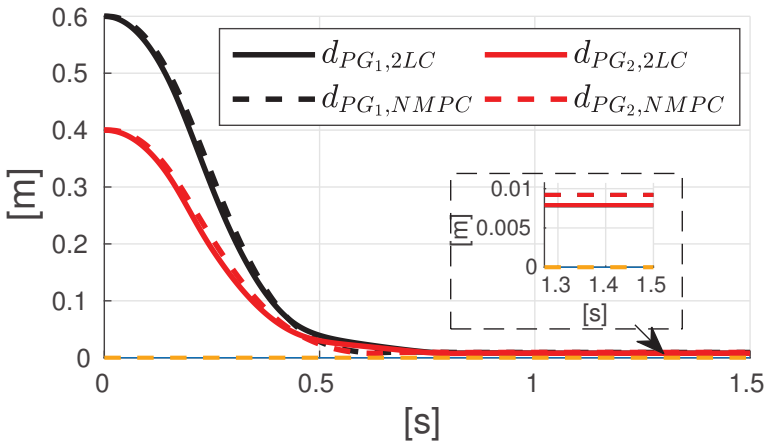


Figure 2.10. Feasibility constraints: note that the distance between each propeller and the ground ($d_{PG_{1,2}}$) is always greater than 0 (highlighted by the yellow line), that is (2.7) are satisfied (the propellers never intersect the ground).

The low level controller performance makes the actual behaviour almost over-imposed to the desired one, although no information on the constraints is used within it. The behavior (of NMPC and 2LC) is not identical because, for a matter of implementation feasibility, the multi-layer controller performs an interpolation of the trajectory generated by the NMPC. As far as the constraints (2.7) are concerned, they are perfectly fulfilled guaranteeing a smooth landing, as can be seen in Figure 2.10. In addition, note that at

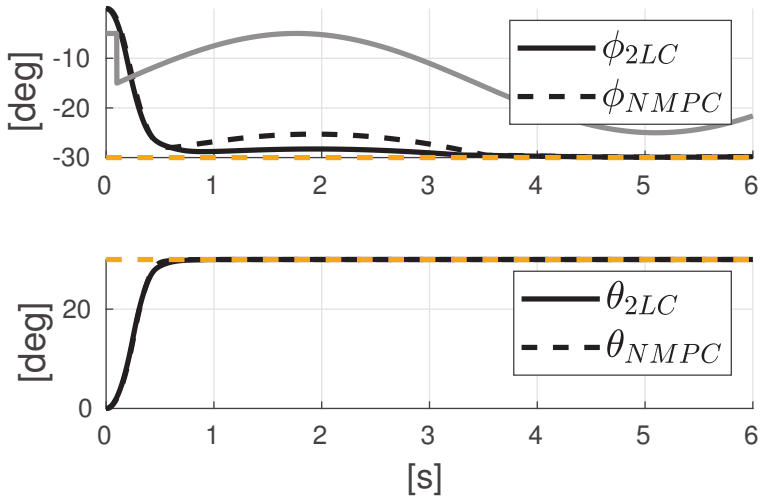


Figure 2.11. Comparison between elevation and attitude angles in presence of wind, described by the grey line. The yellow line instead represents the desired value. Note that the disturbance rejection is better tackled by the 2LC technique with respect to NMPC.

steady-state the propellers do not touch the ground (see the zoomed plot): this is due to the landers which support the robot. Furthermore, the aggressive PID action in the 2LC case, reduces the tracking error with respect to NMPC, bringing both the propellers closer to the ground (thus even the CoM). Hence the thrust needed in this case is smaller at the end of the simulation time, as shown in Figure 2.9. To conclude, from simulations,

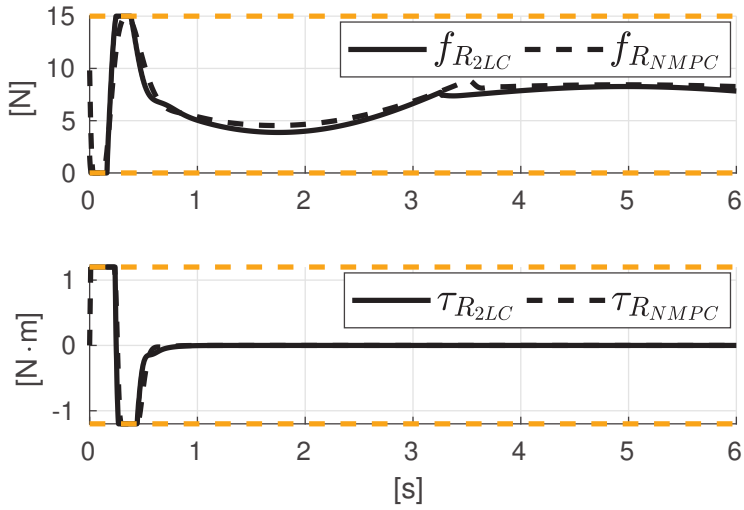


Figure 2.12. Inputs values under the effect of the wind: smaller values are required if the 2LC strategy is used. The yellow line represents the lower and upper bounds of the thrust and torque

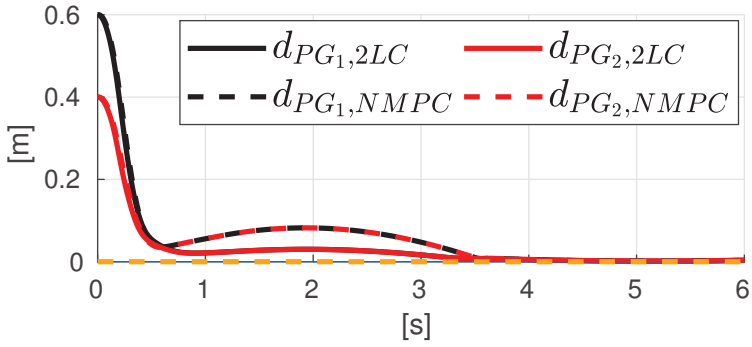


Figure 2.13. Feasibility constraints under the wind effect: the tracking error is greater when using only NMPC instead of the 2LC strategy.

we observed that the constraint on the stress $0 \leq f_L \leq f_{L_{max}}$ is respected during the whole simulation, maintaining the system structure and model validity, being f_L always far from the limits. Moreover, we observed that the maximal on-line computational time of NMPC per sampling instant is less than 5 milliseconds. This demonstrates that the NMPC trajectory generator can comfortably work at 100Hz even if constraints are active, allowing for possible improvements in the model, or in the constraints or in the operating frequency. As anticipated before, to emphasize the benefits of the 2LC strategy under external disturbances, the effect of a sinusoidal wind is studied in an other simulation. Figure 2.11 shows how the low-level controller has a crucial role in the disturbance rejection: it, indeed, significantly reduces the tracking error with respect to the only NMPC case. Observing Figure 2.12, the thrust generated by NMPC is slightly greater than the one generated in the 2LC case. In the first case, indeed, the robot tries to compensate the wind effect, that brings φ_{NMPC} away from the reference. As far as the stress constraint is concerned, even if not reported, it is fulfilled along all the trajectory, with $||\mathbf{f}_L|| \leq 6\,[\mathrm{N}]$. Finally, Figure 2.13 shows that the constraints in (2.8) are satisfied as well.

3

Cooperative Aerial Load Transportation via Sampled Communication: square systems

Contents

In this section, we propose a feedback-based motion planner for a class of multi-agent systems with a sparse kinematics structure. In other words, the agents are coupled together only by the transported object. The goal is to steer the load into a desired configuration. We suppose that a global motion planner generates a sequence of desired configurations that satisfy constraints as obstacles and singularities avoidance. Then, a local planner receives these references and generates the desired agents velocities, which are converted into force inputs for the vehicles. We focus on the local planner design both in the case of continuously available measurements and when they are transmitted to the agents via sampled communication. For the latter problem, we propose two strategies. The first is the discretization of the continuous-time strategy that preserves stability and guarantees exponential convergence regardless of the sampling period. In this case, the planner gain is static and computed off-line. The second strategy requires to collect the measurements from all sensors and to solve online a set of differential equations at each sampling period. However, it has the advantage to provide doubly exponential convergence. Numerical simulations of these strategies are provided for the cooperative aerial manipulation of a cable-suspended load.

3.1 Introduction

The class of tethered aerial vehicles has a large field of applications, like search and rescue, load manipulation and transportation Maza et al. (2010a); Michael et al. (2011). Due to the limited payload of commercially available vehicles, a group of robots is rather employed to transport and manipulate payloads, e.g., for construction and assembly tasks. The most direct approach to control such a multi-robot system is by a centralized kinematic/dynamic inversion Manubens et al. (2013); Masone et al. (2016); Sanalitro, Savino, Tognon, Cortés, and Franchi (2020); Sreenath and Kumar (2013a). However, a distributed approach, where the communication among agents is restricted to neighbors Farivarnejad and Berman (2018a); Wang and Schwager (2016b) or not employed at all Lim, Kwon, Kim, and Ahn (2017); Tognon, Gabellieri, Pallottino, and Franchi (2018), is more favorable. This technique guarantees major robustness and flexibility with respect to a centralized one, where a central unit handles all the computational load and if the latter fails, then the whole task fails. However, the distributed approach lacks global information as the load state and parameters, or the total number of robots. This aspect increases the difficulty of the controller design and might degrade the performance as well. Examples of distributed control methods can be found for groups of ground Franchi, Petitti, and Rizzo (2019); Petitti, Franchi, Di Paola, and Rizzo (2016), underwater Conti, Meli, Ridolfi, and Allotta (2015) and aerial robots Mellinger, Shomin, Michael, and Kumar (2013). Other examples of decentralized approaches in which ground or aerial robots employ robotic manipulators can be found in Sieber and Hirche (2019) and Verginis, Nikou, and Dimarogonas (2018). To reduce communication issues, communication-less approaches relying on a leader-follower paradigm were presented for the problem of cooperative transportation and manipulation Farivarnejad and Berman (2018b); Gabellieri, Tognon, Sanalitro, Palottino, and Franchi (2020); Tagliabue, Kamel, Siegwart, and Nieto (2019); Tsiamis, Verginis, Bechlioulis, and Kyriakopoulos (2015); Wang and Schwager (2016a). Notice that if the leader fails, then the task will fail too, and if one follower fails, the formation might be loosen. In this scenario, the communication is done implicitly by the forces exchanged through the load Marino and Pierri (2018); Tagliabue, Kamel, Siegwart, and Nieto (2017); Tsiamis, Verginis, Bechlioulis, and Kyriakopoulos (2015). However, if a very precise tracking is required, the force feedback is not enough and a pose feedback from the load is needed. This can be measured by a sensor on the load, or retrieved from the robots pose if those are rigidly connected to the load. In detail, the load sensors send the estimated load pose to the robots via a communication channel. Notice that this reasoning can be extended to formation control problems where a group of robots must complete a desired task Franchi and Giordano (2018); Franchi, Masone, Grabe, Ryll, Bülthoff, and Giordano (2012). Also in this case, a communication-based approach allows the robots to exchange measurements about the formation of the group, such as relative distances. In the scenarios where communication among robots is required to meet the task specifications, problems related to it must be properly assessed. In particular, the limited bandwidth of wireless communication channels makes the assumption of continuous signals not valid. Specific strategies to deal with sampled communication and measurements need to be designed.

To the best of our knowledge, in this work we investigate for the first time the problem of cooperative aerial load transportation via sampled communication. We tackle the problem by considering an architecture where a *global motion planner* generates off-line a sequence of reference points, taking into account obstacles and singularities avoidance. These points are then loaded in the computational unit, that can be placed on the transported platform or on the vehicles, depending on the strategy. Then, to connect each pair of consecutive points along the path computed off-line, we propose a method that computes online the reference vehicles velocities, guaranteeing the stability and the exponential convergence of the pose load error, even when the communication is not continuous. In the following, we call this method a *local motion planner*. The design of this planner is our main contribution: we firstly define it in the continuous time scenario and later extend the analysis to the more realistic case in which the measured variables are transmitted via sampled communication to the low-level agent controllers. We focus only on the kinematic model since it is assumed that each agent is endowed with a *dynamic model-based local controller*. For this problem, we propose two strategies: a static feedback technique that exploits the sparsity of the system model and an adaptive one, where the planner gain is adapted online. The second strategy provides faster convergence rate with respect to the first one, but cannot be implemented in a distributed way.
We can resume the contributions of this chapter in the following points:

- We define a class of multi-agent systems characterized by a sparse kinematic structure: in particular, we consider models which are characterized by a square Jacobian matrix

- We propose a two-layer architecture where a global planner generates the sequence of desired configurations for the system; we design the local planner that, working at low frequency, transforms these references into desired velocities for the agents.

- Firstly, we consider the continuous-time scenario, where measurements from the system are continuously available: based on Lyapunov theory, we give necessary conditions on the planner gain in order to obtain stability and exponential convergence guarantees

- Then, we extend the analysis to the case where measurements are available at sampling instants: this is meant to simulate a wireless transmission. Even in this case, we provide necessary conditions for stability and convergence, by proposing two different strategies to compute the planner gain, depending on the implementation that can be centralized or distributed

- We validate our contribution by means of simulations on the kinematic model of a system with three aerial robots transporting a load via cables

- Finally, we also provide the results obtained by applying our strategies to the real system.

3.1.1 Overview

The chapter is organized as follows: in Section 3.2 we formalize the problem for a specific class of systems. In Section 3.2.2 we show two examples of multi-robot systems which are correctly described by this class. In Section 3.2.3 we introduce the problem to be solved. In Section 3.3 we analyze the continuous-time scenario, extended to the sampled case in Section 3.4. In Section 3.5 numerical simulations are described. Finally, in Section 3.6 we also show the results obtained by applying our strategy to a real robotic system.

3.2 Modeling and problem formulation

3.2.1 Kinematics of multi-agent systems

In this section we describe the kinematic model of a multi-agent system composed by N agents which cooperate to fulfill a task; the agents do not communicate among themselves but receive information from a central unit, that we call *pivot*, and we indicate it with the letter V. Notice that it can be an object to be manipulated (in this case we assume that passive, not extendable connections are used, such as cables, bars, joints, etc.), a robot to be escorted or an other object in the space. Instead of robots, one could use 1-DoF actuated cables of variable length or even 3-DoF actuated points flying in 3D space.

The kinematic model is derived as shown in Appendix B.1.1: briefly, we defined a fixed reference frame $\mathcal{F}_O = O - \{\mathbf{x}_O, \mathbf{y}_O, \mathbf{z}_O\}$ and a moving reference frame $\mathcal{F}_V = V - \{\mathbf{x}_V, \mathbf{y}_V, \mathbf{z}_V\}$ fixed on the center of mass of the pivot. Now, we define the position $\mathbf{p}_V^O$ and orientation (described by a rotation matrix) R_V^O of the pivot with respect to the fixed reference frame and the position $\mathbf{p}_i^V$ and orientation R_i^V of each agent with respect to the pivot. Thus, it is possible to express the position $\mathbf{p}_i^O$ and orientation R_i^O of each agent with respect to the fixed reference frame. As a consequence:

$$\mathbf{p}_i^O = \mathbf{p}_V^O + R_V^O \mathbf{p}_i^V$$

that brings directly to the kinematic model of the system:

$$\begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_N \end{bmatrix} = \begin{bmatrix} \mathbf{h}^{(1)}(\mathbf{q}_1, \mathbf{q}_V) \\ \vdots \\ \mathbf{h}^{(N)}(\mathbf{q}_N, \mathbf{q}_V) \end{bmatrix} \tag{3.1}$$

This function maps the Lagrangian coordinates of the system $\mathbf{q} = [\mathbf{q}_1^\top \cdots \mathbf{q}_N^\top \ \mathbf{q}_V^\top]^\top \in \mathbb{R}^{m \times 1}, m > 0$ to the vector collecting the agents configurations $\mathbf{p} = \begin{bmatrix} \mathbf{p}_1^\top & \dots & \mathbf{p}_N^\top \end{bmatrix}^\top \in \mathbb{R}^{n \times 1}, n > 0$ where $\mathbf{p}_i \in \mathbb{R}^{n_i \times 1}$ represents the cables lengths in a cable-driven robot or the positions of the flying points in an aerial system; in case the cables are attached to carrying robots, then this vector represents the position of the i-th robot in the space. The variables $\mathbf{q}_i \in \mathbb{R}^{m_i \times 1}, m_i > 0$ instead gathers the angles and/or distances between

the pivot V and the i-th agent, and $\mathbf{q}_V \in \mathbb{R}^{m_V \times 1}$, $m_V > 0$ represents the pose (position and orientation) of the pivot itself. Notice that $m = \sum_{i=1}^{N} m_i + m_V$ and $n = \sum_{i=1}^{N} n_i$.

Moreover, note that $m_i = 0$ if $\mathbf{p}_i$ can be fully described by the only pivot pose $\mathbf{q}_V$. Otherwise, if $\mathbf{p}_i$ depends also on $\mathbf{q}_i$, then $m_i = 1$ if one angle is sufficient to express the positions $\mathbf{p}_i$ with respect to the pivot pose like for example in Figure 3.1a; in that scenario, if a single cable connects a robot to the pivot instead of two, then the cable orientation is described by two angles, i.e. $m_i = 2$, see Figure 3.1b. Note that we do not consider rotations of the cables about their own axis, hence $m_i \leq 2$.

A key feature of the systems considered in this work is that they exhibit a *star-like interaction topology* where each agent is connected to a central unit (the pivot). This topology is more evident when considering its differential kinematics:

$$\dot{\mathbf{p}} = \mathbf{A}_{\mathbf{q}}\, \dot{\mathbf{q}}, \tag{3.2}$$

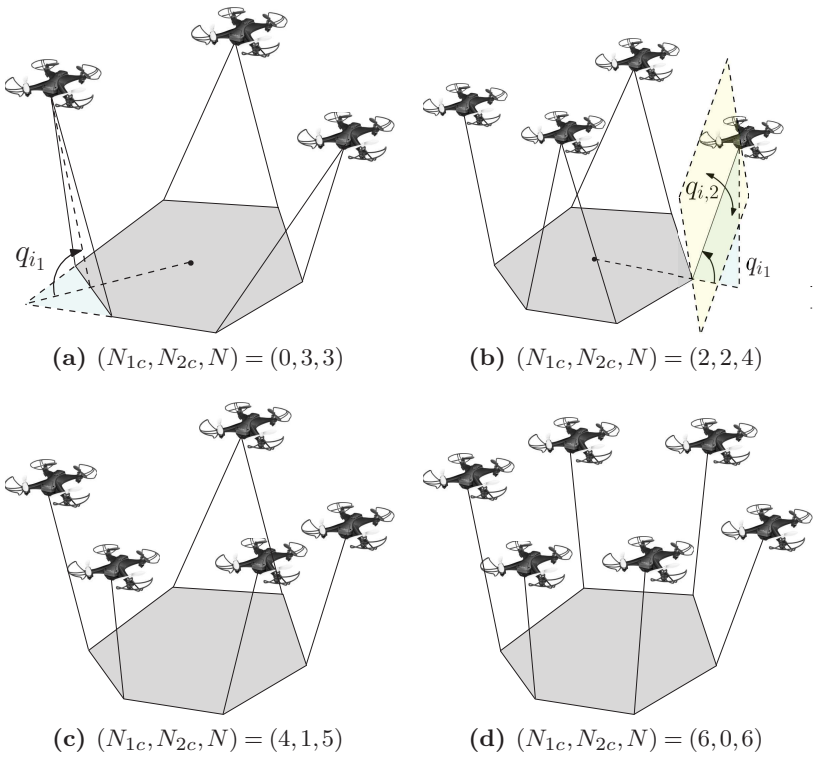
where the Jacobian $\mathbf{A}_{\mathbf{q}} = \frac{\partial \mathbf{h}(\mathbf{q})}{\partial \mathbf{q}} \in \mathbb{R}^{n \times m}$ has the structure

$$\mathbf{A}_{\mathbf{q}} = \left[\begin{array}{ccc|c} \mathbf{A}_{\mathbf{q}_1}^{(1)} & & \mathbf{0} & \mathbf{A}_{\mathbf{q}_V}^{(1)} \\ & \ddots & & \vdots \\ \mathbf{0} & & \mathbf{A}_{\mathbf{q}_N}^{(N)} & \mathbf{A}_{\mathbf{q}_V}^{(N)} \end{array}\right], \tag{3.3}$$

and $\mathbf{A}_{\mathbf{q}_i}^{(i)} = \frac{\partial \mathbf{h}^i(\mathbf{q})}{\partial \mathbf{q}_i} \in \mathbb{R}^{n_i \times m_i}$ and $\mathbf{A}_{\mathbf{q}_V}^{(i)} = \frac{\partial \mathbf{h}^i(\mathbf{q})}{\partial \mathbf{q}_V} \in \mathbb{R}^{n_i \times m_V}$.
Notice that the Jacobian has zero elements outside the main diagonal because of the property of the function $\mathbf{h}(\mathbf{q})$: indeed, the position of each agent depends only on its configuration and on the pivot one, hence it does not depend on the other agents configurations.

Notice that in this section we will focus on the case $n = m$, that is on *square systems*. This choice is motivated by the fact that the cases $n \neq m$ requires a dedicated analysis that is performed in Section 4.1. Indeed, it is not immediate to extend the following results to the more generic scenario of *non square systems*. Let us make an example to justify this: consider a multi-robot system like the one in Figure 3.1a were the robots are linked to a common platform via tight cables; assume that $\mathbf{A}_{\mathbf{q}}$ is invertible and that the input to the system is the vector of desired robots velocities $\dot{\mathbf{p}}^d$. If $n = m$ there always exists a vector $\dot{\mathbf{q}} = \mathbf{A}_{\mathbf{q}}^{-1}\dot{\mathbf{p}}^d$ in the configuration space that allows the robots to achieve the desired velocity. Now if $n \neq m$ then the Jacobian is not square and the system becomes redundant (if $n > m$), or there exist trajectories in the configuration space $\dot{\mathbf{q}}$ which cannot be feasible for any input $\dot{\mathbf{p}}^d$ (if $n < m$). Thus, some artifices must be used to control such systems when $n \neq m$. Moreover, in practice when $n \neq m$, even if the system is on a feasible configuration it may slightly move because of external disturbances and happen to be in one of the to scenarios mentioned before. This may imply that the cables become slack and, as a consequence, the assumption of tight cables could fail and the analysis would become more difficult.

Notice that we are not considering the dynamics of the system because its effects can be neglected for quasi-static motions. This translates into a simpler design of the controller, that is kinematic-based, hence we do not need to estimates a lot of parameters.



(a) $(N_{1c}, N_{2c}, N) = (0,3,3)$ **(b)** $(N_{1c}, N_{2c}, N) = (2,2,4)$

(c) $(N_{1c}, N_{2c}, N) = (4,1,5)$ **(d)** $(N_{1c}, N_{2c}, N) = (6,0,6)$

Figure 3.1. Possible configurations of square systems where a common object is manipulated by a group of UAVs: all the possible combinations of (N_{1c}, N_{2c}, N) introduced in Section 3.2.2 are represented.

3.2.2 Examples of square systems

In literature we can find several systems which can be described as squared systems. In this Section we show that the model in (2.3) includes different multi-robot systems ranging from team of aerial vehicles for cooperative manipulation of cable-suspended loads, to team of generic robots for formation control; An example is the multi-robot system depicted in Figure 3.1a, where three UAVs transport a common platform. In this case each robot is linked to the load through two rigid cables, hence one angle $q_{i_1} \in \mathbb{R}$ suffices to express $\mathbf{p}_i$ with respect to $\mathbf{q}_V$, while $\mathbf{q}_V \in \mathbb{R}^{6\times 1}$ is the load pose; thus, the generalized coordinates are chosen as $\mathbf{q} = [q_1\ q_2\ q_3\ \mathbf{q}_V^\top]^\top$. The robots positions $\mathbf{p}_i \in \mathbb{R}^{3\times 1}$ are collected in the vector $\mathbf{p} = [\mathbf{p}_1^\top\ \mathbf{p}_2^\top\ \mathbf{p}_3^\top]^\top$. Thus, the velocities vectors $\dot{\mathbf{q}}, \dot{\mathbf{p}} \in \mathbb{R}^{9\times 1}$ have the same dimension and $\mathbf{A}_\mathbf{q} \in \mathbb{R}^{9\times 9}$ results to be a square matrix. However this is a particular case of a larger class: different square systems can be obtained by simply changing the number of robots supporting the platform or the number of cables linking each robot to it. Notice that if one cable was used instead of two, as shown in Figure 3.1b, then $\mathbf{q}_i = [q_{i_1}\ q_{i_2}]^\top \in \mathbb{R}^2$ because each cable could move in two directions (assume that movements about the cable axis are not allowed); of course, in this case the system is no more guaranteed to be square. As introduced before, it holds $n = 3N$ and $m = 6 + 2N_{1c} + N_{2c}$ where $N_{1c} \geq 0$ is the number of robots attached to the load through one cable and $N_{2c} \geq 0$ indicates the number of robots attached through two cables. Of course $N = N_{1c} + N_{2c}$ and for a square system $n = m$, that is $3N = 6 + 2N_{1c} + N_{2c}$. From these relations it turns out that

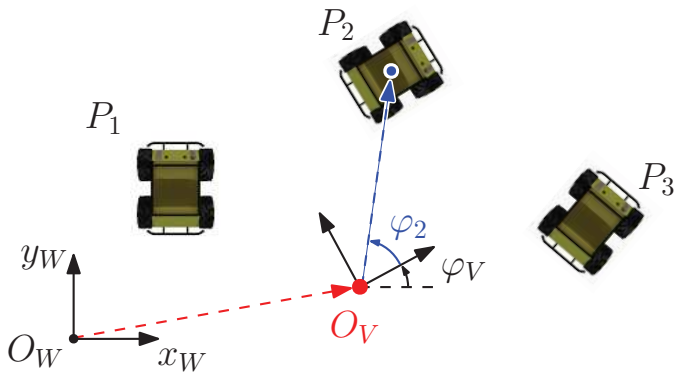


Figure 3.2. Representation of a multi-robot system where three robots must keep the same orientation φ_i with respect to the reference frame centered in the point O_V during the assigned task.

$N_{2c} \leq 3$, $N_{1c} \leq 6$ and $N \leq 6$; in particular, the possible configurations are (N_{1c},N_{2c},N) : (0,3,3) in Figure 3.1a, (2,2,4) in Figure 3.1b, (4,1,5) in Figure 3.1c, (6,0,6) in Figure 3.1d.

An other example of square-system can be found in a different context: consider the case of a formation control problem Franchi et al. (2012), like the one represented in Figure 3.2: three ground robots moves in a 2-dimensional space while respecting some constraints with respect to the frame $\mathcal{F}_V = \{O_V, \mathbf{x}_V, \mathbf{y}_V\}$, that represents the pivot (it can be the CoM of the system or a robot to be escorted). The position $P_i(x_i, y_i)$ of each robot can be described as a function of the pivot coordinates (x_V, y_V), of the distance d_i from P_i to O_V and the angle φ_i between the line $O_V - P_i$ and the axis x_W of the world reference frame $\mathcal{F}_W = \{O_W, \mathbf{x}_W, \mathbf{y}_W\}$. It is trivial to write down the position of P_i in $\mathcal{F}_W$:

$$\mathbf{p}_i = \begin{bmatrix} x_i & y_i \end{bmatrix}^\top = \mathbf{h}(\mathbf{q}_i, \mathbf{q}_V)$$

where $\mathbf{q}_i = [d_i \ \varphi_i]^\top \in \mathbb{R}^2$, $\mathbf{q}_V = [x_V \ y_V \ \varphi_V]^\top \in \mathbb{R}^3$ and

$$\mathbf{h}(\mathbf{q}_i, \mathbf{q}_V) = \begin{bmatrix} x_V \\ y_V \end{bmatrix} + \begin{bmatrix} cos(\varphi_V) & -sin(\varphi_V) \\ sin(\varphi_V) & cos(\varphi_V) \end{bmatrix} \begin{bmatrix} d_i cos(\varphi_i) \\ d_i sin(\varphi_i) \end{bmatrix}$$

Let us define the vector of the robots positions as $\mathbf{p} = [\mathbf{p}_1^\top \ \mathbf{p}_2^\top \ \mathbf{p}_3^\top]^\top \in \mathbb{R}^{6\times 1}$ and the vector of generalized coordinates as $\mathbf{q} = [\mathbf{q}_1^\top \ \mathbf{q}_2^\top \ \mathbf{q}_3^\top \ \mathbf{q}_V^\top]^\top \in \mathbb{R}^{9\times 1}$. Hence we can write $\mathbf{h}(\mathbf{q}) := [\mathbf{h}(\mathbf{q}_1, \mathbf{q}_V)^\top \ \mathbf{h}(\mathbf{q}_2, \mathbf{q}_V)^\top \ \mathbf{h}(\mathbf{q}_3, \mathbf{q}_V)^\top]^\top$. We assume that the robots must perform a task forcing the angles φ_i to keep constant with respect to $\mathcal{F}_V$, hence the kinematic model is:

$$\dot{\mathbf{p}}' = \mathbf{A}_\mathbf{q} \dot{\mathbf{q}}$$

where $\dot{\mathbf{p}}' = [\dot{\mathbf{p}}^\top \ \dot{\varphi}_1 \ \dot{\varphi}_2 \ \dot{\varphi}_3]^\top$ and

$$\mathbf{A}_\mathbf{q} = \begin{bmatrix} \frac{\partial \mathbf{h}(\mathbf{q})}{\partial \mathbf{q}} \\ \\ 0\,1\,0\,0\,0\,0 \ \ldots \ 0 \\ 0\,0\,0\,1\,0\,0 \ \ldots \ 0 \\ 0\,0\,0\,0\,0\,1 \ \ldots \ 0 \end{bmatrix} \in \mathbb{R}^{9\times 9}$$

The previous examples show that the class of square systems includes multi-robot

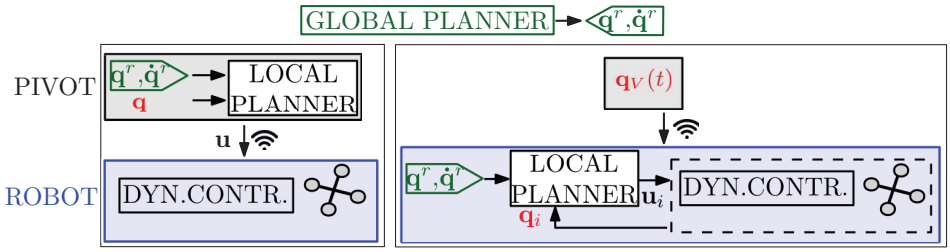


Figure 3.3. Representation of (3.6) (on the left) and (3.5) (on the right). The pivot is represented in gray, each agent (dynamical controller plus robot) in blue, the measurements from the sensors in red and the global planner in green. The dynamical controller converts $\mathbf{u}$ into forces for the robot. The wireless symbol refers to sampled communication.

systems already known in literature, hence it is worth to design a control law for trajectory tracking for this class.

A key feature of the systems considered in this work is that they exhibit a *star-like interaction topology* where each agent is connected to a central unit (the load). This topology is more evident when considering its differential kinematics.

3.2.3 Problem Formulation

The goal of the chapter is to design a feedback-based local planner in order to steer the system from an initial configuration $\mathbf{q}^0$ to a desired one $\mathbf{q}^r$. In general, $\mathbf{q}^r$ could be a waypoint sampled at the time instant $t_\ell, \ell \in \mathbb{N}$ from a trajectory $\mathbf{q}^p(t)$, with $t \in \mathbb{R}$, generated off-line by a global path planner. This can take into account high-level performance metrics such as obstacles avoidance, minimum energy trajectories and singularity avoidance (i.e., the configurations s.t. the Jacobian $\mathbf{A}_{\mathbf{q}^p(t)}$ is not full rank). The objective is to steer the system from a configuration $\mathbf{q}^0 \approx \mathbf{q}^p(t_\ell)$ to a final one $\mathbf{q}^r \approx \mathbf{q}^p(t_{\ell+1})$ faster than the intersample interval $\Delta t = t_{\ell+1} - t_\ell$. We assume that each agent has an inner-loop control sufficiently fast with respect to the dynamics of the entire system, such that the velocities of the variable vector $\mathbf{p}$ are fully controllable, i.e.:

$$\dot{\mathbf{p}}(t) = \mathbf{u}(t), \tag{3.4}$$

where $\mathbf{u} = [\mathbf{u}_1^\top \cdots \mathbf{u}_N^\top]^\top \in \mathbb{R}^n$. We consider two possible architectures, depicted in Figure 3.3:

$$\mathbf{u}_i(t) = \kappa_i^s(\mathbf{q}_i(t), \mathbf{q}_V(t); \mathbf{q}_i^r, \mathbf{q}_V^r), \quad i = 1, \dots, N \tag{3.5}$$

$$\mathbf{u}_i(t) = \kappa_i^c(\mathbf{q}(t); \mathbf{q}^r), \quad i = 1, \dots, N \tag{3.6}$$

where the former tries to maintain the same sparsity of the Jacobian: in this case, indeed, the pivot sends its pose $\mathbf{q}_V$ to all the agents and a local planner, installed on each of them, generates the reference velocity $\mathbf{u}_i$ to be tracked. This velocity can be reached in real-time by each agent thanks to the dynamical based controller that works at higher frequency with respect to the local planner. On the other hand, in (3.6) the local planner is implemented on the pivot that can measure all the system variables and computes directly the desired velocities to be sent to the agents. Notice that in this case the full

knowledge of [1] $\mathbf{q}$ must be available to the pivot. We will show that if $\mathbf{q}(t)$ is continuously accessible to the local planner, then (3.5) is sufficient to drive the system from $\mathbf{q}^0$ to $\mathbf{q}^r$ very effectively. On the other hand, under the more realistic scenario when some of the components of $\mathbf{q}(t)$ need to be sampled and transmitted via wireless, then the two architectures give rise to two strategies with different performance and computational requirements. Before proceeding, we define the set $\mathcal{B}_r(\mathbf{q}^r) := \{\mathbf{q} \in \mathbb{R}^m \,|\, \|\mathbf{q}-\mathbf{q}^r\| < r, r > 0\}$, where $\|\cdot\|$ is the Euclidean norm and the assumption:

Assumption 3.2.1. *The following relations hold:*

1. $n = m$, *i.e.,* $\mathbf{A}_\mathbf{q} \in \mathbb{R}^{n\times n}$
2. *the matrix* $\mathbf{A}_\mathbf{q}$ *is invertible and the map* $\mathbf{h}$ *is twice continuously differentiable for all* $q \in \mathcal{B}_r(\mathbf{q}^r)$. *In addition, these properties can be extended by continuity on the closure of such set, defined as* $\overline{\mathcal{B}}_r(\mathbf{q}^r)$.
3. $\mathbf{q}^0 \in \mathcal{B}_r(\mathbf{q}^r)$

The first point of Assumption 3.2.1 is adopted in the interest of space: we refer to these systems as *square systems* (see Figure 3.1). The second point allows defining the evolution of $\mathbf{q}(t)$ in Section 3.3 and 3.4 and can be satisfied in the points generated by the global path planner by imposing this condition as a constraint in the path generation process, hence such an r exists by continuity arguments. This assumption is common in robotics where the sequence of desired configurations are generated off-line. A possible approach to give such guarantees is presented in Manubens et al. (2013) where the authors define the set of feasible wrenches for the same kind of system and show the Jacobian matrix is not singular in that set.. The last point is guaranteed if the planner properly selects the waypoints such that $\|\mathbf{q}^p(t_{\ell+1}) - \mathbf{q}^p(t_\ell)\| < r$.

3.3 Local planner: continous-time

In this section, we describe the local planner that generates the desired vehicles velocity in the scenario of continuous time measurements. The goal is to steer $\mathbf{q}(t)$ to $\mathbf{q}^r$. Assuming $\mathbf{A}_{\mathbf{q}(t)}$ full-rank, from (3.3) the evolution of $\mathbf{q}$ is ruled by

$$\dot{\mathbf{q}}(t) = \mathbf{A}^{-1}_{\mathbf{q}(t)}\mathbf{u}(t). \tag{3.7}$$

A possible choice to define the desired vehicles velocities is:

$$\mathbf{u}(t) = -\mathbf{A}_{\mathbf{q}(t)}\mathbf{K}(\mathbf{q}(t) - \mathbf{q}^r), \tag{3.8}$$

where $\mathbf{K}$ is a gain matrix to be designed. With this choice, if $\mathbf{A}_{\mathbf{q}(t)}$ is invertible, we have that

$$\dot{\mathbf{q}} = -\mathbf{K}(\mathbf{q}(t) - \mathbf{q}^r). \tag{3.9}$$

[1] Note that we do not pursue control strategies of type $\mathbf{u}(t) = \kappa(\mathbf{p}(t); \mathbf{p}^r)$ where $\mathbf{p}^r := \mathbf{h}(\mathbf{q}^\mathbf{r})$ since the function $\mathbf{h}$ might not be perfectly known and avoidance of singular configurations cannot be guaranteed.

The feedback gain $\mathbf{K}$ needs to guarantee stability, but also to make the trajectory $\mathbf{q}(t)$ not to pass through a singularity. This is established in the next Proposition.

Proposition 3.3.1. *Consider the system (3.7)-(3.8) and Assumption 3.2.1. If $\mathbf{K}+\mathbf{K}^\top$ is strictly positive definite, then*

1. *for all $t \geq 0$, $\mathbf{q}(t) \in \mathcal{B}_r(\mathbf{q}^r)$*
2. *the trajectory $\mathbf{q}(t)$ converges exponentially fast to $\mathbf{q}^r$.*

Proof. 1) Take the Lyapunov function $V(\mathbf{q}) = \|\mathbf{q}-\mathbf{q}^r\|^2$. Then its time derivative is as follow

$$\dot{V}(\mathbf{q}) = -(\mathbf{q}-\mathbf{q}^r)^\top(\mathbf{K}+\mathbf{K}^\top)(\mathbf{q}-\mathbf{q}^r) < 0, \quad \mathbf{q} \neq \mathbf{q}^r,$$

if $\mathbf{K}+\mathbf{K}^\top > 0$, i.e., if the symmetric part of $\mathbf{K}$ is so. Hence $V(\mathbf{q})$ does not increase and therefore $\mathbf{q}(t) \in \mathcal{B}_r(\mathbf{q}^r) \ \ \forall t \geq 0$.
2) The solution of (3.9) is $\mathbf{q}(t) = e^{-\mathbf{K}t}\mathbf{q}^0 + \mathbf{q}^r$ that converges exponentially fast to $\mathbf{q}^r$ if $-\mathbf{K}$ is Hurwitz which is guaranteed by the hypothesis $\mathbf{K}+\mathbf{K}^\top > 0$. ∎

We conclude this section observing that, if we choose $\mathbf{K}$ to be block diagonal of the form $\mathbf{K} = \text{diag}\{\mathbf{K}_1, \dots, \mathbf{K}_N, \mathbf{K}_V\}$, where $\mathbf{K}_i \in \mathbf{R}^{m_i \times m_i}$ and $\mathbf{K}_V \in \mathbf{R}^{m_V \times m_V}$, then

$$\mathbf{u}_i = \mathbf{A}^{(i)}_{\mathbf{q}_i}\mathbf{K}_i(\mathbf{q}_i(t)-\mathbf{q}_i^r) + \mathbf{A}^{(i)}_{\mathbf{q}_L}\mathbf{K}_V(\mathbf{q}_V(t)-\mathbf{q}_V^r), \tag{3.10}$$

i.e., the desired velocity for the i-th robot does not depend on the other robots state, as defined in (3.5).

3.4 Local Planner: Sampled Measurements

We now consider the evolution of (3.7) under sampled dynamics, that is, we assume that $\mathbf{q}$ is measured on the time instants hT, $h = 0,1,2,\dots$ where T is the sampling time. Furthermore, we assume that the vehicles reference velocity $\mathbf{u}(t)$ is kept constant within a time window T using (3.8):

$$\mathbf{u}(t) = \mathbf{u}_h = -\mathbf{A}_{\mathbf{q}_h}\mathbf{K}(\mathbf{q}_h - \mathbf{q}^r), \qquad hT \leq t < (h+1)T,$$

for $h = 1,2,\dots$, and $\mathbf{q}_h := \mathbf{q}(hT)$. $\mathbf{K}$ can be chosen to satisfy Proposition 3.3.1 and s.t. $\mathbf{u}_i$ does not depend on $\mathbf{q}_j$ with $i \neq j$:

$$\mathbf{K} = k\mathbf{I}_n, \quad k \in \mathbb{R}_{>0},$$

with $\mathbf{I}_n \in \mathbb{R}^{n\times n}$ the identity matrix. In this scenario, assuming $A_{\mathbf{q}(t)}$ non singular, the evolution of $\mathbf{q}(t)$ becomes:

$$\dot{\mathbf{q}}(t) = -kA^{-1}_{\mathbf{q}(t)}A_{\mathbf{q}_h}(\mathbf{q}_h - \mathbf{q}^r), \qquad hT \leq t < (h+1)T. \tag{3.11}$$

The main goal of this section is to design k, possibly time varying, i.e., $k = k_h$, such that the stability of the system is still guaranteed. For the sake of notational convenience,

we apply a change of coordinates of type $\mathbf{e}(t) \leftarrow \mathbf{q}(t) - \mathbf{q}^r$ and the simplified notation $\mathcal{B}_0 := \mathcal{B}_r(\mathbf{e}^r)$, with $\mathbf{e}^r = \mathbf{0}$.

The design of k is based on the study of the following auxiliary system whose solution is characterized by interesting and useful properties that will be analyzed later on:

$$\begin{aligned} \dot{\mathbf{e}}'(\tau;\mathbf{e}_h) &= -A^{-1}_{\mathbf{e}'(\tau;\mathbf{e}_h)} A_{\mathbf{e}_h}\mathbf{e}_h =: \mathbf{f}(\mathbf{e}'(\tau;\mathbf{e}_h)) \\ \mathbf{e}'(0;\mathbf{e}_h) &= \mathbf{e}_h; \quad \mathbf{e}_h \in \mathcal{B}_0, \end{aligned} \tag{3.12}$$

where $\tau \in [0, kT]$ and $\mathbf{e}'(\,\cdot\,;\,\cdot\,) \in \mathbb{R}^m$. By direct inspection

$$\mathbf{e}(t) = \mathbf{e}'(k(t-hT);\mathbf{e}_h), \qquad hT \le t < (h+1)T. \tag{3.13}$$

Hence, once the solution $\mathbf{e}'(\tau;\mathbf{e}_h)$ is computed, then $\mathbf{e}(t)$ is obtained through shifting by hT and rescaling by k as long as $\mathbf{e}'(k(t-hT);\mathbf{e}_h)$ exists; then, $\mathbf{q}(t) = \mathbf{e}(t) + \mathbf{q}^r$. The major benefit of this approach is that the analysis of (3.12) is independent of the gain k and the sampling period T. From Assumption 3.2.1, and since $\mathbf{f}$ and $\frac{\partial \mathbf{f}}{\partial \mathbf{e}'}$ are continuous maps on a compact domain, the following properties follow, for some $a, b > 0$

$$\mathbf{f}(\mathbf{e}'(0;\mathbf{e})) = -\mathbf{e}, \quad \forall \mathbf{e} \in \overline{\mathcal{B}}_0 \tag{3.14}$$

$$\|\mathbf{f}(\mathbf{e}'(\tau);\mathbf{e})\| \le \|A^{-1}_{\mathbf{e}'} A_{\mathbf{e}}\|\|\mathbf{e}\| = a\|\mathbf{e}\|, \ \forall \mathbf{q}', \mathbf{e} \in \overline{\mathcal{B}}_0 \tag{3.15}$$

$$\left\| \frac{\partial \mathbf{f}(\mathbf{e}'(\tau);\mathbf{e})}{\partial \mathbf{e}'} \right\| \le b\|\mathbf{e}\|, \quad \forall \mathbf{e}', \forall \mathbf{e} \in \overline{\mathcal{B}}_0, \tag{3.16}$$

Since the flow $\mathbf{f}(\mathbf{e}';\mathbf{e}_h)$ is locally continuously differentiable in $\mathbf{e}'$, then for each $\mathbf{e}_h \in \mathcal{B}_0$ there exists $\delta(\mathbf{e}_h) > 0$ s.t. (3.12) has a unique solution $\mathbf{e}'(\tau;\mathbf{e}_h)$ for $\tau \in [0, \delta(\mathbf{e}_h))$. Without loss of generality we define $\tau_{max}(\mathbf{e}_h)$ the maximum time extension for which the unique solution $\mathbf{e}'(\tau;\mathbf{e}_h)$ exists for $\tau \in [0, \tau_{max}(\mathbf{e}_h))$. An interesting property of $\mathbf{e}'(\tau;\mathbf{e}_h)$ is described in Proposition 3.4.1:

Proposition 3.4.1. *Consider the dynamical system (3.12) and assume Assumption 3.2.1 holds true. Then the solution $\mathbf{e}'(\tau;\mathbf{e}_h)$ satisfies one of these two properties:*

(i) for all $\tau > 0$, it holds $\|\mathbf{e}'(\tau;\mathbf{e}_h)\| < \|\mathbf{e}_h\|$ and in such case $\tau_{max}(\mathbf{e}_h) = \infty$;

(ii) there exists $0 < \bar{\tau} < \tau_{max}(\mathbf{e}_h)$ such that $\|\mathbf{e}'(\bar{\tau};\mathbf{e}_h)\| = \|\mathbf{e}_h\|$ and $\|\mathbf{e}'(\tau;\mathbf{e}_h)\| < \|\mathbf{e}_h\|$ for all $0 < \tau < \bar{\tau}$.

Proof. Only two scenarios are possible, either $\|\mathbf{e}'(\tau;\mathbf{e}_h)\| < \|\mathbf{e}_h\|$ for all $\tau > 0$, from which $\tau_{max}(\mathbf{e}_h) = \infty$ follows, or not. If not, then $\overline{\tau} := \inf_{\tau>0}\{\tau \,|\, \|\mathbf{e}'(\tau;\mathbf{e}_h)\| \ge \|\mathbf{e}_h\|\}$ is well defined and finite. Now, let us assume that $\bar{\tau} = 0$: since $\mathbf{e}'(\tau;\mathbf{e}_h)$ is continuously differentiable in τ and since $\dot{\mathbf{e}}'(0;\mathbf{e}_h) = -\mathbf{e}_h$, then there exists $\dot{\mathbf{e}}'(0;\mathbf{e}_h)$ s.t. $\mathbf{e}_h^\top \dot{\mathbf{e}}'(0;\mathbf{e}_h) \ge 0 \Leftrightarrow -\mathbf{e}_h^\top \mathbf{e}_h = -\|\mathbf{e}_h\|^2 \le 0$ that is a contradiction. This implies that $\bar{\tau} > 0$. Finally, observe that the definition of $\bar{\tau}$ makes sense only if a solution $\mathbf{e}'(\tau;\mathbf{e}_h)$ exists, hence $\overline{\tau} \le \tau_{max}(\mathbf{e}_h)$. ∎

Based on the previous result, we can now define the following temporal variables:

$$\tau_s(\mathbf{e}_h) := \min_{\tau}\{\tau > 0 \,|\, \|\mathbf{e}'(\tau;\mathbf{e}_h)\| = \|\mathbf{e}_h\|\}, \quad \tau_s^* := \inf_{\mathbf{e}_h \in \mathcal{B}_0} \tau_s(\mathbf{e}_h),$$

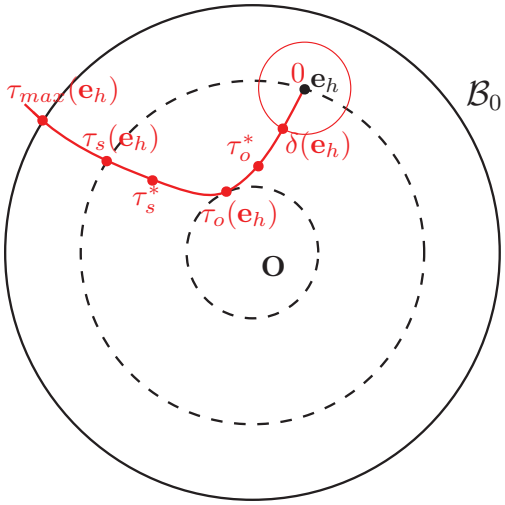


Figure 3.4. Depiction of $\mathcal{B}_0$, $\delta_{\mathbf{e}_h}$, τ_o^*, $\tau_o(\mathbf{e}_h)$, τ_s^*, $\tau_s(\mathbf{e}_h)$ and $\tau_{max}(\mathbf{e}_h)$. The time values are in red, and the points or sets in $\mathbb{R}^m$ in black.

$$\tau_o(\mathbf{e}_h) := \operatorname*{arginf}_{0\le\tau\le\tau_s(\mathbf{e}_h)} \|\mathbf{e}'(\tau;\mathbf{e}_h)\|, \quad \tau_o^* := \inf_{\mathbf{q}_h\in\mathcal{B}_0} \tau_o(\mathbf{q}_h),$$

where $\tau_s(\mathbf{e}_h) = \infty$ if $\|\mathbf{e}'(\tau;\mathbf{e}_h)\| < \|\mathbf{e}_h\|, \forall\tau$. These quantities are sketched in Figure 3.4. Basically, $\tau_s(\mathbf{e}_h)$ represents the first time that the solution $\mathbf{e}'(\tau;\mathbf{e}_h)$ hits the boundary of the ball centered at the origin and passing through the initial condition $\mathbf{e}_h$, while $\tau_o(\mathbf{e}_h)$, represents the time that $\mathbf{e}'(\tau;\mathbf{e}_h)$ is closest to the origin. We will show in the next section, that for any given set $\mathcal{B}_0$ we can find $0 < \bar{\tau}_o < \bar{\tau}_s$ and $0 \le \rho < 1$ s.t. $\|\mathbf{e}'(\bar{\tau}_o;\mathbf{e}_h)\| \le \rho\|\mathbf{e}_h\|$ for all $\mathbf{e}_h \in \mathcal{B}_0$, and $\bar{\tau}_o \le \tau_o^*$, $\bar{\tau}_s \le \tau_s^*$. The variables $\bar{\tau}_o$ and $\tau_o(\mathbf{e}_h)$ allow proposing two different strategies to design the gain k. The first is based on the observation that if $k = \frac{\bar{\tau}_o}{T}$, then $\mathbf{e}'(\tau;\mathbf{e}_h) \to \mathbf{0}\ \forall\mathbf{e}_h$ at a convergence rate ρ. In fact, from (3.13) we have $\|\mathbf{e}_{h+1}\| = \|\mathbf{e}'(kT;\mathbf{e}_h)\| = \|\mathbf{e}'(\bar{\tau}_o;\mathbf{e}_h)\| \le \rho\|\mathbf{e}_h\|$. This suggests an *offline procedure* to select k that will be described in the next Sect. 3.4.1. However, based on the definition of $\tau_o(\mathbf{e}_h)$, it might be likely that $\|\mathbf{e}'(\tau_o(\mathbf{e}_h);\mathbf{e}_h)\| < \|\mathbf{e}'(\tau_o^*;\mathbf{e}_h)\|$ for most $\mathbf{e}_h \in \mathcal{B}_0$. Therefore, an alternative approach is to select k at each instant h such that $k_h := \frac{\tau_o(\mathbf{e}_h)}{T}$. This idea suggests an *online strategy* that will be described in Sect. 3.4.2. Notice that the quantities τ_s^* and τ_o^* are of existential type and numerical estimations are not needed to perform our analysis. Even the time $\tau_s(\mathbf{e}_h)$ will be not computed for each configuration $\mathbf{e}_h$ because we will only need to know the value of its lower bound $\bar{\tau}_s$. Actually, the only value we need to estimate is $\tau_o(\mathbf{e}_h)$, as we can see in Section 3.4.2.

3.4.1 Off-line procedure (Stability and convergence rate)

From (3.13), if we show that $\tau_s^* > 0$, then the original system (3.12) is asymptotically stable for

$$kT < \tau_s^* \quad \forall\, \mathbf{e}(0) \in \mathcal{B}_0.$$

Note that if this condition is not satisfied, then we can find a time instant $\tau' \in [\tau_s^*, +\infty]$ and $\mathbf{e}(0)$ such that $\|\mathbf{e}'(\tau';\mathbf{e}(0))\| > \|\mathbf{e}'(0;\mathbf{e}(0))\|$, that is $\|\mathbf{e}(kT)\| > \|\mathbf{e}(0)\|$ for $k = \frac{\tau'}{T}$.

Although this does not imply instability of the whole trajectory, it is an undesired behavior. We now want to find an explicit lower bound $\bar{\tau}_s > 0$ for τ_s^*. To do that, we consider an expansion of the solution of (3.12) and numerically estimate the upper bound of the approximation error via an additional parameter μ. This allows deriving an analytical expression of $\bar{\tau}_s$.We recall that the solution of (3.12) can also be written as:

$$\mathbf{e}'(\tau;\mathbf{e}_h) = \mathbf{e}_h + \int_0^\tau \mathbf{f}(\mathbf{e}'(\tau';\mathbf{e}_h))d\tau', \quad 0 \le \tau < \tau_s(\mathbf{e}_h).$$

By using Taylor's theorem for multivariate functions with integral form of the remainder, it becomes

$$\begin{aligned}\mathbf{e}'(\tau;\mathbf{e}_h) &= \mathbf{e}_h + \tau\,\mathbf{f}(\mathbf{e}'(0);\mathbf{e}_h) + \\ &\quad + \tau^2 \int_0^1 (1-\epsilon)\frac{\partial \mathbf{f}(\mathbf{e}'(\epsilon\tau);\mathbf{e}_h)}{\partial \mathbf{e}'}\mathbf{f}(\mathbf{e}'(\epsilon\tau);\mathbf{e}_h)d\epsilon \\ &= (1-\tau)\mathbf{e}_h + \tau^2\mathbf{d}(\tau,\mathbf{e}_h), \quad 0 \le \tau < \tau_s(\mathbf{e}_h),\end{aligned} \tag{3.17}$$

where the reminder $\mathbf{d}$ has the property[2]:

Proposition 3.4.2. *There exists $0 < \lambda < \infty$ such that for all $q_h \in \mathcal{B}_0$ and for all $0 < \tau < \tau_s(\mathbf{e}_h)$ it holds $\|\mathbf{d}(\tau,\mathbf{e}_h)\| \le \lambda\|\mathbf{e}_h\|^2$.*

Proof. As a consequence of the properties (3.15) and (3.16), the reminder in (3.17) becomes

$$\begin{aligned}\|\mathbf{d}(\tau,\mathbf{e}_h)\| &\le \int_0^1 (1-\epsilon)\left\|\frac{\partial \mathbf{f}(\mathbf{e}'(\epsilon\tau);\mathbf{e}_h)}{\partial \mathbf{e}'}\right\|\|\mathbf{f}(\mathbf{e}'(\epsilon\tau);\mathbf{e}_h)\|d\epsilon \\ &\le \int_0^1 (1-\epsilon)ab\|\mathbf{e}_h\|^2 d\epsilon = \frac{1}{2}ab\|\mathbf{e}_h\|^2 = \lambda\|\mathbf{e}_h\|^2\end{aligned}$$

where $\lambda := \frac{1}{2}ab$ and a, b are the bounding constants introduced in (3.15) and (3.16). ∎

Notice that since $\mathbf{e}_h \in \mathcal{B}_o$, then $\|\mathbf{e}_h\| \le r$ and there exists

$$\mu := \frac{1}{2}abr, \tag{3.18}$$

such that $\|\mathbf{d}(\tau,\mathbf{e}_h)\| \le \frac{1}{2}ab\|\mathbf{e}_h\|^2 \le \mu\|\mathbf{e}_h\|$. Note that μ represents a rough estimate of the upper bound of $\|\mathbf{d}(\tau,\mathbf{e}_h)\|$. However, this estimate can be refined as follows. Let

$$\begin{aligned}\mathbf{d}'(\tau,\mathbf{e}_h) :=& \tau^2\mathbf{d}(\tau,\mathbf{e}_h) \overset{(3.17)}{=} \mathbf{e}'(\tau,\mathbf{e}_h) - (1-\tau)\mathbf{e}_h \quad \text{and} \\ \mu^* :=& \inf_\gamma\{\gamma \mid \|\mathbf{d}'(\tau,\mathbf{e}_h)\| \le \gamma\|\mathbf{e}_h\|\tau^2, \\ & \forall \mathbf{e}_h \in \mathcal{B}_0, \forall \tau \in (0,\tau_s(\mathbf{e}_h))\}.\end{aligned}$$

We are interested in providing an estimate $\hat{\mu}^*$ of μ^*. To this aim, we randomly pick

[2]Note that the reminder scales as $\|\mathbf{e}_h\|^2$ which implies that as $\mathbf{e}'$ becomes closer to the origin, the faster it converges.

samples in $\mathcal{B}_0$, $\mathbf{e}_h^i \in \mathcal{B}_0$, and for each of them we simulate $\mathbf{e}'(\tau_i;\mathbf{e}_h^i)$ for $\tau_i \in [0,\tau_s(\mathbf{e}_h^i)]$.

$$\hat{\mu}^* := \max_i \left\{ \frac{\|\mathbf{d}'(\tau_i,\mathbf{e}_h^i)\|}{\tau_i^2\|\mathbf{e}_h^i\|}, \forall \mathbf{e}_h^i \in \mathcal{B}_0, \forall \tau_i \in (0,\tau_s(\mathbf{e}_h^i)) \right\}. \tag{3.19}$$

Since $\hat{\mu}$ is computed on a sampled $\mathcal{B}_0$, we may discard some configurations which would give a larger value of the estimate. Hence $\hat{\mu} \leq \mu$ and $\hat{\mu} \to \mu$ as the number of samples increases. From (3.17) and Proposition 3.4.2, for $\mathbf{e}_h \in \mathcal{B}_0$ we have that

$$\|\mathbf{e}'(\tau;\mathbf{e}_h)\| \leq (|1-\tau| + \mu\tau^2)\|\mathbf{e}_h\|, \tag{3.20}$$

for all $0 \leq \tau < \tau_s(\mathbf{e}_h)$. In order to evaluate upper bounds for the convergence rate, we need to study the following function

$$g(\tau;\mu) := |1-\tau| + \mu\tau^2$$

Notice that

$$g(\tau;\mu) := |1-\tau| + \mu\tau^2 = \begin{cases} 1-\tau+\mu\tau^2 =: g^-(\tau;\mu) & \tau < 1 \\ -1+\tau+\mu\tau^2 =: g^+(\tau;\mu) & \tau \geq 1 \end{cases}$$

We will study the function $g(\tau;\mu)$ in three different scenarios: $\mu \in [0,\frac{1}{2})$, $\mu \in [\frac{1}{2},1)$ and $\mu \geq 1$. We start by observing that

$$g(0;\mu) = 1,\ g(1;\mu) = \mu,\ \frac{dg^+}{d\mu} = 1 + 2\mu\tau > 0$$

and by defining the minimum of $g^-(\tau;\mu)$ and its minimizer with respect to τ as

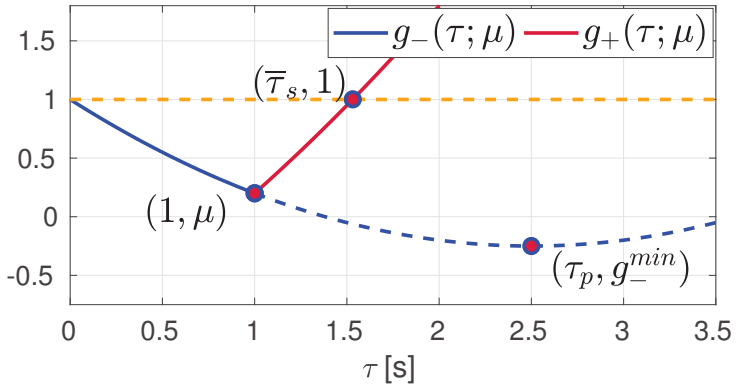
$$\tau_p(\mu) = \arg\min_\tau g^-(\tau;\mu) \Leftrightarrow \frac{dg^-(\tau;\mu)}{d\tau} = 0 \Longrightarrow \tau_p(\mu) = \frac{1}{2\mu}$$

We now note that in the first scenario $\mu \in [0,\frac{1}{2})$, $\tau_p(\mu) \geq 1$ which implies that the function $g(\tau;\mu)$ is monotonically decreasing for $\tau \in [0,1]$ and monotonically increasing for $\tau > 1$. In the second scenario $\mu \in [\frac{1}{2},1)$, $\tau_p(\mu) < 1$, therefore $g(\tau;\mu)$ is monotonically decreasing for $\tau \in [0,\tau_p(\mu)]$ and monotonically increasing for $\tau > \tau_p(\mu)$.
Finally note that for $\mu < 1$, $g(1;\tau) < 1$, therefore there exists a unique $\overline{\tau}_s(\mu)$ such that $g(\overline{\tau}_s(\mu);\mu) = g^+(\overline{\tau}_s(\mu);\mu) = 1$, while for $\mu > 1$, $g(1;\tau) > 1$, therefore there exists a unique $\overline{\tau}_s(\mu)$ such that $g(\overline{\tau}_s(\mu);\mu) = g^-(\overline{\tau}_s(\mu);\mu) = 1$. A pictorial representation of the three scenarios is shown in Figure 3.5. We are now ready to compute the stability region and convergence rate.

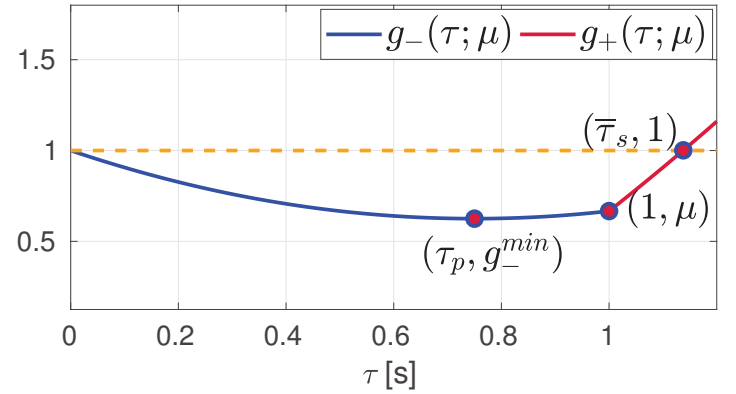
Proposition 3.4.3. *Given μ, the function $g(\tau;\mu)$ is strictly smaller than 1, i.e., $g(\tau;\mu) < 1$ for $\tau \in (0,\overline{\tau}_s(\mu))$, where*

$$\overline{\tau}_s(\mu) := \begin{cases} \frac{4}{1+\sqrt{1+8\mu}} & \text{if} \quad \mu < 1 \\ \frac{1}{\mu} & \text{if} \quad \mu \geq 1 \end{cases}, \tag{3.21}$$

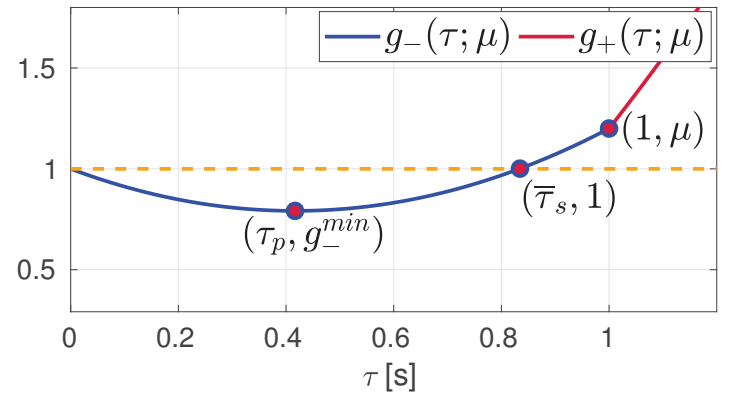
where $\overline{\tau}_s(\mu)$ is such that $g(\overline{\tau}_s(\mu);\mu) = 1$.



(a) $\mu \leq \frac{1}{2}$



(b) $\frac{1}{2} \leq \mu \leq 1$



(c) $\mu \geq 1$

Figure 3.5. Representation of $g(\tau;\mu)$ in the three scenarios.

Proof. According to the analysis above, the stability set is given by:

$$\mathcal{T} := \{\tau \,|\, g(\tau;\mu) < 1\} = (0, \overline{\tau}_s(\mu))$$

More specifically, we have two scenarios depending whether the parameter μ is smaller or grater than unity.

If $\mu < 1$ then $-1+\tau+\mu\tau^2 = 1$. Hence:

$$\overline{\tau}_s(\mu) = \frac{-1+\sqrt{1+8\mu}}{2\mu} = \frac{4}{1+\sqrt{1+8\mu}}$$

while

$$\mu > 1 \Longrightarrow 1-\tau+\mu\tau^2 = 1 \Longrightarrow \overline{\tau}_s(\mu) = \frac{1}{\mu}$$

which can be summarized in

$$\overline{\tau}_s(\mu) \begin{cases} \frac{4}{1+\sqrt{1+8\mu}} & \mu < 1 \\ \frac{1}{\mu} & \mu \geq 1 \end{cases} \tag{3.22}$$

■

Note that since $\overline{\tau}_s$ is obtained by using upper bounds on some terms, it holds that $\overline{\tau}_s \leq \tau_s^* \leq \tau_{max}(\mathbf{e}_h)$. Hence, in the following set

$$\mathcal{T} := \{\tau \,|\, g(\tau;\mu) < 1\} = (0, \overline{\tau}_s(\mu)),$$

the norm $\|\mathbf{e}'(\tau;\mathbf{e}_h)\|$ decreases with respect to $\|\mathbf{e}_h\|$. We now want to find the time $\overline{\tau}_o(\mu) \leq \tau_o^*$ in order to maximally decrease toward the origin, and the relative decreasing rate $\rho(\mu)$, i.e.,

$$\overline{\tau}_o(\mu) := \arg\min_{\tau} g(\tau;\mu), \quad \rho(\mu) = g(\overline{\tau}_o(\mu);\mu).$$

The following proposition provides the values attained by $\overline{\tau}_o(\mu)$ and $\rho(\mu)$, which are obtained from their definitions and the computation of $g(\tau;\mu)$.

Proposition 3.4.4. *Consider function $g(\tau;\mu)$. Then*

$$\overline{\tau}_o(\mu) = \begin{cases} 1 & \mu < \frac{1}{2} \\ \frac{1}{2\mu} & \mu \geq \frac{1}{2} \end{cases} \quad \textit{and} \quad \rho(\mu) = \begin{cases} \mu & \mu < \frac{1}{2} \\ 1-\frac{1}{4\mu} & \mu \geq \frac{1}{2} \end{cases} .$$

Proof. We want to find the optimal stopping time $\overline{\tau}_o(\mu)$ in order to maximally decrease toward the origin, and the relative decrease rate $\rho(\mu)$, i.e.

$$\overline{\tau}_o(\mu) := \arg\min_{\tau} g(\tau;\mu), \quad \rho(\mu) = g(\overline{\tau}_o(\mu);\mu)$$

We can distinguish two scenarios, depending whether the parameter μ is smaller or greater than $\frac{1}{2}$. More specifically, for $\mu < \frac{1}{2}$ the function $g^-(\tau;\mu)$ is monotonically decreasing for $\tau < 1$, and therefore $\tau_o(\mu) = 1$, while for $\mu > \frac{1}{2}$ then $\tau_o(\mu) = \tau_p(\mu) = \frac{1}{2\mu}$. This can be summarized as

$$\overline{\tau}_o(\mu) = \begin{cases} 1 & \mu < \frac{1}{2} \\ \frac{1}{2\mu} & \mu \geq \frac{1}{2} \end{cases} \tag{3.23}$$

By substitution is easy to verify that

$$\rho(\mu) = \begin{cases} \mu & \mu < \frac{1}{2} \\ 1-\frac{1}{4\mu} & \mu \geq \frac{1}{2} \end{cases} \tag{3.24}$$

■

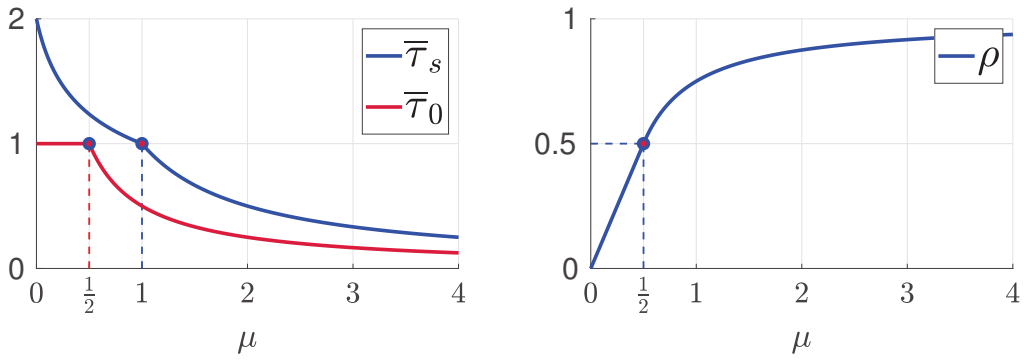


Figure 3.6. Representation of $\overline{\tau}_s(\mu)$, $\overline{\tau}_0(\mu)$ and $\rho(\mu)$.

Note that $\rho < 1$ as shown in Figure 3.6. The proof is available at Rossi, Tognon, Carli, Cortés, and Franchi (2019). A representation of $\overline{\tau}_s(\mu)$, $\overline{\tau}_o(\mu)$ and $\rho(\mu)$ is reported in Figure 3.6. Note that if $\mu = 0$ then $\rho = 0$ and we obtain a dead-beat controller. Indeed, from (3.20) with $k = \frac{\bar{\tau}_o}{T}$, it holds $\|\mathbf{e}(t)\| = \|\mathbf{e}'(\bar{\tau}_o(\mu); \mathbf{e}_h)\| = 0$ in one step. Moreover, for $\mu < \frac{1}{2}$, $\overline{\tau}_o(\mu) = 1$ regardless of μ and this is an indication of robustness. The previous result suggests that a possible choice for the optimal offline gain k, once the sampling time T is known, is

$$k^* = \frac{\overline{\tau}_o(\mu)}{T}, \tag{3.25}$$

as formally established in the next proposition.

Proposition 3.4.5. *For all* $\mathbf{e}(0) \in \mathcal{B}_0$ *the following inequality holds:*

$$\|\mathbf{e}(hT)\| \leq \rho^h(\mu)\|\mathbf{e}(0)\|$$

and $\|\mathbf{e}(t)\| \leq \|\mathbf{e}(hT)\|$ *for all* $hT \leq t < (h+1)T$.

Proof. From Proposition 3.3.1 and (3.20) and recalling that $\rho(\mu) < 1$:

$$\|\mathbf{e}_{h+1}\| \leq \|\mathbf{e}'(\tau; \mathbf{e}_h)\| \leq \rho(\mu)\|\mathbf{e}_h\| \leq g(\tau; \mu)\|\mathbf{e}_h\|$$

Hence $\|\mathbf{e}_{h+1}\| \leq \rho(\mu)\|\mathbf{e}_h\| \leq \rho(\mu)^2\|\mathbf{e}_{h-1}\| \leq \rho(\mu)^{h+1}\|\mathbf{e}_0\|$.

■

Notice that, from Proposition 3.4.5, it turns out that the origin is an asymptotically stable equilibrium for the system and the proposed *offline* strategy converges exponentially fast with a rate at least $\rho(\mu)$, $\forall \mathbf{e}(0) \in \mathcal{B}_0$ that is included in the corresponding basin of attraction. We conclude this section observing that, since μ can be computed apriori before running the algorithm, then the offline strategy is amenable of both distributed and centralized implementations.

3.4.2 Online model-predictive procedure

In this section we consider the possibility to numerically compute the future trajectory $\mathbf{e}'(\tau, \mathbf{e}_h)$ based on the model dynamics $\mathbf{f}(\mathbf{q}; \mathbf{e}_h)$ and the current position $\mathbf{e}_h$. This implies that also $\tau_o(\mathbf{e}_h)$ can be computed at any time step h. If so, under the assumption that

the input is kept constant for the following time interval T, we can propose the following input

$$\mathbf{u}(t) = \mathbf{u}_h = -k_h A_{\mathbf{e}_h} \mathbf{e}_h, \qquad hT \le t < (h+1)T,$$

where

$$k_h := \frac{\tau_o(\mathbf{e}_h)}{T}. \tag{3.26}$$

A more precise characterization of the convergence properties of this strategy is stated in the next proposition.

Proposition 3.4.6. *Consider the system in (3.11) with a time varying sequence of gains $k_0, k_1, k_2, \ldots$, where the generic k_h is given as in (3.26). Then the system satisfies the following properties:*

- $\|\mathbf{e}(t)\| \le \|\mathbf{e}(hT)\|$ *for all* $hT \le t < (h+1)T$;
- *the convergence rate of the sampled dynamics is at least quadratic;*
- *the gain k_h tends to $\frac{1}{T}$ as $h \to \infty$, that is* $\lim_{h\to\infty} k_h = \frac{1}{T}$.

Proof. Observe that, according to (3.26), we necessarily have:

$$\|\mathbf{e}_{h+1}\| = \|\mathbf{e}'(\tau_o(\mathbf{e}_h); \mathbf{e}_h)\| \le \|\mathbf{e}'(\overline{\tau}_o; \mathbf{e}_h)\| \le \rho \|\mathbf{e}_h\|,$$

hence the proposed scheme is exponentially stable with rate ρ for any T. Since in the online scenario $r = \|\mathbf{e}_h\| \to 0$, then $\mu = \frac{1}{2}ab\|\mathbf{e}_h\| \to 0$. As so, there exists $\bar{h}$ s.t. $\mu < \frac{1}{2}$ for $h > \bar{h}$. Then from Proposition 3.4.4, $\rho(\mu) = \mu$ and $\|\mathbf{e}_{h+1}\| \le \rho\|\mathbf{e}_h\| = \frac{1}{2}ab\|\mathbf{e}_h\|^2$ for $h > \bar{h}$. As a consequence

$$\limsup_{h\to+\infty} \frac{\|\mathbf{e}_{h+1}\|}{\|\mathbf{e}_h\|^2} \le \frac{1}{2}ab,$$

and $\|\mathbf{e}_h\| \le (\frac{1}{2}ab\|\mathbf{e}_0\|)^{(2^h-1)}\|\mathbf{e}_0\|$. Since $ab > 0$, then the quadratic convergence of the sequence $\|\mathbf{e}_h\|$ is guaranteed. Moreover, $\mu \to 0$ implies $\|\mathbf{d}(\tau, \mathbf{e}_h)\| \to 0$ and the second term in (3.17) becomes negligible. So, $\mathbf{e}'(\tau; \mathbf{e}_h) \to (1-\tau)\mathbf{e}_h$ is minimized for $\tau = 1$. In the online scenario, the optimal gain is chosen as in (3.26), where $\tau_o(\mathbf{e}_h)$ corresponds to the minimum norm, hence $\tau_o(\mathbf{e}_h) = 1$ and $k_h = \frac{1}{T}$. ∎

Based on the definition of $\tau_o(\mathbf{e}_h)$ and on Proposition 3.4.6, we expect the online strategy to exhibit a faster convergence than the offline one. This fact is supported also by the numerical results reported in the next section. However, the higher rate of convergence comes at the price of a heavier computational load. Indeed $\tau_o(\mathbf{e}_h)$ needs to be estimated at each iteration and a global knowledge of the vector $\mathbf{e}_h$ is required; this implies that the online strategy cannot be implemented distributively, but only in a centralized fashion.

3.5 Simulation Results

The performance of the proposed steering method is evaluated and compared in this section, reporting the results obtained in Matlab. We chose the system depicted in Figure 3.7 where three flying robots transport a common platform: this is a square

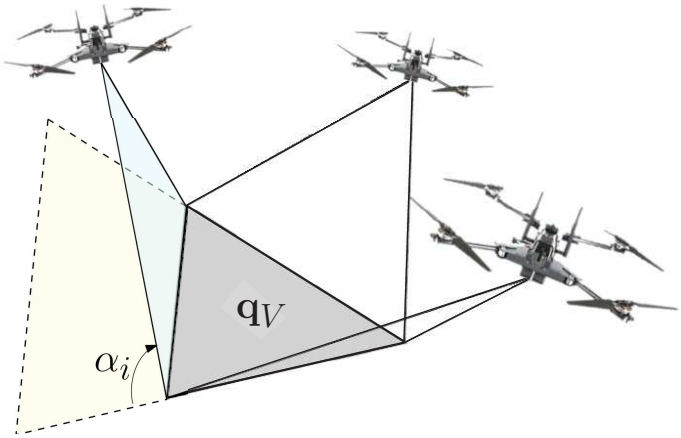


Figure 3.7. Example of a multi-agent system that transports a payload. α_i is the angle between the cables and the load.

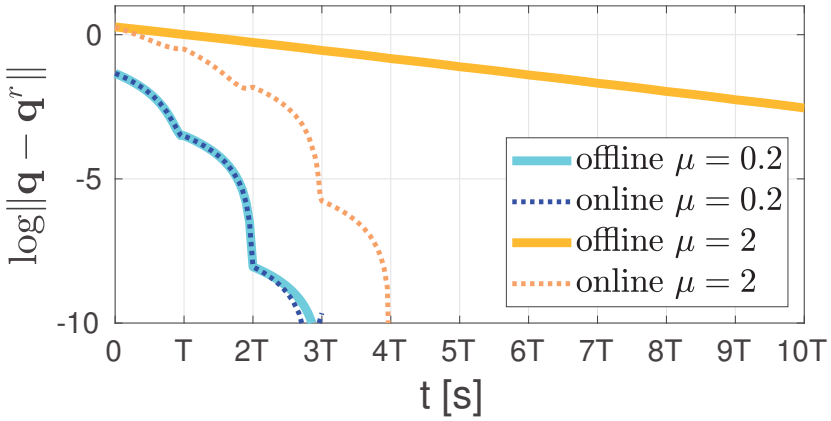


Figure 3.8. In the figure, label *offline* refers to the choice (3.25), while label *online* refers to (3.26).

system and can be represented by (3.1) and (3.3). The kinematic model of the system was derived by following the steps in Appendix B.1.1. In particular, the platform edges of the simulated system are $1\,[m]$ long and the cables linking the robots to the load measure $1.5\,[m]$. The sampling time is $T = 0.01\,[s]$. The quantity $\hat{\mu}^*$ is estimated both for a small $\mathcal{B}_o$ with radius $r_1 = 0.08$ (that gave $\hat{\mu}^* < \frac{1}{2}$) and for a larger one with $r_2 = 0.5$ ($\hat{\mu}^* > \frac{1}{2}$), following the reasoning of Section 3.4.1. Observe that r could be set as an optimization parameter to generate the reference path. Then, a desired configuration $\mathbf{q}^r$ s.t. $\mathbf{q}^r_V = \frac{1}{2}[1\,1\,1\,0\,0\,0]^\top$ and $\mathbf{q}^r_\alpha = [75°\ 75°\ 75°]^\top$ is chosen and two initial conditions are considered, one s.t. $\|\mathbf{q}_1(0) - \mathbf{q}^r\| < r_1$ and the second s.t. $r_1 < \|\mathbf{q}_2(0) - \mathbf{q}^r\| < r_2$. Finally, from Proposition 3.4.4, $\bar{\tau}_o$ is obtained based on the estimate $\hat{\mu}^*$, while the gain k^* is computed with (3.25). At this point, the system in (3.12) is simulated until $\tau = \bar{\tau}_o$ using the Matlab function ODE; then, the solution of the original system, for $hT \leq t < (h+1)T$, is retrieved using (3.13). The procedure is repeated for every iteration. Conversely, in the online strategy the optimal gain $k = k_h$ that brings the system the closest to the desired configuration, is recomputed at each instant hT based on the time $\tau_o(\mathbf{e}_h)$. We implicitly assumed that the machine used for the simulations can compute k_h in

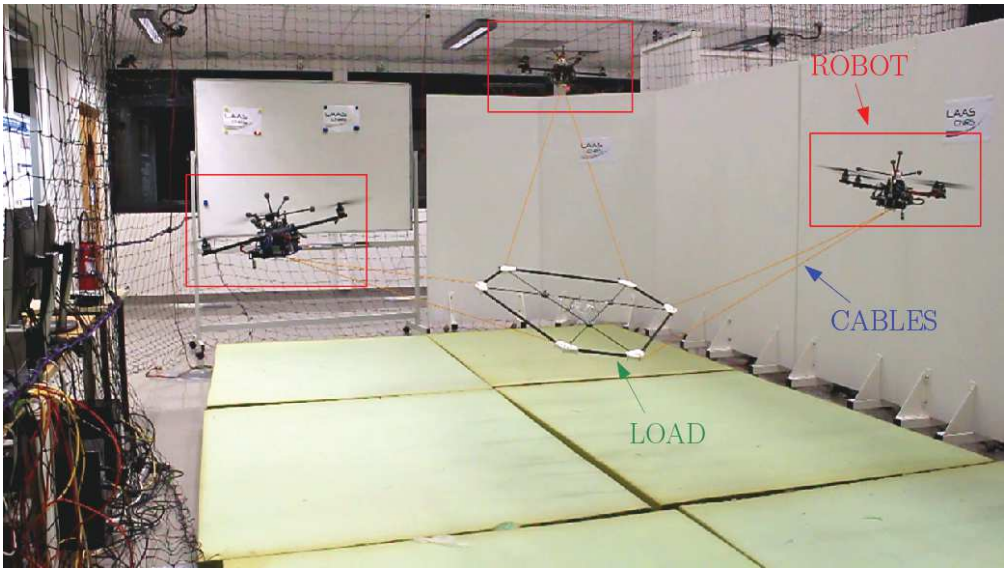


Figure 3.9. Robotic system called Fly-crane used for experiments.

less than T seconds. Moreover, T can be designed by the path planner to satisfy this condition. Figure 3.8 shows the simulation results described above, where, for μ smaller and greater than $\frac{1}{2}$, the online and offline strategies are compared. We observe that for $\mu < \frac{1}{2}$ the two strategies have quite the same behavior, as expected from the proof of Proposition 3.4.5. Instead, as $\mu > \frac{1}{2}$, the offline strategy is not as efficient as the online one, since the convergence rate $\rho(\mu) \to 1$, as described in Proposition 3.4.4. In particular, it is emphasized the difference between the exponential (offline strategy) and doubly exponential (online) convergence rate described respectively in Proposition 3.4.5 and 3.4.6. The soundness of the proposed strategy is confirmed by preliminary simulations on a dynamic simulator with force controller, not reported here for space limitation.

3.6 Experiments

In this section we are interested in the validation of our results on a real system, that was derived as described in Appendix B.1.1: we tested our strategy on the Fly-crane, depicted in Figure 3.9, where three aerial robots transport a common platform through six cables. The system was built at Laboratoire d'analyse et d'architecture des systèmes (LAAS-CNRS) in Toulouse, France. We performed our experiments in-door and the use of a real system allows to check if our solution is robust to disturbances and model uncertainties, given that it is based on the kinematic model of the system. Each robot has a mass of 1.03 [Kg], the platform 0.338 [Kg], and the length of each cable is 1.2 [m]. Each aerial vehicle is equipped with a standard flight-controller, four brushless motor controllers regulating the propeller speed in closedloop Franchi and Mallet (2017), and an on-board PC that runs the state estimator and the velocity controller[3]. The proposed kinematic controller, implemented in Matlab-Simulink, runs on a desktop PC which sends the commanded velocities to the robots through a wifi connection at 100 [Hz]. The control loop is then

[3]The software framework is based on TeleKyb which is open-source and available at https://git.openrobots.org/ projects/telekyb.

closed based on the estimated state of the vehicles and of the platform. These estimations are computed onboard at 1 [kHz] by an unscented Kalman filter that fuses the Motion Capture (MoCap) System measurements (at 120 [Hz]) with the IMU measurements (at 1 [kHz]). During the first phase of the experiment, the platform is lifted from ground, and the system is brought to a non-singular initial configuration. In this phase, the aerial vehicles are independently controlled by a standard position controller. The proposed planner is activated right after. The trajectory is designed to take the system from its initial configuration $\mathbf{q}_V(0) = [0[\text{m}]\ 0[\text{m}]\ 0.4[\text{m}]\ 0^\circ\ 0^\circ\ 0^\circ]^\top$ and $\mathbf{q}_i(0) = 55^\circ, i = 1,2,3$ to the final configuration $\mathbf{q}_V^r = \mathbf{q}_V(0) + [0.3[\text{m}]\ 0[\text{m}]\ 0.3[\text{m}]\ 20^\circ\ 10^\circ\ 45^\circ]^\top$ and $\mathbf{q}_1^r = \mathbf{q}_1(0) - 15^\circ$, $\mathbf{q}_2^r = \mathbf{q}_2(0) + 5^\circ$ and $\mathbf{q}_3^r = \mathbf{q}_3(0) + 15^\circ$.

Actually, motivated by the fact the usually in robotics a system should track a trajectory instead of simply reaching a constant configuration, we tested our strategies by assigning a sequence of configurations to the system. Indeed, we explained in Section 5.2 that in general, $\mathbf{q}^r$ could be a waypoint sampled at the time instant $t_\ell, \ell \in \mathbb{N}$ from a trajectory $\mathbf{q}^p(t)$, with $t \in \mathbb{R}$, generated off-line by a global path planner. Thus the objective is to steer the system from a configuration $\mathbf{q}^0 \approx \mathbf{q}^p(t_\ell)$ to the desired one $\mathbf{q}^r \approx \mathbf{q}^p(t_{\ell+1})$, faster than the intersample interval $\Delta t = t_{\ell+1} - t_\ell$. Hence, we generated a path such that, firstly, the load is lifted of 0.3[m], then moved on the x-axis of the same distance. Then the load is rotated of 40° about the z-axis and in the end, from this configuration, at 30[s], the final configuration is assigned. Notice that the final configuration reached by the fly-crane at the end of the first phase is slightly different from $\mathbf{q}_0$: this is due to external disturbances and model uncertaintes which do not allow the robots to stay exactly still while sustaining the pay-load. In order to compute the off-line gain we adopted the strategy proposed in Section 3.4.1: in particular, we considered a ball $\mathcal{B}_0$ characterized by a radius r_0 large enough s.t. the ball contains all the configurations along the path. Then, we could compute the off-line gain and compare the technique to the on-line scenario, where the gain is updated every time a new measurement is available. In Figure 3.10 the behavior of the controlled system is represented for both the on-line and off-line techniques, under a sampling time of $T = 1.5$[s]. In detail, on the first row the position of the load is depicted, while on the second row its orientation (roll, pitch and yaw angles). The last row represents the angles between the cables and the load. As expected from the theoretical analysis, the on-line strategy allows to converge to the desired configurations much faster with respect to the off-line technique. Notice that in both cases there is a delay in tracking the reference trajectory; this behavior is motivated by the fact that no velocity term is present in the planner law. In Section 5 we propose a solution to this problem, by adding a feed-forward to the local planner law. In Figure 3.11 instead the desired robots velocities, generated by the local planner, are represented. Since they are updated only when a new measurement is available, they are piece-wise constant signals with a period of T seconds. Notice that the desired velocities generated with the on-line technique are larger with respect to the ones obtained by choosing the off-line strategy: indeed, in the first case the planner is more aggressive since the gain is updated based on the current configuration, hence the system can perform faster maneuvers. This is confirmed by Figure 3.12 where both the optimal stopping time τ_0 and the gain k are depicted. Actually, for the off-line case, we plotted $\bar{\tau}_0$ that was computed using (3.23). Notice that the behavior of these variables reflects Proposition 3.4.6, where we stated

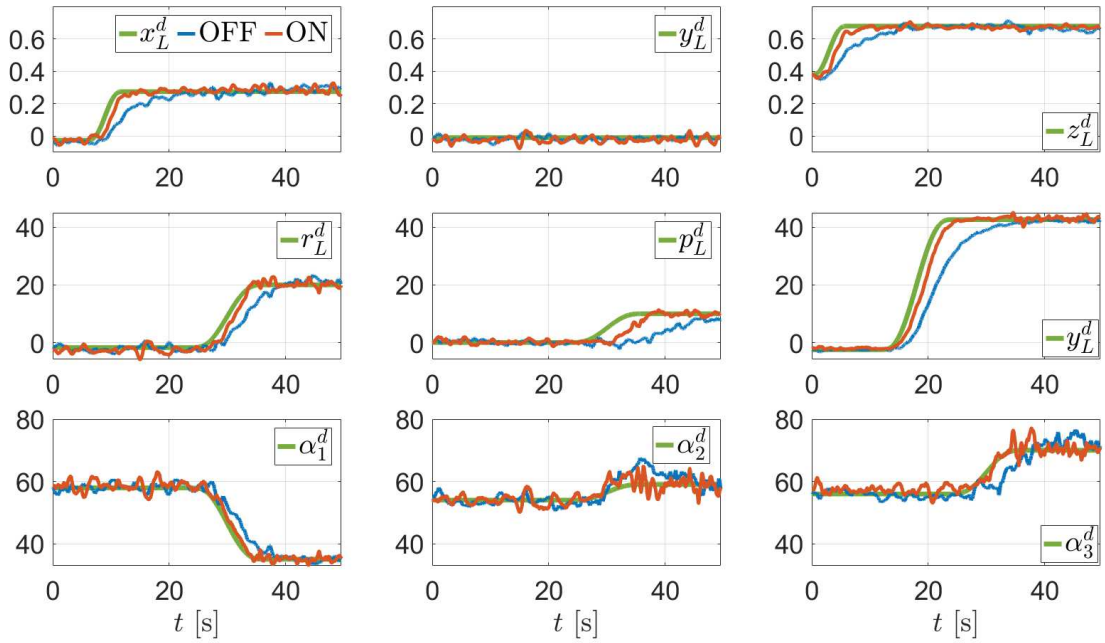


Figure 3.10. Representation of the Lagrangian variables $\mathbf{q}(t)$ when the off-line and on-line technique are compared. The used sampling time is $T = 1.5$ [s]. The first two rows represent respectively the position x_V, y_V, z_V and orientation r_V, p_V, y_V (roll, pitch and yaw) of the load. On the last row the angles $\alpha_i := q_i$, $i = 1,2,3$ between the cables and the load are depicted.

that $\tau_0 \to 1$ and $k_h \to \frac{1}{T}$ as $\mathbf{q} \to \mathbf{q}^r$.

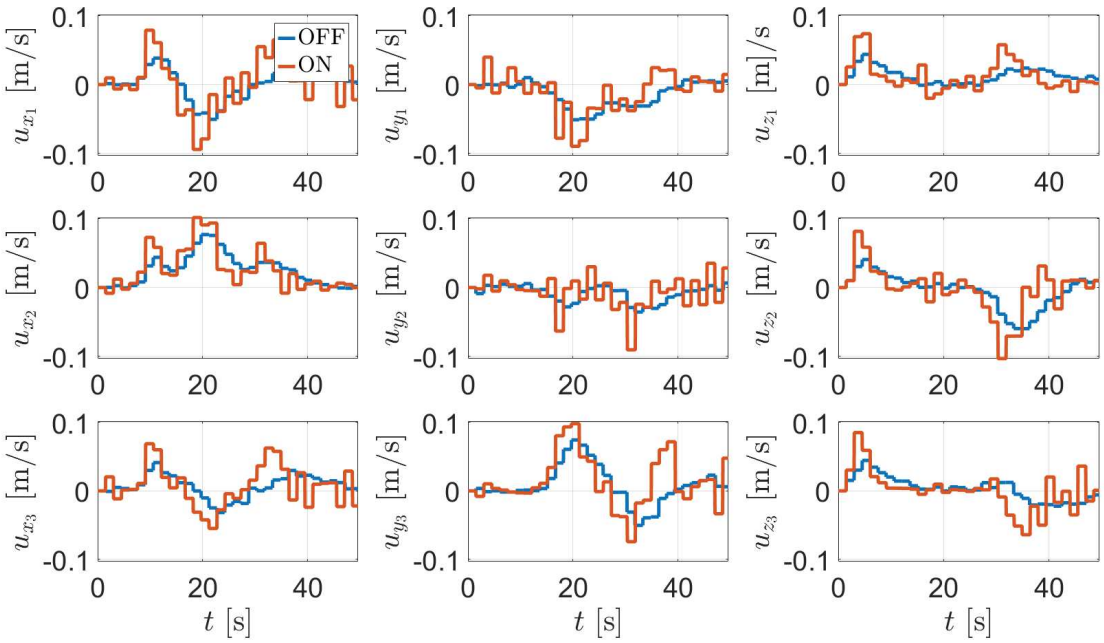


Figure 3.11. Representation of robots velocities $\dot{\mathbf{p}}(t)$ when the off-line and on-line technique are compared. The sampling time is $T = 1.5$ [s]. Each rows reports one robot velocity.

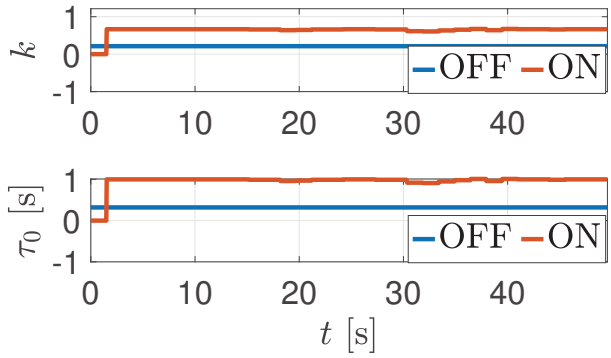


Figure 3.12. Representation of the time τ_0 and gain k computed for the two strategies.

4

Cooperative Aerial Load Transportation via Sampled Communication: non-square systems

Contents

In this Chapter we are interested to extend the analysis of the problem faced in Chapter 3 where we developed two strategies to implement point-stabilization control for a class of squared systems with a sparse kinematic structure. Similarly to the previous chapter we consider wireless communication and therefore sampled dynamics. Differently, in this work we study a class of systems characterized by a rectangular Jacobian matrix, which are defined non-square. In practice, this means that the number of robots is redundant, i.e. a desired system configuration can be obtained by different configurations of the robots, since the number of inputs to the system is greater than the number of degrees of freedom. We exploit the same architecture where a global motion planner is assumed to generate off-line a sequence of reference points for the system, taking into account obstacles and singularities avoidance. Then, a local motion planner generates the reference velocities for the robots to move from the current configuration to a desired one.

4.1 Introduction

In aerospace, vehicle, robot and other high tech industry, over-actuated systems are commonly used because of their advantage of increasing maneuverability and robust performance in fault situations. But there is some weakness: over-actuating a system provides a certain amount of redundancy for the controller, thus potentially allowing for recovery from off-nominal conditions. Due to this redundancy, it is necessary to compute a unique solution of a system of equation, which has more unknowns than the number of equations. Since such a system has infinite number of solutions, the problem is to find at least one that satisfies control input constraints and some additional optimization criterion. Our choice is to find a unique solution by selecting the closest feasible velocity to the desired one in case it is not admissible for the system: the strategy is based on the kinematics of the system, but we show that this corresponds to the behavior of the dynamical system when an unfeasible velocity is assigned to it. In detail, our contribution can be resumed in the following points:

- Firstly, we define a class of multi-agent systems with a sparse kinematic structure, characterized by a non-square Jacobian matrix: this corresponds to have an over or under-actuated system
- We exploit the same multi-layer architecture of the previous chapter, but this time it may happen that the reference velocities generated by the global planner are not feasible for the system, because of the over-actuation. Thus, we give a solution in this sense, but assigning to the system a desired velocity value that is a s close as possible to the reference one.
- We derive stability and convergence results in the continuous-time scenario and we give the condition such that the kinematic model is compatible with the behavior of the dynamical
- Every result is analysed both under the weighted 2-norm and ∞-norm: the weight is useful when the components of the configuration variables are expressed in different units of measure (angles and distances in our case); moreover, the ∞-norm is also taken into consideration because it is useful to express upper and lower bounds in a constrained problem
- We propose the design of the local planner for non-square systems both in the continuous and sampled-time scenarios: in particular, we give two procedures to compute the planner gain
- We validate our results both on the kinematic and dynamical model of the system
- Finally, we test the strategies in a far-from-ideal environment, where we use the dynamical model and simulate a wireless network instead of simplifying it with a sampling process.

4.1.1 Overview

The chapter is organized as follows: in Section 4.2 we derive the kinematic model of the class of non-square systems. Moreover, we introduce the problem that we want to solve. We also give some definitions which allows to define the scenario that we aim at studying. In Section 4.3 we show that the kinematic model arises naturally when applying to the mechanical system of interest a feedback force with the aim of tracking a desired profile of velocities. In Chapter 4.4 we define the local planner law in the continuous time scenario. We then study the planner in the sampled scenario in Section 4.5; we provide two strategies to compute the planner gain. Finally, in Section 4.6 we validate the theoretical results by means of simulations.

4.2 MODELING AND PROBLEM FORMULATION

4.2.1 Kinematics of multi-agent systems

In this section we recall the kinematic model of a multi-robot system composed by N robots which cooperate to fulfill a task: a more detailed description is given in Section 3.2. The robots do not communicate among themselves and receive information from a central unit, called *pivot* that we indicate with the letter V. It can be an object to be manipulated, a robot to be escorted or an other object in the space. The kinematic model, introduced in (3.1), is derived as in Appendix B.1.1:

$$\begin{bmatrix}\mathbf{p}_1 & \dots & \mathbf{p}_N\end{bmatrix}^\top = \mathbf{h}(\mathbf{q}) = \begin{bmatrix}\mathbf{h}^{(1)}(\mathbf{q}_1, \mathbf{q}_V) & \dots & \mathbf{h}^{(N)}(\mathbf{q}_N, \mathbf{q}_V)\end{bmatrix}^\top \tag{4.1}$$

This function maps the Lagrangian coordinates of the system $\mathbf{q} = [\mathbf{q}_1^\top \cdots \mathbf{q}_N^\top \ \mathbf{q}_V^\top]^\top \in \mathbb{R}^{m\times 1}, m > 0$ to the vector collecting the robots configurations $\mathbf{p} = \begin{bmatrix}\mathbf{p}_1^\top & \dots & \mathbf{p}_N^\top\end{bmatrix}^\top \in \mathbb{R}^{n\times 1}, n > 0$: $\mathbf{p}_i \in \mathbb{R}^{n_i \times 1}$ is the position of the i-th robot in the space, $\mathbf{q}_i \in \mathbb{R}^{m_i \times 1}, m_i > 0$ gathers the angles and/or distances between the pivot V and the i-th robot, and $\mathbf{q}_V \in \mathbb{R}^{m_V \times 1}, m_V > 0$ represents the pose (position and orientation) of the pivot itself.
The differential kinematics can be derived differentiating (4.1):

$$\dot{\mathbf{p}} = \mathbf{A}_\mathbf{q}\, \dot{\mathbf{q}},$$

where the Jacobian $\mathbf{A}_\mathbf{q} = \frac{\partial \mathbf{h}(\mathbf{q})}{\partial \mathbf{q}} \in \mathbb{R}^{n\times m}$ has the structure

$$\mathbf{A}_\mathbf{q} = \left[\begin{array}{ccc|c} \mathbf{A}_{\mathbf{q}_1}^{(1)} & & \mathbf{0} & \mathbf{A}_{\mathbf{q}_V}^{(1)} \\ & \ddots & & \vdots \\ \mathbf{0} & & \mathbf{A}_{\mathbf{q}_N}^{(N)} & \mathbf{A}_{\mathbf{q}_V}^{(N)} \end{array}\right], \tag{4.2}$$

and $\mathbf{A}_{\mathbf{q}_i}^{(i)} = \frac{\partial \mathbf{h}^i(\mathbf{q})}{\partial \mathbf{q}_i} \in \mathbb{R}^{n_i \times m_i}$ and $\mathbf{A}_{\mathbf{q}_V}^{(i)} = \frac{\partial \mathbf{h}^i(\mathbf{q})}{\partial \mathbf{q}_V} \in \mathbb{R}^{n_i \times m_V}$.

We conclude this section by introducing a parametrization of the configurations space that is compatible with the structural constraints (4.1). It allows to define the local planner law even when $n \neq m$ (non-square systems). Let us introduce the manifold $\mathcal{M} := \{(\mathbf{p},\mathbf{q}) \,|\, \mathbf{p} = \mathbf{h}(\mathbf{q})\}$ and its tangent space at a point $(\mathbf{p},\mathbf{q})$ as $T_{(\mathbf{p},\mathbf{q})}\mathcal{M} := \{(\dot{\mathbf{p}},\dot{\mathbf{q}}) \,|\, \dot{\mathbf{p}} = \frac{\partial \mathbf{h}}{\partial \mathbf{q}}\dot{\mathbf{q}}\}$. Under the assumption that $\frac{\partial \mathbf{h}}{\partial \mathbf{q}} = \mathbf{A}_{\mathbf{q}} \in \mathbb{R}^{n\times m}$ is full rank, then a possible parametrization of such space is given by:

$$T_{(\mathbf{p},\mathbf{q})}\mathcal{M} = \{(\mathbf{A}_{\mathbf{q}}\dot{\mathbf{q}},\dot{\mathbf{q}}), \forall \dot{\mathbf{q}} \in \mathbb{R}^m\}$$

i.e. the tangent space is a m-dimensional linear space embedded into an $(n+m)$-dimensional space.

4.2.2 Problem Formulation

As in Chapter 3 the goal of this chapter is to design a feedback-based local planner in order to steer the system from an initial configuration $\mathbf{q}^0$ to a desired one $\mathbf{q}^r$, where the latter could be a way point of a trajectory that is generated offline.

In this scenario, we assume that each agent has an inner-loop control sufficiently faster than the dynamics of the entire system, able to control the velocities of the variable vector $\mathbf{p}$ with the respect to a vector of desired velocities $\mathbf{u} = [\mathbf{u}_1^\top \cdots \mathbf{u}_N^\top]^\top \in \mathbb{R}^n$.

In presence of square systems, that is $n = m$, we might assume the velocities $\mathbf{p}$ to be fully controllable, i.e.:

$$\dot{\mathbf{p}}(t) = \mathbf{u}(t). \tag{4.3}$$

Indeed in this case the matrix $\mathbf{A}_{\mathbf{q}(t)}$, defined in (4.2), is a square matrix and the assigned $\mathbf{u}$ belongs to $\operatorname{Im}\mathbf{A}_{\mathbf{q}(t)} = \mathbb{R}^n$, provided that $\mathbf{A}_{\mathbf{q}(t)}$ is full rank. Instead, if $n > m$, then the system is over-actuated, that is there exist some unfeasible desired velocities which cannot be achieved by any configuration $\mathbf{q}$. Indeed, in this case, $\dim(\operatorname{Im}\mathbf{A}_{\mathbf{q}(t)}) = m < n$, so there exists some $\mathbf{u} \in \mathbb{R}^n$ which does not belong to $\operatorname{Im}\mathbf{A}_{\mathbf{q}(t)}$. On the other hand, if $n < m$ then the system is under-actuated, that is the number of control variables is strictly smaller than the controlled ones and not all the configurations $\mathbf{q}$ can be reached by choosing the input $\mathbf{u}$. Notice that this does not mean that the system cannot be controlled, but it is really unlikely.

In this work we are interested in the analysis of the over-actuated scenario where $n > m$, hence all the possible configurations $\mathbf{q}$ can be achieved by assigning a vector of desired vehicles velocities $\mathbf{u}$. On the other hand it may happen that $\mathbf{u} \notin \operatorname{Im}\mathbf{A}_{\mathbf{q}}$ and in this case we propose a solution such that the assigned velocities become feasible. More specifically, if $\mathbf{u} \in \operatorname{Im}\mathbf{A}_{\mathbf{q}(t)}$, that is, $\mathbf{u} = \mathbf{A}_{\mathbf{q}(t)}\dot{\mathbf{q}}(t)$, then $(\mathbf{u}(t),\dot{\mathbf{q}}(t)) \in T_{(\mathbf{p},\mathbf{q})}\mathcal{M}$; instead if $\mathbf{u} \notin \operatorname{Im}\mathbf{A}_{\mathbf{q}(t)}$ the idea is to find the closest feasible velocity to the desired one and this is obtained by projecting $\mathbf{u}$ onto $\operatorname{Im}\mathbf{A}_{\mathbf{q}(t)}$, see Fig.4.1. Formally, we are interested in computing the vector $\mathbf{u}_{\mathbf{q}}$ defined as follows:

$$\mathbf{u}_{\mathbf{q}}(t) := \arg \min_{\mathbf{v} \in \operatorname{Im}\mathbf{A}_{\mathbf{q}(t)}} \|\mathbf{u}(t) - \mathbf{v}\| = \Pi_{\mathbf{q}(t)}\mathbf{u}(t) \tag{4.4}$$

where $\Pi_{\mathbf{q}(t)} = \mathbf{A}_{\mathbf{q}(t)}\mathbf{A}^{\dagger}_{\mathbf{q}(t)} \in \mathbb{R}^{n\times n}$ is the orthogonal projector onto $\operatorname{Im}\mathbf{A}_{\mathbf{q}(t)}$ being $\mathbf{A}^{\dagger}_{\mathbf{q}(t)}$

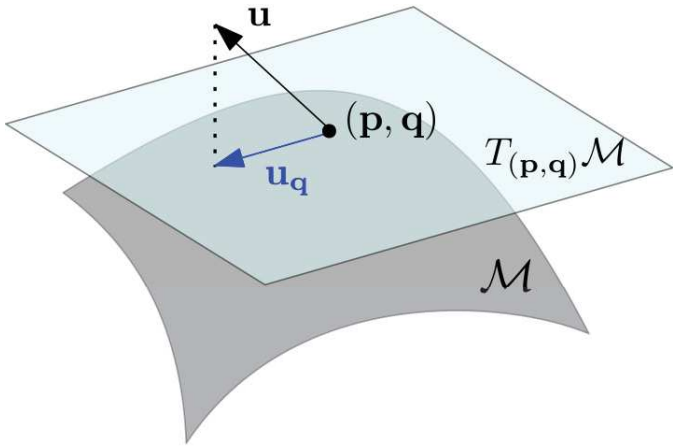


Figure 4.1. Manifold of structural constraints and tangent space for projection.

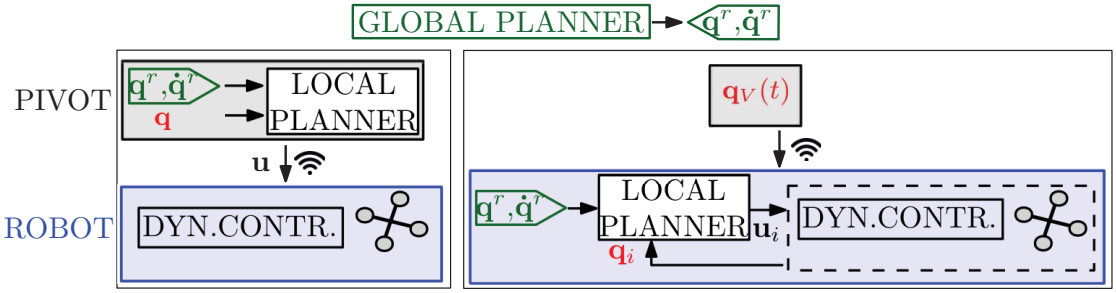


Figure 4.2. Representation of (4.7) (right) and (4.8) (left) where the gray area represents the load and the blue one an agent (i.e. dynamical controller plus robot). $\mathbf{q}^r$ is generated by the global planner (in green), the red labels are the measurements from the sensors. The dynamical controller converts $\mathbf{u}$ into forces for the robot. The wireless symbol indicates that the quantity can be transmitted via sampled communication.

the Moore-Penrose inverse of $\mathbf{A}_{\mathbf{q}(t)}$, i.e.,

$$\mathbf{A}^{\dagger}_{\mathbf{q}(t)} = (\mathbf{A}^{\top}_{\mathbf{q}(t)}\mathbf{A}_{\mathbf{q}(t)})^{-1}\mathbf{A}^{\top}_{\mathbf{q}(t)}.$$

It turns out that the velocities of vector $\mathbf{p}$ satisfy

$$\dot{\mathbf{p}} = \mathbf{u}_{\mathbf{q}}(t) = \Pi_{\mathbf{q}(t)}\mathbf{u}(t); \tag{4.5}$$

equivalently, if $\mathbf{A}_{\mathbf{q}(t)}$ is full rank, we have

$$\dot{\mathbf{q}} = \mathbf{A}^{\dagger}_{\mathbf{q}(t)}\mathbf{u}(t). \tag{4.6}$$

Note that if $n \leq m$, than $\Pi_{\mathbf{q}(t)} = \mathbf{I}_n$, hence no projection is needed since the space of all the possible values of $\mathbf{u}(t)$ lies is a subset of the space Im $\mathbf{A}_{\mathbf{q}(t)}$; this scenario includes the case $n = m$ that was already described in Chapter 3, where the space of the inputs $\mathbf{u}(t)$ is exactly equal to Im $\mathbf{A}_{\mathbf{q}(t)} = \mathbb{R}^n$. We consider two possible architectures, depicted in Fig.4.2 and B.1.:

$$\mathbf{u}_i(t) = \kappa^s_i(\mathbf{q}_i(t), \mathbf{q}_V(t); \mathbf{q}^r_i, \mathbf{q}^r_V), \quad i = 1, \ldots, N \tag{4.7}$$

$$\mathbf{u}_i(t) = \kappa^c_i(\mathbf{q}(t); \mathbf{q}^r), \quad i = 1, \ldots, N \tag{4.8}$$

where the former tries to maintain the same sparsity of the Jacobian while the latter exploit the full knowledge of the vector $\mathbf{q}$[1]. We will show that, if $\mathbf{q}(t)$ is continuously accessible to the local planner, then (4.7) is sufficient to drive the system to $\mathbf{q}^0$; otherwise, if $\mathbf{q}(t)$ is available at sampled time instants, then the two architectures give rise to two different strategies with different performance and computational requirements.

Before proceeding, for $R > 0$, we define the sets

$$\mathcal{B}_2(\mathbf{q}^r) = \{\mathbf{q} \in \mathbb{R}^m \,|\, \|\mathbf{q} - \mathbf{q}^r\|_{\boldsymbol{D},2} < R\}$$
$$\mathcal{B}_\infty(\mathbf{q}^r) = \{\mathbf{q} \in \mathbb{R}^m \,|\, \|\mathbf{q} - \mathbf{q}^r\|_{\boldsymbol{D},\infty} < R\}$$

where $\boldsymbol{D}$ is a weighting matrix such that $\boldsymbol{D} = \operatorname{diag}\{\boldsymbol{D}_1, \dots, \boldsymbol{D}_N, \boldsymbol{D}_V\}$, with $\boldsymbol{D}_i = \frac{1}{d_i^2}\mathbf{I}_{m_i}$ and $\boldsymbol{D}_V = \frac{1}{d_V^2}\mathbf{I}_{m_V}$, and where

$$\|\mathbf{q} - \mathbf{q}^r\|_{\boldsymbol{D},2} = \|\boldsymbol{D}(\mathbf{q} - \mathbf{q}^r)\|_2 = \sqrt{\sum_i \frac{\|q_i - q_i^r\|^2}{d_i^2}}$$
$$\|\mathbf{q} - \mathbf{q}^r\|_{\boldsymbol{D},\infty} = \|\boldsymbol{D}(\mathbf{q} - \mathbf{q}^r)\|_\infty = \max_i \frac{\|q_i - q_i^r\|_\infty}{d_i^2}$$

are respectively the 2-norm and ∞-norm, weighted with $\boldsymbol{D}$. Without loss of generality we make the following assumption on the weights of matrix $\boldsymbol{D}$,

$$\min\{d_1, \dots, d_N, d_V\} \geq 1. \tag{4.9}$$

This allows to derive some stability related properties in Section 4.5.1. From now on, for the sake of notational convenience, by the symbol $\mathcal{B}_*(\mathbf{q}^r)$ we denote interchangeably either the set $\mathcal{B}_2(\mathbf{q}^r)$ or the set $\mathcal{B}_\infty(\mathbf{q}^r)$.

We conclude this section with the following assumption.

Assumption 4.2.1. *The following relations hold:*

1. *the matrix $\mathbf{A}_\mathbf{q}$ is full-rank and the map $\mathbf{h}$ is twice continuously differentiable for all $q \in \mathcal{B}_*(\mathbf{q}^r)$. In addition, these properties can be extended by continuity on the closure of such set, defined as $\overline{\mathcal{B}}_*(\mathbf{q}^r)$;*

2. *$\mathbf{q}^0 \in \mathcal{B}_*(\mathbf{q}^r)$.*

4.3 Physical interpretation of the projection operator

The goal of this section is to provide a physical interpretation of the projection step in (4.4). In particular we want to show that the kinematic model in (4.6) arises naturally when applying to the mechanical system of interest a feedback force with the aim of tracking a desired profile of velocities $\mathbf{u}(t)$.

[1] Note that we do not pursue control strategies of type $\mathbf{u}(t) = \kappa(\mathbf{p}(t); \mathbf{p}^r)$ where $\mathbf{p}^r := \mathbf{h}(\mathbf{q}^\mathbf{r})$ since the function $\mathbf{h}$ might not be perfectly known and avoidance of singular configurations cannot be guaranteed.

To do that, let us consider the dynamical model

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{J}^{\top}(\mathbf{q})\mathbf{F} \tag{4.10}$$

where $\mathbf{M}(\mathbf{q})$ is the joint-space inertia matrix, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$ is the Coriolis and centripetal coupling matrix, $\mathbf{g}(\mathbf{q})$ is the gravity term, and $\mathbf{J}^{\top}(\mathbf{q})\mathbf{F}$ describes the effect of the external forces applied to the system, being $\mathbf{J}(\mathbf{q}) = \mathbf{A}_{\mathbf{q}}$ the Jacobian matrix.
Without loss of generality, we can neglect $\mathbf{g}(\mathbf{q})$ since a gravity compensation action can be implemented by the controller. Hence (4.10) becomes

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} = {\mathbf{A}_{\mathbf{q}}}^{\top}\mathbf{F}$$

Let us assume to apply an external force $\mathbf{F}$ to the robot of the type

$$\mathbf{F} := -\alpha(\dot{\mathbf{p}} - \mathbf{u}) \tag{4.11}$$

where $\alpha > 0$ is a constant gain, $\dot{\mathbf{p}}$ is the current velocity vector and $\mathbf{u} \in \mathbb{R}^n$ is the vector of desired velocities.

Then the dynamical model can be rewritten as

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} = -\alpha{\mathbf{A}_{\mathbf{q}}}^{\top}\mathbf{A}_{\mathbf{q}}\dot{\mathbf{q}} - \alpha{\mathbf{A}_{\mathbf{q}}}^{\top}\mathbf{u}$$

or equivalently

$$\epsilon\ddot{\mathbf{q}} = \mathbf{M}(\mathbf{q})^{-1}\Big(-{\mathbf{A}_{\mathbf{q}}}^{\top}\mathbf{A}_{\mathbf{q}}\dot{\mathbf{q}} + \mathbf{A}_{\mathbf{q}}^{\top}\mathbf{u} - \epsilon\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}\Big) \tag{4.12}$$

where $\epsilon = 1/\alpha$. Observe that for small ϵ we have that

$$-{\mathbf{A}_{\mathbf{q}}}^{\top}\mathbf{A}_{\mathbf{q}}\dot{\mathbf{q}} + \mathbf{A}_{\mathbf{q}}^{\top}\mathbf{u} \approx 0,$$

that is,

$$\dot{\mathbf{q}} \approx \mathbf{A}^{\dagger}_{\mathbf{q}(t)}\mathbf{u},$$

which is the kinematic model in (4.6). Notice that this result can be seen in a multi time-scale scenario, where for small values of ϵ the dynamics of the system is fast enough such that it achieves $\dot{\mathbf{p}} \approx \mathbf{u}$ before the next reference velocity $\mathbf{u}$ is assigned. Otherwise, for larger values of the parameter, the time-scale separation is no more achieved and the stability of the controlled system is no more guaranteed.

In order to practically show the meaning of this result, we choose the system depicted in Figure 4.3a where four aerial robots transport a payload. We define the Lagrangian variables as in Section 4.2.1, hence we have $\mathbf{q} = [q_1\ q_2\ q_3\ q_4\ \mathbf{q}_V]^{\top} \in \mathbb{R}^{10}$ where $q_i \in \mathbb{R}$ are the angles of the cables with respect to the load and $\mathbf{q}_V \in \mathbb{R}^6$ is the load pose. We derive the dynamical model of the system following the procedure in Appendix B.1.2. The initial configuration is $\mathbf{q}_0$ and we define $\dot{\mathbf{q}}_0$ such that the load moves upwards on the axis z and the angles q_i increase with velocity $\dot{q}_i(0)$, as shown in Figure 4.3b. This velocity is then transformed into velocities for the robots with $\mathbf{u} = \mathbf{A}_{\mathbf{q}_0}\dot{\mathbf{q}}_0$ and the vector $\mathbf{u}$ is then converted into forces to be applied to the cables ends through equation (4.11). At the beginning $\mathbf{u} \in \operatorname{Im}\mathbf{A}_{\mathbf{q}}$, hence it is feasible; then, as the system dynamics evolves, it happens that $\mathbf{u} \notin \operatorname{Im}\mathbf{A}_{\mathbf{q}}$ because the angles q_i keep increasing becoming even greater

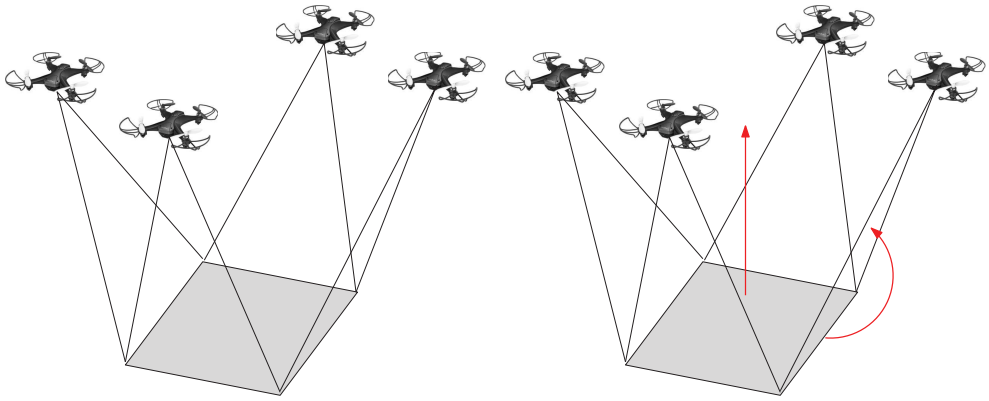


Figure 4.3. Example of a rectangular multi-agent system where four aerial robots transport a payload. On the right, a movement of the system is depicted to give a better intuition of the plot in Figure 4.4.

than $\frac{\pi}{2}$. In Figure 4.4 you can see what happens in this scenario, that is when unfeasible velocities are given as references to the system: the actual robots velocities $\dot{\mathbf{p}}$ behave like $\mathbf{u_q}$ in (4.4), stopping tracking the vector $\mathbf{u}$ as soon as it becomes unfeasible.

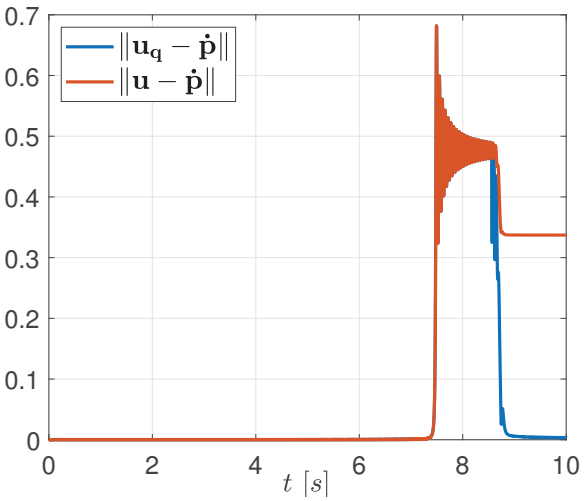


Figure 4.4. Behavior of the robots velocities $\dot{\mathbf{p}}$ when the reference ones $\mathbf{u}$ are unfeasible: in practice $\dot{\mathbf{p}} \to \mathbf{u_q}$ where $\mathbf{u_q}$ are defined in (4.4).

4.4 Local planner: continous-time

In this section we describe the local planner that generates the desired vehicles velocities in the scenario of continuous time measurements. The goal is to steer $\mathbf{q}(t)$ to $\mathbf{q}^r$.

As introduced in Sec. 4.2.2 if $\mathbf{u} \notin \operatorname{Im} \mathbf{A}_{\mathbf{q}(t)}$, then we choose $\mathbf{u}(t)$ as the smallest norm velocity that is compatible with (4.1); hence, the dynamics of $\mathbf{q}(t)$ given these reference velocities is:

$$\dot{\mathbf{q}}(t) = \arg \min_{\mathbf{v} \in \mathbb{R}^m} \{\|\mathbf{v}\| \,|\, \Pi_{\mathbf{q}(t)}\mathbf{u}(t) = \mathbf{A}_{\mathbf{q}(t)}\mathbf{v}\} = \mathbf{A}^{\dagger}_{\mathbf{q}(t)}\mathbf{u}(t) \tag{4.13}$$

Note that according to this modeling the velocities $(\dot{\mathbf{p}}, \dot{\mathbf{q}})$ are compatible with the tangent manifold, i.e.

$$(\Pi_{\mathbf{q}(t)}\mathbf{u}(t), \mathbf{A}^{\dagger}_{\mathbf{q}(t)}\mathbf{u}(t)) \in T_{(\mathbf{p},\mathbf{q})}\mathcal{M}$$

A possible choice to define the velocities $\mathbf{u}$ is

$$\mathbf{u}(t) = -\mathbf{A}_{\mathbf{q}(t)}\mathbf{K}(\mathbf{q}(t) - \mathbf{q}^r) \tag{4.14}$$

where $\mathbf{K}$ is a gain matrix to be designed. Note that according to (4.14), the dynamics of $\mathbf{q}(t)$ evolves according to

$$\dot{\mathbf{q}}(t) = -\mathbf{A}^{\dagger}_{\mathbf{q}(t)}\mathbf{A}_{\mathbf{q}(t)}\mathbf{K}(\mathbf{q}(t) - \mathbf{q}^r) = -\mathbf{K}(\mathbf{q}(t) - \mathbf{q}^r) \tag{4.15}$$

The feedback gain $\mathbf{K}$ must guarantee stability, possibly maintain the decentralized structure induced by $\mathbf{A}_{\mathbf{q}(t)}$, and to make the trajectory $\mathbf{q}(t)$ not to pass through a singularity.

In this paper, we assume that the gain matrix has the form $\mathbf{K} = k \cdot \overline{\mathbf{K}}$ where k is a positive scalar gain, i.e., $k > 0$, and $\overline{\mathbf{K}}$ is a diagonal matrix with the following structure

$$\overline{\mathbf{K}} = \operatorname{diag}\{\mathbf{K}_1, \ldots, \mathbf{K}_N, \mathbf{K}_V\},$$

where $\mathbf{K}_i = k_i \mathbf{I}_{m_i}$, $\mathbf{K}_V = k_V \mathbf{I}_{m_V}$ being $k_i, k_V > 0$.

We have the following result.

Proposition 4.4.1. *Consider system described by (4.13) and (4.14), where $\mathbf{K}$ is defined as above. Assume Assumption 4.2.1 holds true. Then, for all $(N+2)$-upla of positive real numbers $k, k_1, \ldots, k_N, k_V$, we have that*

1. *for all $t \geq 0$, $\mathbf{q}(t) \in \mathcal{B}_*(\mathbf{q}^r)$;*
2. *the trajectory $\mathbf{q}(t)$ converges exponentially fast to $\mathbf{q}^r$.*

Proof. 1. Consider the Lyapunov function $V(\mathbf{q}) = \|\mathbf{q} - \mathbf{q}^r\|^2_*$, then it is easy to see that $\dot{V}(\mathbf{q}) < 0$ for every $\mathbf{q} \neq \mathbf{q}^r$ by following the same reasoning as in the proof of Proposition 3.3.1.

2. From (4.15) and since $\mathbf{K}$ is diagonal with all positive entries, then it is straightforward to conclude that $\mathbf{q}(t)$ converges exponentially fast to $\mathbf{q}^r$.

■

Notice that, according to the structure of the matrix $\mathbf{K}$, a different weight is assigned to each $\mathbf{q}_i$, modifying the corresponding convergence rate. For instance, the choice $\overline{\mathbf{K}} = \text{diag}\{\mathbf{I}_{m_1}, \dots, \mathbf{I}_{m_N}, k_V \mathbf{I}_{m_V}\}$ with $k_V > 1$ allows for a faster convergence of $\mathbf{q}_V$ w.r.t. the other variables.

Interestingly, the exponential stability property established in the previous Proposition for the kinematic model (4.13) when the control input (4.14) is adopted, holds true also for the dynamic model in (4.12) provided the gain α is sufficiently high. This fact is formally stated in the next theorem.

Theorem 4.4.1. *Consider the dynamic model (4.12) with $\mathbf{F}$ given in (4.11) and $\mathbf{u}$ defined as in (4.14). There exist $\overline{\alpha} > 0$ and a neighborhood of $\mathbf{q}^r$, say $\bar{\mathcal{B}}(\mathbf{q}^r)$, contained in $\mathcal{B}_*(\mathbf{q}^r)$, i.e., $\bar{\mathcal{B}}(\mathbf{q}^r) \subseteq \mathcal{B}_*(\mathbf{q}^r)$, such that, if $\mathbf{q}(0) \in \bar{\mathcal{B}}(\mathbf{q}^r)$ and $\alpha > \bar{\alpha}$, then*

1. *for all $t \geq 0$, $\mathbf{q}(t) \in \bar{\mathcal{B}}(\mathbf{q}^r)$;*
2. *the trajectory $\mathbf{q}(t)$ converges exponentially fast to $\mathbf{q}^r$.*

The proof of the Theorem is reported in Appendix B.3.

4.5 Local Planner: Sampled Measurements

We now consider the evolution of (4.13) under sampled dynamics, that is we assume that $\mathbf{q}$ is measured at the time instants hT, $h = 0, 1, 2, \dots$ where T is the sampling time. Moreover, we assume that the vehicles reference velocity $\mathbf{u}(t)$ is kept constant within a time window T using (4.14):

$$\mathbf{u}(t) = \mathbf{u}_h = -k \mathbf{A}_{\mathbf{q}_h} \overline{\mathbf{K}} (\mathbf{q}_h - \mathbf{q}^r), \qquad hT \leq t < (h+1)T,$$

for $h = 1, 2, \dots$, and where we defined $\mathbf{q}_h = \mathbf{q}(hT)$. In this scenario, assuming $A_{\mathbf{q}(t)}$, $t \geq 0$ full rank, the evolution of $\mathbf{q}(t)$ becomes:

$$\dot{\mathbf{q}}(t) = -k A^{\dagger}_{\mathbf{q}(t)} A_{\mathbf{q}_h} \overline{\mathbf{K}} (\mathbf{q}_h - \mathbf{q}^r), \;\; hT \leq t < (h+1)T. \tag{4.16}$$

In this section we assume that $\overline{\mathbf{K}}$ is assigned a-priori (or, equivalently, that the values of k_V and k_i, $i = 1, \dots, N$, are assigned a-priori), while k is a positive parameter to be designed, possibly time varying, i.e., $k = k_h$, such that the stability of the system is still guaranteed.

The approach we propose for the design of k within each interval $[hT, (h+1)T]$, is based on the analysis of the following auxiliary system whose solution is characterized by interesting and useful properties that will be analyzed later on

$$\begin{aligned} \dot{\mathbf{e}}'(\tau; \mathbf{e}_h) &= -A^{\dagger}_{\mathbf{q}^r + \mathbf{e}'(\tau; \mathbf{e}_h)} A_{\mathbf{q}^r + \mathbf{e}_h} \overline{\mathbf{K}} \, \mathbf{e}_h =: \mathbf{f}(\mathbf{e}'(\tau; \mathbf{e}_h)) \\ \mathbf{e}'(0; \mathbf{e}_h) &= \mathbf{e}_h \end{aligned} \tag{4.17}$$

where $\mathbf{e}_h$ is such that $\mathbf{q}^r + \mathbf{e}_h \in \mathcal{B}_{2,\infty}(\mathbf{q}^r)$ and $\mathbf{e}'(\,\cdot\,;\,\cdot\,) \in \mathbb{R}^m$.

Let $\mathbf{e}(t) := \mathbf{q}(t) - \mathbf{q}^r$. Note that by direct inspection we have

$$\mathbf{e}(t) = \mathbf{e}'(k(t-hT);\mathbf{e}_h), \qquad hT \le t < (h+1)T. \tag{4.18}$$

Hence, once the solution $\mathbf{e}'(\tau;\mathbf{e}_h)$ is computed, then $\mathbf{e}(t)$ can be obtained through shifting by hT and rescaling by k, i.e., as long as $\mathbf{e}'(k(t-hT);\mathbf{e}_h)$ exists; then, $\mathbf{q}(t) = \mathbf{e}(t) + \mathbf{q}^r$. The major benefit of this approach is that the analysis of (4.17) is independent of the gain k and the sampling period T.

Remark 4.5.1. In the following the symbol $\|\cdot\|$ indicates that the relation where it is used holds both for $\|\cdot\|_{\mathbf{D},2}$ and $\|\cdot\|_{\mathbf{D},\infty}$ for the equivalence of norms. The only difference could be some inequality coefficients.

From Ass. 4.2.1 the following properties hold true for all $\mathbf{e}$, $\mathbf{e}'$ such that $\mathbf{q}^r + \mathbf{e}$, $\mathbf{q}^r + \mathbf{e}'$ belong to $\mathcal{B}_*(\mathbf{q}^r)$

$$\mathbf{f}(\mathbf{e}'(0;\mathbf{e})) = -\overline{\mathbf{K}}\mathbf{e}, \tag{4.19}$$

$$\|\mathbf{f}(\mathbf{e}'(\tau);\mathbf{e})\| \le \|A^{\dagger}_{\mathbf{q}^r+\mathbf{e}'} A_{\mathbf{q}^r+\mathbf{e}} \overline{\mathbf{K}}\| \|\mathbf{e}\| = a\|\mathbf{e}\|, \tag{4.20}$$

$$\left\| \frac{\partial \mathbf{f}(\mathbf{e}'(\tau);\mathbf{e})}{\partial \mathbf{e}'} \right\| \le b\|\mathbf{e}\|, \quad , \tag{4.21}$$

for some $a, b > 0$ since $\mathbf{f}$ and $\frac{\partial \mathbf{f}}{\partial \mathbf{e}'}$ are continuous maps on a compact domain.

Since the flow $\mathbf{f}(\mathbf{e}';\mathbf{e}_h)$ is locally continuously differentiable in $\mathbf{e}'$, then for each $\mathbf{e}_h$ such that $\mathbf{q}^r + \mathbf{e}_h \in \mathcal{B}_{2,\infty}(\mathbf{q}^r)$, there exists $\delta(\mathbf{e}_h) > 0$ s.t. (4.17) has a unique solution $\mathbf{e}'(\tau;\mathbf{e}_h)$ for $\tau \in [0, \delta(\mathbf{e}_h))$. Without loss of generality, let us define $\tau_{max}(\mathbf{e}_h)$ the maximum time extension for which the unique solution $\mathbf{e}'(\tau;\mathbf{e}_h)$ exists (see Khalil (2002)) for $0 \le \tau < \tau_{max}(\mathbf{e}_h)$. The following Proposition shows an interesting property of the solution $\mathbf{e}'(\tau;\mathbf{e}_h)$.

Proposition 4.5.1. *Consider the dynamical system (4.17) where we assume $\mathbf{e}_h + \mathbf{q}^r \in \mathcal{B}_*(\mathbf{q}^r)$. Then the solution $\mathbf{e}'(\tau;\mathbf{e}_h)$ satisfies one of these two properties:*

(i) for all $\tau > 0$, it holds $\|\mathbf{e}'(\tau;\mathbf{e}_h)\| < \|\mathbf{e}_h\|$ and in such case $\tau_{max}(\mathbf{e}_h) = \infty$;

(ii) there exists $0 < \bar{\tau} < \tau_{max}(\mathbf{e}_h)$ such that $\|\mathbf{e}'(\bar{\tau};\mathbf{e}_h)\| = \|\mathbf{e}_h\|$ and $\|\mathbf{e}'(\tau;\mathbf{e}_h)\| < \|\mathbf{e}_h\|$ for all $0 < \tau < \bar{\tau}$.

Proof. Only two scenarios are possible, either $\|\mathbf{e}'(\tau;\mathbf{e}_h)\| < \|\mathbf{e}_h\|$ for all $\tau > 0$, from which $\tau_{max}(\mathbf{e}_h) = \infty$ follows, or not. If not, then $\overline{\tau} := \inf_{\tau>0}\{\tau \,|\, \|\mathbf{e}'(\tau;\mathbf{e}_h)\| \ge \|\mathbf{e}_h\|\}$ is well defined and finite. Now, let us assume that $\bar{\tau} = 0$: since $\mathbf{e}'(\tau;\mathbf{e}_h)$ is continuously differentiable in τ and since $\dot{\mathbf{e}}'(0;\mathbf{e}_h) = -\overline{\mathbf{K}}\mathbf{e}_h$, then there exists $\dot{\mathbf{e}}'(0;\mathbf{e}_h)$ s.t. $\mathbf{e}_h^\top \dot{\mathbf{e}}'(0;\mathbf{e}_h) \ge 0 \Leftrightarrow -\mathbf{e}_h^\top \overline{\mathbf{K}}\mathbf{e}_h \le 0$ that is a contradiction. This implies that $\bar{\tau} > 0$. Finally, observe that the definition of $\bar{\tau}$ makes sense only if a solution $\mathbf{e}'(\tau;\mathbf{e}_h)$ exists, hence $\overline{\tau} \le \tau_{max}(\mathbf{e}_h)$. ∎

Based on the previous Proposition, we can introduce the following temporal variables

$$\tau_s(\mathbf{e}_h) := \min_{\tau}\{\tau > 0 \,|\, \|\mathbf{e}'(\tau;\mathbf{e}_h)\| = \|\mathbf{e}_h\|\}$$
$$\tau_o(\mathbf{e}_h) := \operatorname*{arginf}_{0 \le \tau \le \tau_s(\mathbf{e}_h)} \|\mathbf{e}'(\tau;\mathbf{e}_h)\|,$$

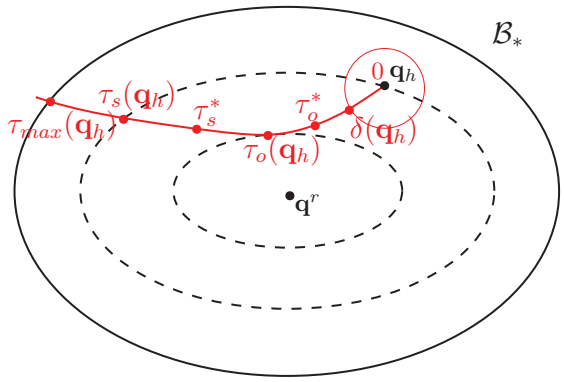


Figure 4.5. Representation of $\mathcal{B}_*$, $\delta_{\mathbf{e}_h}$, τ_o^*, $\tau_o(\mathbf{e}_h)$, τ_s^*, $\tau_s(\mathbf{e}_h)$ and $\tau_{max}(\mathbf{e}_h)$. The red labels represent time values, while the black ones are points or sets in $\mathbb{R}^m$.

where $\tau_s(\mathbf{e}_h) = \infty$, if $\|\mathbf{e}'(\tau;\mathbf{e}_h)\| < \|\mathbf{e}_h\|$ for all $\tau > 0$. Basically, $\tau_s(\mathbf{e}_h)$ represents the first time that the solution $\mathbf{e}'(\tau;\mathbf{e}_h)$ hits the boundary of the ball centered at the origin and passing though the initial condition $\mathbf{e}_h$, , while $\tau_o(\mathbf{e}_h)$, represents the time that $\mathbf{e}'(\tau;\mathbf{e}_h)$ is closest to the origin.

Based on $\tau_s(\mathbf{e}_h)$ and $\tau_o(\mathbf{e}_h)$, we can define also

$$\tau_s^* := \inf\{\tau_s(\mathbf{e}_h) \,|\, \mathbf{q}^r + \mathbf{e}_h \in \mathcal{B}_*(\mathbf{q}^r)\} \tag{4.22}$$

and, analogously,

$$\tau_o^* := \inf\{\tau_o(\mathbf{e}_h) \,|\, \mathbf{q}^r + \mathbf{e}_h \in \mathcal{B}_*(\mathbf{q}^r)\}. \tag{4.23}$$

All the above quantities are graphically sketched in Fig.4.5.

The following result provides a further interesting characterization of the trajectories $\mathbf{e}'(\tau;\mathbf{e}_h)$.

Proposition 4.5.2. *There exist* $\bar{\tau}_o$ *and* $0 \le \rho < 1$ *such that, for all* $\mathbf{e}_h$ *s.t.* $\mathbf{q}^r + \mathbf{e}_h \in \mathcal{B}_*(\mathbf{q}^r)$*, it holds*

$$0 < \bar{\tau}_o \le \tau_o^* \le \infty \quad \textit{and} \quad \|\mathbf{e}'(\bar{\tau}_o;\mathbf{e}_h)\| \le \rho\|\mathbf{e}_h\|.$$

The proof is constructive and it is provided in the next subsection where we show how to numerically compute a pair $(\bar{\tau}_o, \rho)$ satisfying these relations. It is worth stressing that $\bar{\tau}_o$ is independent of $\mathbf{e}_h$, but, as we will see later on, it is possibly dependent on $\mathcal{B}_*(\mathbf{q}^r)$.

The variables $\bar{\tau}_o$ and $\tau_o(\mathbf{e}_h)$ allow to propose two different strategies to design the gain k. The first is based on the observation that if $k = \frac{\bar{\tau}_o}{T}$, then $\mathbf{e}'(\tau;\mathbf{e}_h) \to \mathbf{0} \;\; \forall \mathbf{e}_h$ at a convergence rate ρ. In fact, from (4.18) we have $\|\mathbf{e}_{h+1}\| = \|\mathbf{e}'(kT;\mathbf{e}_h)\| = \|\mathbf{e}'(\bar{\tau}_o;\mathbf{e}_h)\| \le \rho\|\mathbf{e}_h\|$. This suggests an *offline procedure* to select k that will be described in the next subsection 4.5.1. However, based on the definition of $\tau_o(\mathbf{e}_h)$, it might be likely that $\|\mathbf{e}'(\tau_o(\mathbf{e}_h);\mathbf{e}_h)\| < \|\mathbf{e}'(\tau_o^*;\mathbf{e}_h)\|$ for most $\mathbf{e}_h$. Therefore, an alternative approach is to select k at each instant h such that $k_h := \frac{\tau_o(\mathbf{e}_h)}{T}$. This idea suggests an *online strategy* that will be described in subsection 4.5.2.

4.5.1 Off-line procedure (Stability and convergence rate)

We start this section by observing that, if $\tau_s^* > 0$, then the original system (4.16) is asymptotically stable for

$$kT < \tau_s^*.$$

Note that if this condition is not satisfied, then we can find a time instant $\tau' \in [\tau_s^*, +\infty]$ and $\mathbf{e}(0)$ such that $\|\mathbf{e}'(\tau'; \mathbf{e}(0))\| > \|\mathbf{e}'(0; \mathbf{e}(0))\|$, that is $\|\mathbf{e}(kT)\| > \|\mathbf{e}(0)\|$ for $k = \frac{\tau'}{T}$. Although this does not imply instability of the whole trajectory, it is an undesired behavior.

To show that $\tau_s^* > 0$, we next provide an explicit lower bound $\bar{\tau}_s > 0$ for τ_s^*. To do that, we consider an expansion of the solution of (4.17) and numerically estimate the upper bound of the approximation error via an additional parameter μ. This allows to derive an analytical expression of $\bar{\tau}_s$.We recall that the solution of (4.17) can also be written as:

$$\mathbf{e}'(\tau; \mathbf{e}_h) = \mathbf{e}_h + \int_0^\tau \mathbf{f}(\mathbf{e}'(\tau'; \mathbf{e}_h))d\tau', \quad 0 \le \tau < \tau_s(\mathbf{e}_h).$$

By using Taylor's theorem for multivariate functions with integral form of the remainder, it becomes

$$\begin{aligned}\mathbf{e}'(\tau; \mathbf{e}_h) &= \mathbf{e}_h + \tau\, \mathbf{f}(\mathbf{e}'(0; \mathbf{e}_h)) + \\ &\quad + \tau^2 \int_0^1 (1-\epsilon) \frac{\partial \mathbf{f}(\mathbf{e}'(\epsilon\tau; \mathbf{e}_h))}{\partial \mathbf{e}'} \mathbf{f}(\mathbf{e}'(\epsilon\tau; \mathbf{e}_h)) d\epsilon \\ &= (\mathbf{I} - \tau\overline{\mathbf{K}})\mathbf{e}_h + \tau^2 \mathbf{d}(\tau, \mathbf{e}_h), \quad 0 \le \tau < \tau_s(\mathbf{e}_h), \end{aligned} \tag{4.24}$$

where the reminder $\mathbf{d}$ has the property[2]:

Proposition 4.5.3. *For all $\mathbf{e}_h$, such that $\mathbf{q}^r + \mathbf{e}_h \in \mathcal{B}_*(\mathbf{q}^r)$ and for all $0 < \tau < \tau_s(\mathbf{e}_h)$ it holds $\|\mathbf{d}(\tau, \mathbf{e}_h)\| \le \frac{1}{2} ab \|\mathbf{e}_h\|^2$, where a, b are the bounding constants introduced in (4.20) and (4.21).*

Proof. As a consequence of the properties (4.20) and (4.21), the reminder in (4.24) becomes

$$\begin{aligned}&\|\mathbf{d}(\tau, \mathbf{e}_h)\|_{\mathbf{D},2} \le \\ &\le \int_0^1 (1-\epsilon) \left\| \frac{\partial \mathbf{f}(\mathbf{e}'(\epsilon\tau); \mathbf{e}_h)}{\partial \mathbf{e}'} \right\|_{\mathbf{D},2} \|\mathbf{f}(\mathbf{e}'(\epsilon\tau); \mathbf{e}_h)\|_{\mathbf{D},2} d\epsilon \\ &\le \int_0^1 (1-\epsilon) ab \|\mathbf{e}_h\|_{\mathbf{D},2}^2 d\epsilon = \frac{1}{2} ab \|\mathbf{e}_h\|_{\mathbf{D},2}^2 \end{aligned}$$

By equivalence of norms the same relation holds for $\|\cdot\|_{\mathbf{D},\infty}$. ∎

Notice that since $\mathbf{e}_h$ is such that $\mathbf{q}^r + \mathbf{e}_h \in \mathcal{B}_*(\mathbf{q}^r)$, then $\|\mathbf{e}_h\| \le r$ and, hence, by

[2] Note that the reminder scales as $\|\mathbf{e}_h\|^2$ which implies that as $\mathbf{e}'$ becomes closer to the origin, the faster it converges.

defining,

$$\mu := \frac{1}{2} abr, \tag{4.25}$$

we can write $\|\mathbf{d}(\tau, \mathbf{e}_h)\| \leq \mu \|\mathbf{e}_h\|$. Note that μ, as defined in (4.25), represents a rough estimate of the upper bound of $\|\mathbf{d}(\tau, \mathbf{e}_h)\|$. However this estimate can be refined as follows. Let

$$\mathbf{d}'(\tau, \mathbf{e}_h) := \tau^2 \mathbf{d}(\tau, \mathbf{e}_h) \overset{(4.24)}{=} \mathbf{e}'(\tau, \mathbf{e}_h) - (\mathbf{I} - \tau \overline{\mathbf{K}}) \mathbf{e}_h,$$

and

$$\mu^* := \inf_{\gamma} \{\gamma \mid \|\mathbf{d}'(\tau, \mathbf{e}_h)\| \leq \gamma \|\mathbf{e}_h\| \tau^2, \quad \forall \mathbf{e}_h,\ \forall \tau \in (0, \tau_s(\mathbf{e}_h))\}. \tag{4.26}$$

We are interested in providing an estimate $\hat{\mu}^*$ of μ^*. To do that, we adopt a Monte-Carlo sampling method. Specifically, given the positive integer number N, we randomly pick samples $\mathbf{e}_h^i$, $i = 1, \ldots, N$, such that $\mathbf{q}^r + \mathbf{e}_h \in \mathcal{B}_*(\mathbf{q}^r)$ for all $1 \leq i \leq N$, and for each of them, we simulate $\mathbf{e}'(\tau_i; \mathbf{e}_h^i)$ for $\tau_i \in [0, \tau_s(\mathbf{e}_h^i)]$.

$$\hat{\mu}^* := \max_i \left\{ \frac{\|\mathbf{d}'(\tau_i, \mathbf{e}_h^i)\|}{\tau_i^2 \|\mathbf{e}_h^i\|}, \forall \mathbf{e}_h^i, \forall \tau_i \in (0, \tau_s(\mathbf{e}_h^i)) \right\} \tag{4.27}$$

Since $\hat{\mu}^*$ is computed on a sampled set, we may discard some configurations which would give a larger value of the estimate. Hence we conclude that $\hat{\mu}^* \leq \mu^*$ and $\hat{\mu}^* \to \mu^*$ as the number of samples increases, that is, as $N \to \infty$. From (4.24) and Prop.4.5.3, given $\mathbf{e}_h$ such that $\mathbf{q}^r + \mathbf{e}_h \in \mathcal{B}_*(\mathbf{q}^r)$, we have that

$$\begin{aligned} \|\mathbf{e}'(\tau; \mathbf{e}_h)\| &\leq \|(\mathbf{I} - \tau \overline{\mathbf{K}}) \mathbf{e}_h\| + \mu \tau^2 \|\mathbf{e}_h\| \\ &\leq \left(\|(\mathbf{I} - \tau \overline{\mathbf{K}})\| + \mu \tau^2 \right) \|\mathbf{e}_h\| \end{aligned} \tag{4.28}$$

for all $0 \leq \tau < \tau_s(\mathbf{e}_h)$, where $\|(\mathbf{I} - \tau \overline{\mathbf{K}})\|$ denotes the matrix norm induced by the corresponding vector norm. In order to evaluate upper bounds for the convergence rate, we need to study the following function

$$g(\tau; \mu) := \|\mathbf{I} - \tau \overline{\mathbf{K}}\| + \mu \tau^2$$

Since $\overline{\mathbf{K}}$ is diagonal we can write

$$g(\tau; \mu) = \max_i \frac{|1 - \tau \bar{k}_i|}{d_i^2} + \mu \tau^2. \tag{4.29}$$

Observe that $g(\tau; \mu)$ is a strictly convex function and that, from (4.9), it follows

$$g(0; \mu) = \max_i \frac{1}{d_i^2} \leq 1.$$

Moreover, $g'(0; \mu) < 0$. Based on these properties of the function g we can define

$$\overline{\tau}_s(\mu) := \{\tau > 0 \,|\, g(\tau; \mu) = 1\}$$

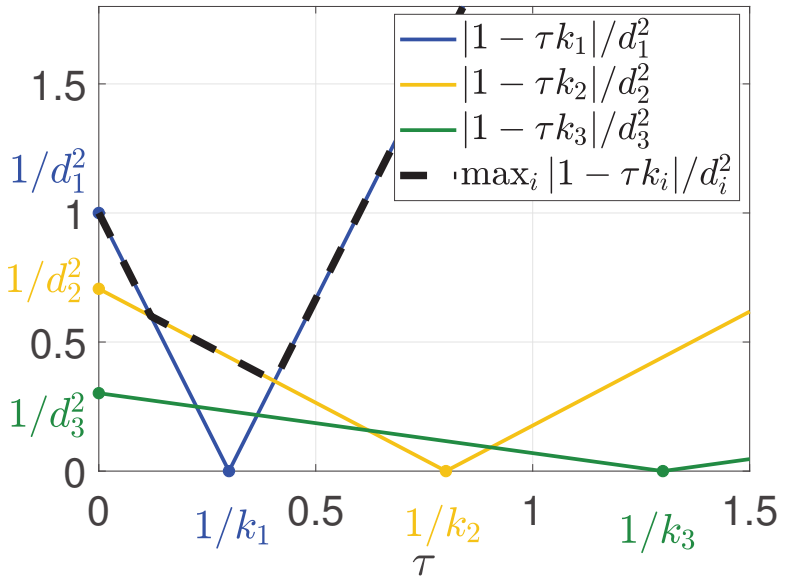


Figure 4.6. Representation of $\frac{|1-\tau k_i|}{d_i^2}$ and of the maximum.

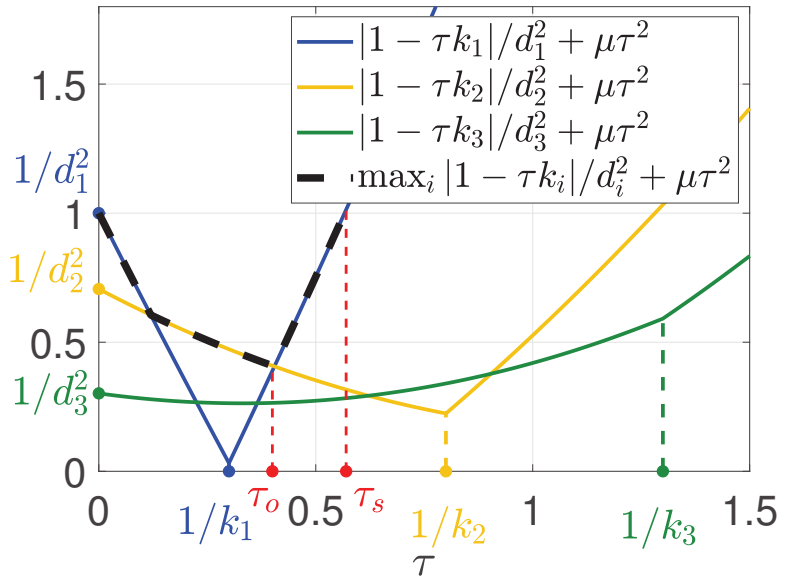


Figure 4.7. Representation of $\frac{|1-\tau k_i|}{d_i^2}+\mu\tau^2$ and of the maximum.

and, in turn,

$$\overline{\tau}_o(\mu) := \underset{0\leq\tau\leq\overline{\tau}_s(\mu)}{\operatorname{argmin}} \; g(\tau;\mu).$$

Accordingly let

$$\rho(\mu) = g(\overline{\tau}_o(\mu);\mu).$$

Since $\rho(\mu) < 1$, it turns out that the pair $(\overline{\tau}_o(\mu), \rho(\mu))$ satisfies the properties stated in Proposition 4.5.2.

The strict-convexity property of the function g allows to compute $\overline{\tau}_s(\mu)$, $\overline{\tau}_o(\mu)$ and $\rho(\mu)$ in a efficient way, by resorting to standard numerical toolboxes. In order to better understand the properties of the function g just described, we depict in Figure 4.6 the terms $\frac{|1-\tau k_i|}{d_i^2}$ and in Figure 4.7 $\frac{|1-\tau k_i|}{d_i^2} + \mu\tau^2$ and their maxima. We chose random values for k_i and d_i such that (4.9) is satisfied and we show also the values $\bar{\tau}_s(\mu)$, where $g = 1$ and $\bar{\tau}_0(\mu)$ where g has its minimum. As you can notice by comparing the two pictures, because of the quadratic term $\mu\tau^2$, the minimum of g can never be zero, neither for $\mu = 0$; indeed, if in Chapter 3 g was minimized in $\tau = 1$ when $\mu = 0$, in this case the value of the minimizer depends on the values k_i which can be different from 1. As a consequence we cannot obtain a dead-beat controller anymore.

4.5.2 Online model-predictive procedure

In this section we consider the possibility to numerically compute the future trajectory $\mathbf{e}'(\tau, \mathbf{e}_h)$ based on the model dynamics $\mathbf{f}(\mathbf{q};\mathbf{e}_h)$ and the current position $\mathbf{e}_h$. This implies that also $\tau_o(\mathbf{e}_h)$ can be computed at any time step h. If so, under the assumption that the input is kept constant for the following time interval T, we can propose the following input

$$\mathbf{u}(t) = \mathbf{u}_h = -k_h \mathbf{A}_{\mathbf{e}_h} \overline{\mathbf{K}} \mathbf{e}_h, \qquad hT \leq t < (h+1)T,$$

where

$$k_h := \frac{\tau_o(\mathbf{e}_h)}{T}. \tag{4.30}$$

A more precise characterization of the convergence properties of this strategy is stated in the next proposition.

Proposition 4.5.4. *Consider the system in (4.16) with a time varying sequence of gains $k_0, k_1, k_2, \ldots$, where the generic k_h is given as in (4.30). Then we have that*

$$\|\mathbf{e}(t)\| \leq \|\mathbf{e}(hT)\|$$

for all $hT \leq t < (h+1)T$, and

$$\|\mathbf{e}(hT)\| \leq \rho^h(\mu)\, \|\mathbf{e}(0)\|.$$

Remarkably, the convergence is at least quadratic if one of the following two facts is verified

- *the weights of the matrix $\mathbf{D}$ are all equal, that is,*

$$d_1 = d_2 = \ldots = d_N = d_V;$$

- *the gains defining the matrix* $\overline{\mathbf{K}}$ *are all equal, that is,*

$$k_1 = k_2 = \ldots = k_N = k_V.$$

Proof. Observe that, according to (4.30), we necessarily have:

$$\|\mathbf{e}_{h+1}\| = \|\mathbf{e}'(\tau_o(\mathbf{e}_h); \mathbf{e}_h)\| \leq \|\mathbf{e}'(\overline{\tau}_o; \mathbf{e}_h)\| \leq \rho \|\mathbf{e}_h\|,$$

hence the proposed scheme is exponentially stable with rate ρ for any T. This proves the first part of the Proposition.

Consider now the case where all the d_i are equal to the same value $\bar{d}$. In this case we have that (4.29) can be rewritten as

$$g(\tau; \mu) = \max_i \frac{|1 - \tau \overline{k}_i|}{\bar{d}} + \mu \tau^2 \tag{4.31}$$

$$= \frac{|1 - \tau \overline{k}_{min}|}{\bar{d}} + \mu \tau^2 \tag{4.32}$$

where $\overline{k}_{min} = \min_i k_i$.

Since in the online scenario $r = \|\mathbf{e}_h\| \to 0$, then $\mu = \frac{1}{2} ab \|\mathbf{e}_h\| \to 0$. This implies that $\overline{\tau}_o(\mu) \to 1/\overline{k}_{min}$ and, in turn, $\rho(\mu) \to \frac{1}{\overline{k}_{min}^2} \mu$. As so, $\|\mathbf{e}_{h+1}\| \leq \rho \|\mathbf{e}_h\| \to \frac{1}{2\overline{k}_{min}^2} ab \|\mathbf{e}_h\|^2$.

As a consequence

$$\limsup_{h \to +\infty} \frac{\|\mathbf{e}_{h+1}\|}{\|\mathbf{e}_h\|^2} \leq \frac{1}{2\overline{k}_{min}^2} ab,$$

and $\|\mathbf{e}_h\| \leq (\frac{1}{2\overline{k}_{min}^2} ab \|\mathbf{e}_0\|)^{(2^h - 1)} \|\mathbf{e}_0\|$. Since $ab > 0$, then the quadratic convergence of the sequence $\|\mathbf{e}_h\|$ is guaranteed.

Similar reasonings hold for the scenario where the gains defining the matrix $\overline{\mathbf{K}}$ are all equal with each other. ∎

Based on the definition of $\tau_o(\mathbf{e}_h)$ and on Prop. 4.5.4, we expect the online strategy to exhibit a faster convergence than the offline one. This fact is supported also by the numerical results reported in the next section. However, the higher rate of convergence comes at the price of a heavier computational load. Indeed $\tau_o(\mathbf{e}_h)$ needs to be estimated at each iteration and a global knowledge of the vector $\mathbf{e}_h$ is required; this implies that the online strategy cannot be implemented distributively, but only in a centralized fashion.

4.6 Simulations

In the first part of this section we show the results of the simulations performed on Matlab-Simulink when the off-line (Section 4.5.1) and on-line (Section 4.5.2) techniques are applied to the kinematic model of the system depicted in Figure 4.3a. In detail, we assume that the sampling time is $T = 1.5$ $[s]$ and the weight and gain matrices are $\bar{\mathbf{K}} = \mathbf{I}_{10}$ and $\mathbf{D} = \text{diag}\{\mathbf{I}_4, \ 4 \cdot \mathbf{I}_6\}$. In this case we chose to weight differently the variables $q_i, i = 1, 2, 3, 4$ with respect to $\mathbf{q}_V$. In Figure 4.8 we compare the behavior of the system when the 2-norm is considered and the the off-line and on-line strategies are compared.

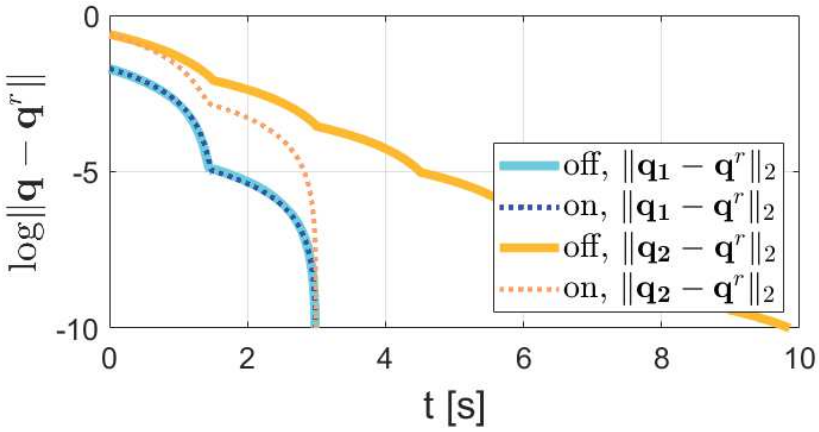


Figure 4.8. Evolution of the error norm $\|\mathbf{q}-\mathbf{q}^r\|_2$ for different initial conditions ($\mathbf{q}_1$, $\mathbf{q}_2$) and techniques (on-line and off-line) when the robots velocities generated by the planner are applied to the kinematic system. In this case $\mathbf{K}=\mathbf{I}_{10}$ and $\mathbf{D}=\text{diag}\{\mathbf{I}_4, 4\cdot\mathbf{I}_6\}$.

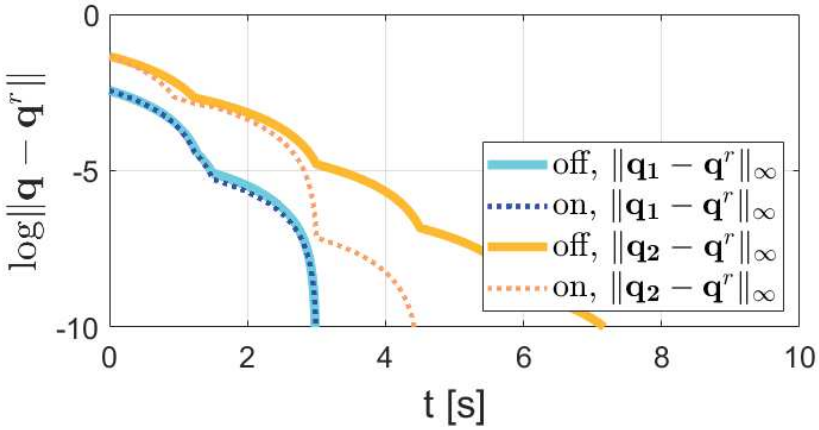


Figure 4.9. Evolution of the error norm $\|\mathbf{q}-\mathbf{q}^r\|_\infty$ for different initial conditions ($\mathbf{q}_1$, $\mathbf{q}_2$) and techniques (on-line and off-line) when the robots velocities generated by the planner are applied to the kinematic system. In this case $\mathbf{K}=\mathbf{I}_{10}$, $\mathbf{D}=\text{diag}\{\mathbf{I}_4, 4\cdot\mathbf{I}_6\}$.

In addition, two different initial conditions are taken into account. The picture shows that, when $\mathbf{q}(0)=\mathbf{q}_1$ is close to $\mathbf{q}^r$, then the two strategies have more or less the same performance. Indeed, in this case, a smaller value of μ is obtained from procedure (4.27): as a consequence, the minimum of g is smaller and the convergence rate is closer to the one of the on-line scenario. Instead, the difference between the two strategies is evident when $\mathbf{q}(0)=\mathbf{q}_2$ that is further from the reference configuration. In Figure 4.9 a similar result is obtained: the only difference is the norm used to perform the simulations, that is $\|\cdot\|_\infty$. Finally, in Figure 4.10 we performed more simulations but using the dynamical model of the system, derived as explained in Appendix B.1.2: in particular, our goal is to graphically show the validity of Theorem 4.4.1. For this reason we only consider the 2-norm, indeed the ∞-norm would give the same result. The picture highlights that for large values of α the convergence of the system is achieved, while for lower values the norm of the error is not guaranteed to converge to zero. As explained above in Section 4.3, this means that for large α the approximation $\dot{\mathbf{q}}\approx\mathbf{A_q}^\dagger\mathbf{u}$ holds true and the dynamical model can be approximated with the kinematic one.

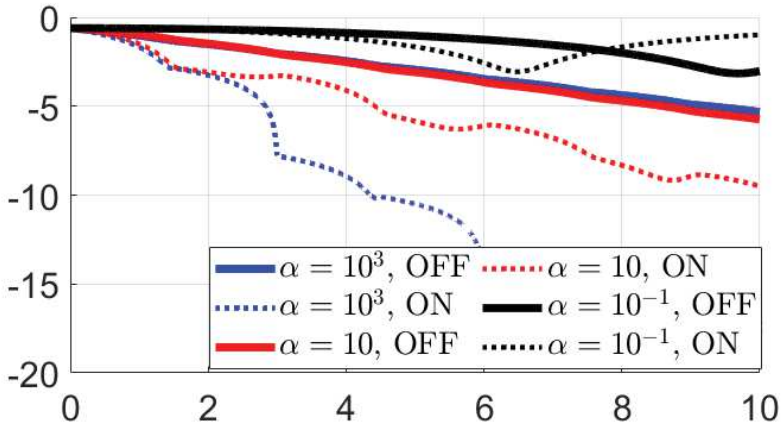


Figure 4.10. Evolution of the error norm $\|\mathbf{q} - \mathbf{q}^r\|_2$ when the on-line and off-line techniques are applied: moreover, three different values of the gain α are chosen.

4.6.1 Truetime simulations

In this section we describe the results obtained by simulating the system in Figure 4.3a and comparing the off-line and on-line strategies. The meaningful difference with respect to the previous simulation is that the proposed solutions are tested in a more realistic environment. Indeed, the sampled communication is implemented as a wireless network by using *Truetime* that will be described in the following; moreover, the desired velocities for the robots are given as input to the dynamical model of the system instead of the kinematic one. Note that the simulation is related to the special case of four robots which carry the load. Hence $m_i = 1, \quad i = 1,2,3,4$, $m_V = 6$ and $m = 10$; moreover, $N = 4$ and $n = 3N = 12 > m$, so this is a particular case of an over-actuated system. TrueTime is a Matlab/Simulink-based simulator for real-time control systems. It facilitates co-simulation of controller task execution in real-time kernels, network transmissions, and continuous plant dynamics. We implemented our architecture simulating a wireless network 802.11b where five nodes (the four robots and the transported load) are connected by one network block. In our scenario, the load, depending on the chosen implementation, tries to transmit its pose $\mathbf{q}_V$ or the vector of desired velocities $\mathbf{u}$ to the robots via wireless with a data rate equal to 1.2Mbits/s. Hence, a triggering signal is sent to the network block on the corresponding input channel. When the simulated transmission of the message is finished, the network block sends a new triggering signal on the output channel corresponding to the receiving node. The transmitted message is put into a buffer at the receiving node. In order to compare the performance of the controlled system when different packets loss probabilities are considered, in Figure 4.11 we report the norm of the configuration error $\mathbf{e} = \mathbf{q} - \mathbf{q}^r$ for different values of the parameter: $0\,\%, 30\,\%, 70\,\%$. The initial configuration is set to $\mathbf{q}^0 = [80^\circ\ 80^\circ\ 80^\circ\ 80^\circ\ 0[m]\ 0[m]\ 0[m]\ 0^\circ\ 0^\circ\ 0^\circ]^\top$ to the final one to $\mathbf{q}^r = [60^\circ\ 60^\circ\ 60^\circ\ 60^\circ\ 0[m]\ 0[m]\ 0[m]\ 0^\circ\ 0^\circ\ 0^\circ]^\top$. As expected, since the input $\mathbf{u}$ to the system is updated only when a new measurement arrives, then the performance of the controlled system gets worse as the probability of loosing packets increases.

In Section A.1 we also report more detailed plots of an additional simulation where we set the packet loss probability equal to 20 %: we show the behavior of the Lagrangian

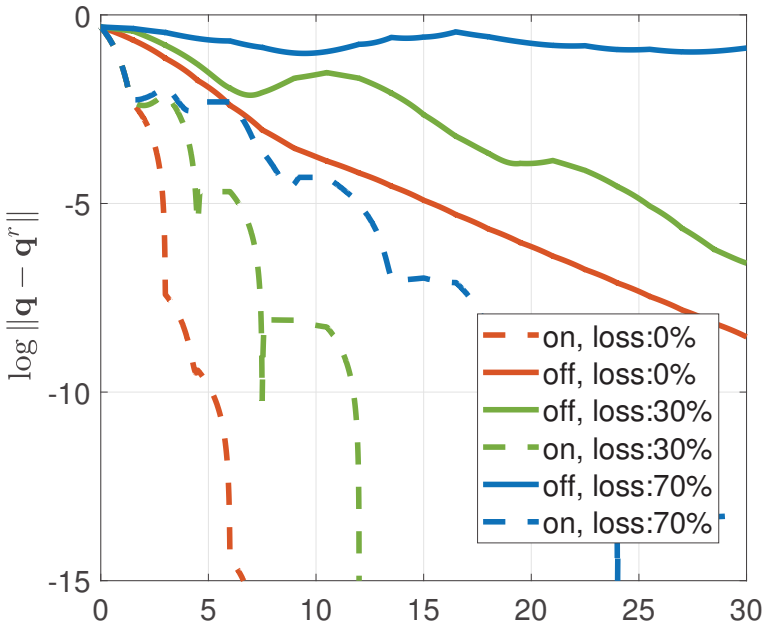


Figure 4.11. Plot of the error norm when the off-line and on-line techniques are implemented on the dynamical model of the system depicted in Figure 4.3a. Different packet loss probabilities are considered.

variables together with the reference velocities $\mathbf{u}$ generated by the local planner. We also show how the allocation of common resources is handled: in particular the figures show when information is sent on the network and if a packet is lost.

5

Cooperative trajectory tracking over sampled communication

Contents

In this Section we propose an inverse-kinematics controller for a class of multi-robot systems in the scenario of sampled communication. The goal is to make a group of robots cooperate to fulfill a common goal. In particular, we make three aerial robots transport a common object via cables: a trajectory is assigned and tracked by the system. Given a feasible desired trajectory in the configuration space, the controller receives the measurements from the system at sampled time instants and computes the desired robots velocities. We provide two strategies to choose the controller gain. In one case it is computed off-line: once the sampling time is assigned, we propose a method to compute the controller gain that guarantees stability and asymptotic zero-tracking error. In this case the controller is implementable in a distributed way. In the second scenario the gain is updated on-line and only a centralized implementation can be achieved: in this scenario we practically show by means of simulations that the proposed control method provides a faster convergence rate while tracking the assigned trajectory. Notice that both strategies work with no communication among robots, but only with the common

load. Numerical simulations of the two techniques are provided for the cooperative aerial manipulation of a cable-suspended load (the Fly-crane). We also compare our strategies to other possible solutions, showing that we can obtain better performance and stability guarantees.

5.1 Introduction

Generally, a group of robots allows to improve the task performance with respect to the use of a single robot: problems like the limited payload and time of flight can be mitigated using multiple robots. However, groups of mobile robots requires a careful consideration of cooperation strategies to achieve a common objective. Let us consider the scenario where a group of UAVs transport a common object while avoiding obstacles or passing through a narrow aperture. In this case, it is not sufficient that each UAV avoids obstacles. The overall multi-robot system should move in a cooperative way such that the transported load avoids obstacles as well. In Chapter 3 we already discussed the reasons why we are interested in a decentralized implementation of the controller where we include the possibility to exchange information among the robots and a central unit.

We also investigated the problem of cooperative aerial load transportation via sampled communication. In particular, considering the problem of rest-to-rest motions, we designed an inverse kinematic planner that makes the system move from an initial to a final configuration with zero velocity. In particular, we analysed the behavior of the system under sampled communication (e.g., in case of wi-fi transmission), namely, when the desired robots velocities are transmitted at sampled time instants to the robots. In this paper we propose a *Sampled communication-aware Inverse-Kinematic controller for Multi-robot systems* (SIKM) to address the more relevant challenging problem of trajectory tracking under sampled communication. We depart from the work of Chapter 3 considering a more generic framework. The contributions of this chapter can be resumed in the following points:

- We exploit the kinematic model derived for square systems and we exploit the same multi-layer architecture: the difference is that we want the system to track a desired trajectory (positions and velocities in time), hence no more a sequence of configurations.
- We design the local planner both in the continuous-time and sampled scenarios
- In the sampled case we show that the strategy developed in Chapter 3 can still be applied for trajectory tracking but cannot guarantee zero-tracking error. We provide a solution that gives stability and convergence guarantees
- In the sampled scenario, we give the definition of ρ monotonically contractiveness in order to derive stability and convergence characterizations both during a sampling interval and among different time intervals.
- Based on the definition of ρ monotonically contractiveness, we derive explicit bounds for the stability set and the convergence rate

- Finally, we perform simulations with the Fly-crane in a far-from ideal scenario; sensors noise and software in the simulation are considered as well.

Remark 5.1.1. It is worth motivating the choice of the strategy name (*controller*) with respect to the *planner* introduced in Chapter 3: even if in both cases the reference velocities for the vehicles are generated, in the first scenario a sequence of velocities is planned from the initial to the final configuration of the system. In this scenario, instead, for each point of the trajectory to be tracked, a desired velocity is computed, based on the received measurements, hence no sequence of reference velocities is generated between two consecutive configurations. For this reason we decided to distinguish the technique proposed in this chapter by using the term *controller*.

5.1.1 Overview

In this section we outline the structure of the chapter in order to facilitate the reading. In Section 5.2 we introduce the class of multi-robot systems that we aim at controlling to perform trajectory tracking. In particular, we define the kinematic model of the class (Section 5.2.1), recalling that different types of real systems can be represented by such model (Section 3.2.2). Then, we introduce and formalize the problem in exam, detailing our contribution (Section 5.2.2): we design a SIKM controller to perform trajectory tracking with a multi-robot system under sampled communication. We firstly consider the case where measurements are continuously available and we show that the combination of a feed-back and feed-forward term guarantees perfect tracking. We then analyze in Section 5.3 the sampled scenario, when measurements are available at discrete times. To this aim, we firstly (Section 5.3.1) recall the solution proposed in Chapter 3 for the point-stabilization problem: this is helpful to better understand our contribution. Then (Section 5.3.2) we show that the technique can be exploited also to track a trajectory, even though no guarantees about stability can be proved. Thus, in Section 5.3.3 we introduce our solution to design a controller to perform trajectory tracking in the scenario of sampled communication (SIKM). Until this point the results are of existential type, hence in Section 5.4 we find explicitly the sets of the controller parameters which guarantee the system to keep stable during the maneuver and to converge as fast as possible. Finally, in Section 5.5 we provide the results obtained by simulating our strategy on a far-from-ideal dynamical system. We show that it gives better performance with respect to the other techniques introduced before.

5.2 Modeling and Problem Formulation

5.2.1 Kinematics of multi-robot systems

In this section we briefly recall the kinematic model of a multi-robot system composed by N robots which cooperate to fulfill a task: a more detailed description is given in Section 3.2. The robots do not communicate among themselves but receive information from a

central unit, that is called *pivot* and is indicated with the letter V. Notice that it can be an object to be manipulated (in this case we assume that the object is linked to the robots through passive, non extendable connections), a robot to be escorted or an other object in the space. The kinematic model was introduced in (3.1) and is derived as in Manubens et al. (2013):

$$\begin{bmatrix}\mathbf{p}_1 & \dots & \mathbf{p}_N\end{bmatrix}^\top = \mathbf{h}(\mathbf{q}) = \begin{bmatrix}\mathbf{h}^{(1)}(\mathbf{q}_1,\mathbf{q}_V) & \dots & \mathbf{h}^{(N)}(\mathbf{q}_N,\mathbf{q}_V)\end{bmatrix}^\top$$

This function maps the Lagrangian coordinates of the system $\mathbf{q} = [\mathbf{q}_1^\top \cdots \mathbf{q}_N^\top \ \mathbf{q}_V^\top]^\top \in \mathbb{R}^{m\times 1}, m > 0$ to the vector collecting the robots configurations $\mathbf{p} = \begin{bmatrix}\mathbf{p}_1^\top & \dots & \mathbf{p}_N^\top\end{bmatrix}^\top \in \mathbb{R}^{n\times 1}, n > 0$: $\mathbf{p}_i \in \mathbb{R}^{n_i\times 1}$ is the position of the i-th robot in the space, $\mathbf{q}_i \in \mathbb{R}^{m_i\times 1}, m_i > 0$ gathers the angles and/or distances between the pivot V and the i-th robot, and $\mathbf{q}_V \in \mathbb{R}^{m_V\times 1}, m_V > 0$ represents the pose (position and orientation) of the pivot itself.

The differential kinematics of the system is:

$$\dot{\mathbf{p}} = \mathbf{A}_\mathbf{q}\,\dot{\mathbf{q}},$$

where the Jacobian $\mathbf{A}_\mathbf{q} = \frac{\partial \mathbf{h}(\mathbf{q})}{\partial \mathbf{q}} \in \mathbb{R}^{n\times m}$ has the structure

$$\mathbf{A}_\mathbf{q} = \left[\begin{array}{ccc|c} \mathbf{A}_{\mathbf{q}_1}^{(1)} & & \mathbf{0} & \mathbf{A}_{\mathbf{q}_V}^{(1)} \\ & \ddots & & \vdots \\ \mathbf{0} & & \mathbf{A}_{\mathbf{q}_N}^{(N)} & \mathbf{A}_{\mathbf{q}_V}^{(N)} \end{array}\right]$$

and $\mathbf{A}_{\mathbf{q}_i}^{(i)} = \frac{\partial \mathbf{h}^i(\mathbf{q})}{\partial \mathbf{q}_i} \in \mathbb{R}^{n_i\times m_i}$ and $\mathbf{A}_{\mathbf{q}_V}^{(i)} = \frac{\partial \mathbf{h}^i(\mathbf{q})}{\partial \mathbf{q}_V} \in \mathbb{R}^{n_i\times m_V}$.
In this section we will focus on the case $n = m$, that is on *square systems*. Anyway, we obtained interesting results in Section 4.1 about point-stabilization control for non-square systems, hence we expect that similar extensions can be applied to implement trajectory tracking.

5.2.2 Problem Formulation and contribution

In this work we formulate a trajectory tracking problem where a square system must follow a sequence of desired configurations and velocities $(\mathbf{q}^r(t), \dot{\mathbf{q}}^r(t))$, $t \geq 0$, assigned a-priori in the configuration space. The trajectory could be generated by a planner that takes into account high-level performance metrics such as obstacles avoidance, minimum energy trajectories and singularity avoidance. We assume that each robot is endowed with a dynamical controller sufficiently fast with respect to the dynamics of the entire system, s.t. the velocities of the robots are fully controllable, i.e.:

$$\dot{\mathbf{p}}(t) = \mathbf{u}(t), \tag{5.1}$$

where $\mathbf{u} = [\mathbf{u}_1^\top \cdots \mathbf{u}_N^\top]^\top \in \mathbb{R}^n$. We consider two possible architectures to implement the SIKM controller:

$$\mathbf{u}_i(t) = \kappa_i^s(\mathbf{q}_i(t), \mathbf{q}_V(t); \mathbf{q}_i^r(t), \mathbf{q}_v^r(t), \dot{\mathbf{q}}^r(t)), \qquad i = 1,\dots N \tag{5.2}$$

$$\mathbf{u}_i(t) = \kappa_i^c(\mathbf{q}(t); \mathbf{q}^r(t), \dot{\mathbf{q}}^r(t)), \qquad i = 1,\dots N \tag{5.3}$$

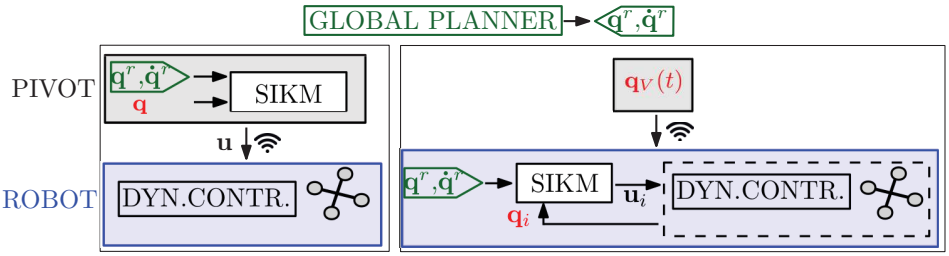


Figure 5.1. Representation of (5.2) (right) and (5.3) (left). The pivot is represented in gray, each robot (endowed with a dynamical controller) in blue, the measurements from the sensors in red and the desired trajectory in green. The dynamical controller converts $\mathbf{u}$ into forces for the robot. The wireless symbol refers to sampled communication.

where in (5.2) $\mathbf{u}_i(t)$ does not depend on $\mathbf{q}_j(t)$ with $j \neq i$, while in (5.3) $\mathbf{u}_i(t)$ depends on the whole vector $\mathbf{q}$. We refer to the former architecture as *sparse feedback*, and to the latter one as *centralized feedback*. In Figure 5.1 the two implementations are depicted: on the left, the centralized SIKM controller is implemented on the pivot and it receives all the state measurements $\mathbf{q}$; then, the computed input $\mathbf{u}$ is sent via wireless to the robots. Differently, in the sparse implementation, the controller can be located separately on each robot and the pivot pose is received via wireless. We will show that if $\mathbf{q}(t)$ is continuously accessible to the SIKM controller, then (5.2) is sufficient to make the system track the desired trajectory, hence the knowledge of the whole vector $\mathbf{q}$ is not necessary, see (5.6). On the other hand, in the more realistic scenario when some of the components of $\mathbf{q}(t)$ need to be sampled and transmitted via wireless, then we can prove that (5.2) guarantees the stability of the system and asymptotic zero-tracking error (strategy 2 in the following). Instead for strategy (5.3) we cannot provide similar theoretical results, but only practical ones (strategy 1).

We have shown in Chapter 3, in Proposition 1, that the control law

$$\mathbf{u}(t) = -k\mathbf{A}_{\mathbf{q}(t)}(\mathbf{q}(t) - \mathbf{q}^r) := \mathbf{u}_k(t), \qquad k > 0, \tag{5.4}$$

drives exponentially fast the state $\mathbf{q}(t)$ to the desired constant configuration $\mathbf{q}^r$. In order to track a trajectory the feed-forward velocity term

$$\mathbf{u}_{ff}(t) = \mathbf{A}_{\mathbf{q}(t)}\dot{\mathbf{q}}^r(t)$$

is typically added to the control input $\mathbf{u}_k(t)$ Sanz (2009). Indeed, by applying the control law

$$\mathbf{u}(t) = \mathbf{u}_k(t) + \mathbf{u}_{ff}(t) \tag{5.5}$$

the kinematics of system becomes

$$\dot{\mathbf{q}}(t) = -k\Big(\mathbf{q}(t) - \mathbf{q}^r(t)\Big) + \dot{\mathbf{q}}^r(t)$$

Defining $\mathbf{e}(t) = \mathbf{q}(t) - \mathbf{q}^r(t)$ we get

$$\dot{\mathbf{e}}(t) = \dot{\mathbf{q}}(t) - \dot{\mathbf{q}}^r(t) = -k\Big(\mathbf{q}(t) - \mathbf{q}^r(t)\Big) = -k \cdot \mathbf{e}(t)$$

which shows that the error goes exponentially fast to zero if $k > 0$. Moreover notice that

$$\mathbf{u}_i(t) = k\mathbf{A}^{(i)}_{\mathbf{q}_i(t)}(\mathbf{q}_i(t) - \mathbf{q}^r_i(t)) + k\mathbf{A}^{(i)}_{\mathbf{q}_V(t)}(\mathbf{q}_V(t) - \mathbf{q}^r_V(t)) \tag{5.6}$$

$$+ \mathbf{A}^{(i)}_{\mathbf{q}_i(t)}\dot{\mathbf{q}}^r_i(t) + \mathbf{A}^{(i)}_{\mathbf{q}_V(t)}\dot{\mathbf{q}}^r_V(t), \tag{5.7}$$

that is, the control law exhibits the structure in (5.2), hence it is amenable of a sparse implementation.

To deal with sampled measurements, that is when information about $\mathbf{q}$ is available only every T seconds, being T the sampling time, in Chapter 3 we have proposed the following sampled version of (5.4)

$$\mathbf{u}(t) = \mathbf{u}_k(hT) = -k\mathbf{A}_{\mathbf{q}(hT)}(\mathbf{q}(hT) - \mathbf{q}^r), \tag{5.8}$$

for $t \in [hT, (h+1)T)$, $h = 0, 1, 2, \ldots$. Under mild assumptions that we review in next Section, (5.8) is shown to still drive $\mathbf{q}$ exponentially fast to $\mathbf{q}^r$.

We conclude this section by making the following assumption.

Assumption 5.2.1. *The following relations hold:*

1. *The reference trajectory $\mathbf{q}^r(t) \in \mathcal{Q}$ is twice continuously differentiable and $\mathcal{Q}$ is a compact set; moreover speed and accelerations are bounded, i.e. $\|\dot{\mathbf{q}}^r(t)\| \leq v_{max}$ and $\|\ddot{\mathbf{q}}^r(t)\| \leq a_{max}$*
2. *There exist $d > 0$ such that, for any $\mathbf{q}^r(t)$ and $\mathbf{q}$ being at distance smaller than d from the the trajectory to be followed, i.e., $\|\mathbf{q}^r(t) - \mathbf{q}\| < d$ for some t, then $\mathbf{A}_q$ is twice continuously differentiable and invertible;*
3. *$\|\mathbf{q}(0) - \mathbf{q}^r(0)\| < d$.*

In the following, we are interested to find a control law that guarantees asymptotic zero-tracking error.

5.3 SIKM: sampled measurements

In this section we analyze the solutions we propose to deal with the trajectory tracking problem formulated in the previous section in presence of sampled measurements. The section is organized as follows. We first review the positioning control law proposed in Chapter 3. Based on those results we characterize the performance of that strategy in the more general scenario of trajectory tracking. Then we introduce two strategies to improve the performance and give stability guarantees. In the remaining of the chapter we assume that the sampled measurements of $\mathbf{q}$ are available every T seconds.

5.3.1 Review of the positioning control law proposed in Chapter 3

It is straightforward to see that the dynamics of the system $\dot{\mathbf{q}} = \mathbf{A}_{\mathbf{q}}^{-1}\mathbf{u}$, under the action of (5.8) becomes

$$\dot{\mathbf{q}}(t) = -k\mathbf{A}^{-1}_{\mathbf{q}(t)}\mathbf{A}_{\mathbf{q}(hT)}(\mathbf{q}(hT) - \mathbf{q}^r).$$

In Chapter 3 we have shown that, if $\mathbf{q}(0)$ belongs to a ball $\mathcal{B}_r(\mathbf{q}^r)$ of a certain radius r centered on the reference point $\mathbf{q}^r$ such that the Jacobian $\mathbf{A}_\mathbf{q}$ is twice continuously differentiable for any $\mathbf{q} \in \mathcal{B}_r(\mathbf{q}^r)$, then there exist a k^* such that if $k < k^*$ then exponential stability is achieved. More precisely, defining $\mathbf{e}(t) = \mathbf{q}(t) - \mathbf{q}^r(t)$, it has been proved that there exists $0 < \rho < 1$ such that $\mathbf{e}((h+1)T) < \rho \mathbf{e}(hT)$ and $\mathbf{e}(t) < \mathbf{e}(hT)$ for $hT < t \leq (h+1)T$. It is easy to see that, if k is assigned offline then (5.8) can be implemented in a sparse way.

In Chapter 3 we have shown that the performance of (5.8) can be improved by designing the gain k online leading to a time-varying sequence k_h. In particular the gain k_h is designed bby studying the following auxiliary system

$$\begin{aligned} \dot{\mathbf{q}}'(\tau;\mathbf{q}_h) &= -\mathbf{A}^{-1}_{\mathbf{q}'(\tau;\mathbf{q}_h)}\mathbf{A}_{\mathbf{q}_h}\mathbf{q}_h =: \mathbf{f}(\mathbf{q}'(\tau);\mathbf{q}_h) \\ \mathbf{q}'(0;\mathbf{q}_h) &= \mathbf{q}_h; \quad \mathbf{q}_h \in \mathcal{B}_r(\mathbf{q}^r), \end{aligned} \tag{5.9}$$

being $\mathbf{q}_h := \mathbf{q}(hT)$. Since $\mathbf{f}(\mathbf{q},\mathbf{q}) = -\mathbf{q}$, then $\|\mathbf{q}'(0^+,\mathbf{q}_h) - \mathbf{q}^r\| < \|\mathbf{q}_h - \mathbf{q}^r\|$. Therefore only two scenarios are possible, either $\|\mathbf{q}'(\tau;\mathbf{q}_h) - \mathbf{q}^r\| < \|\mathbf{q}_h - \mathbf{q}^r\|$ for all $\tau > 0$, or not. Let us introduce

$$\tau_s(\mathbf{q}_h) := \min_\tau \{\tau > 0 \,|\, \|\mathbf{q}'(\tau;\mathbf{q}_h)\| = \|\mathbf{q}_h\|\} \tag{5.10}$$

and, accordingly,

$$\tau_o(\mathbf{q}_h) := \operatorname*{arginf}_{0 \leq \tau \leq \tau_s(\mathbf{q}_h)} \|\mathbf{q}'(\tau;\mathbf{q}_h)\|. \tag{5.11}$$

where $\tau_s(\mathbf{q}_h) = \infty$ if $\|\mathbf{q}'(\tau;\mathbf{q}_h) - \mathbf{q}^r\| < \|\mathbf{q}_h - \mathbf{q}^r\|, \forall \tau$. Then k_h is designed as

$$k_h = \frac{\tau_o(\mathbf{q}_h)}{T}. \tag{5.12}$$

Interestingly in Chapter 3 we have shown that, adopting the time-varying sequence $k_0, k_1, k_2, \ldots$, generated as in (5.12), then the convergence rate of the sampled dynamics is at least quadratic. Moreover the gain k_h tends to $1/T$ as $h \to \infty$.

Notice that the computation of $\tau_o(\mathbf{q}_h)$ requires the knowledge of the entire vector $\mathbf{q}(hT)$, hence the design of k_h must be done online and in a centralized way.

5.3.2 First approach: point-stabilization control

In the previous Section, we have discussed how, given a set $\mathcal{B}_o$ of possible configurations of the system, it is possible to compute an off-line gain k s.t. $\mathbf{u}_k(\,\cdot\,)$ in (5.8), drives the system to a desired configuration $\mathbf{q}^r \in \mathcal{B}_o$ exponentially fast.

Given that performance, the first idea to perform trajectory tracking is to exploit the same technique by assigning, at a certain frequency, a sequence of points as references to the system, instead of a single point. Let us sample the reference trajectory every T seconds, obtaining the sequence of points $(\mathbf{q}^r(hT), \dot{\mathbf{q}}^r(hT))$. Then the applied control law is

$$\mathbf{u}(t) = \mathbf{u}_k(hT) = -k\mathbf{A}_{\mathbf{q}(hT)}(\mathbf{q}(hT) - \mathbf{q}^r(hT)), \tag{5.13}$$

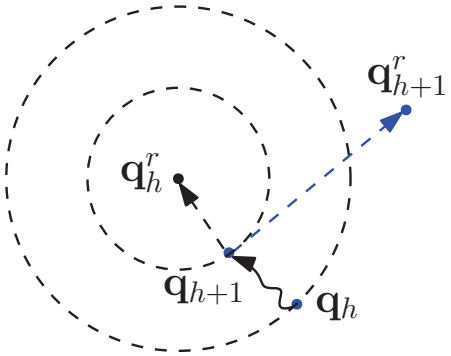


Figure 5.2. Trajectory tracking by using the point stabilization strategy.

for $t \in [hT, (h+1)T)$.

In the next proposition the performance of (5.13) is characterized when k is assigned off-line.

Proposition 5.3.1. *Consider the system described by (3.2) and (5.1), where* $\mathbf{u}$ *is given by (5.13). There exist* $r > 0$, $\beta > 0$, $0 < \rho < 1$ *and* k^* *such that, if* $\mathbf{q}(0) \in \mathcal{B}_r(\mathbf{q}^r(0))$, $\|\mathbf{q}^r((h+1)T) - \mathbf{q}^r(hT)\| < \beta$ *for all* h *and* $0 < k < k^*$, *then the tracking error* $\mathbf{e}(hT) := \mathbf{q}(hT) - \mathbf{q}^r(hT)$ *is bounded. In particular,*

$$\|\mathbf{e}(hT)\| \leq \frac{\beta}{1-\rho}.$$

Proof. We want to prove that the tracking error is bounded, hence we give some definitions and derive the error dynamics when the point-stabilization technique is applied. Observe Figure 5.2 and note that at time $t = hT$ the system configuration is $\mathbf{q}_h = \mathbf{q}(hT)$ and the reference point is $\mathbf{q}^r_h$; then, at time $t = (h+1)T$, the system state is $\mathbf{q}_{h+1} = \mathbf{q}((h+1)T)$ and a new reference $\mathbf{q}^r_{h+1}$ is assigned and this strategy is applied every T seconds. Recall Proposition 3.4.5 of Chapter 3 where we have shown that, given a configuration $\mathbf{q}_h$ and a desired one $\mathbf{q}^r$, we can find an optimal gain for the control law such that

$$\|\mathbf{e}((h+1)T)\| \leq \rho \|\mathbf{e}(hT)\|, \quad \|\mathbf{e}(hT)\| \leq \rho^h \|\mathbf{e}(0)\|$$

where $\rho \in (0,1)$ was defined in Proposition 5 of Chapter 3: it depends on the how the set $\mathcal{B}_o$ is chosen and sampled. This relation tells us that the error decreases between two consecutive sampling times and also in each time interval $t \in (hT, (h+1)T)$. Now we define the quantities

$$\mathbf{e}^+_{h+1} := \mathbf{q}_{h+1} - \mathbf{q}^r_{h+1}, \qquad \mathbf{e}^-_{h+1} := \mathbf{q}_{h+1} - \mathbf{q}^r_h$$

In practice $\mathbf{e}^-_{h+1}$ represents the distance between the configuration $\mathbf{q}_{h+1}$ and the desired one $\mathbf{q}^r_h$. Then, $\mathbf{e}^+_{h+1}$ represents the error between $\mathbf{q}_{h+1}$ and the next desired configuration $\mathbf{q}^r_{h+1}$. From the previous definitions and properties, we can write

$$\begin{aligned}
\|\mathbf{e}_{h+1}\| = \|\mathbf{e}^-_{h+1}\| \leq \rho \|\mathbf{e}_h\| = \rho \|\mathbf{q}_h - \mathbf{q}^r_h\| = \rho \|\mathbf{e}^+_h\| \\
\|\mathbf{e}^+_{h+1}\| = \|\mathbf{q}_{h+1} - \mathbf{q}^r_h + \mathbf{q}^r_h - \mathbf{q}^r_{h+1}\| \leq \|\mathbf{q}_{h+1} - \mathbf{q}^r_h\| + \\
+ \|\mathbf{q}^r_h - \mathbf{q}^r_{h+1}\| = \|\mathbf{e}^-_{h+1}\| + \|\mathbf{q}^r_h - \mathbf{q}^r_{h+1}\| \leq
\end{aligned}$$

$$\leq \rho\|\mathbf{e}_h^+\| + \beta$$

where $\beta > 0$ is the maximum distance between two consecutive reference points. Now we find an upper bound of the steady-state tracking error: from the last inequality, we can start from $h = 0$ and write

$$\begin{aligned}
&\|\mathbf{e}_1^+\| \leq \rho\|\mathbf{e}_0^+\| + \beta \\
&\|\mathbf{e}_2^+\| \leq \rho\|\mathbf{e}_1^+\| + \beta \leq \rho^2\|\mathbf{e}_0^+\| + \rho\beta + \beta \\
&\|\mathbf{e}_3^+\| \leq \rho\|\mathbf{e}_2^+\| + \beta \leq \rho^3\|\mathbf{e}_0^+\| + \rho^2\beta + \rho\beta + \beta \\
&\quad\vdots \\
&\|\mathbf{e}_h^+\| \leq \rho^h\|\mathbf{e}_0^+\| + \beta\sum_{i=0}^{h-1}\rho^i = \rho^h\|\mathbf{e}_0^+\| + \beta\frac{1-\rho^h}{1-\rho}
\end{aligned}$$

Since $\rho < 1$, then for $h \to \infty$ we have

$$\|\mathbf{e}_h^+\| \leq \frac{\beta}{1-\rho}$$

and we conclude that the tracking error is bounded. ■

As we have just proved, the point-stabilization control gives some guarantees even when a sequence of different points is assigned as reference, instead of a constant desired configuration. Nevertheless it is not sufficient to guarantee zero tracking error. Intuitively, a better bound could be obtained by designing a sequence of time-varying gains $k_0, k_1, k_2, \ldots$ as described in the second part of Section 5.3.1. However this improvement of the performance would come at the price of an higher computational cost, since k_h needs to be computed via a centralized architecture.

5.3.3 Second approach: feed-forward technique

In the previous Section we have seen that the point-stabilization technique does not allow to track a varying reference with zero tracking error. Thus, the addition of a feed-forward velocity term is required as in the continuous-time scenario. A possible choice is to obtain the control law by sampling the continuous-time one (5.5):

$$\mathbf{u}(hT+\tau) = -k\mathbf{A}_{\mathbf{q}(hT)}(\mathbf{q}(hT) - \mathbf{q}^r(hT)) + \mathbf{A}_{\mathbf{q}(hT)}\dot{\mathbf{q}}^r(hT+\tau) \tag{5.14}$$

where $h \in \mathbb{N}, \tau \in [0, T)$ and $\dot{\mathbf{q}}^r(hT+\tau)$ is assumed to be continuously available[1]. Now we derive the dynamics of the tracking error $\mathbf{e}(hT+\tau) := \mathbf{q}(hT+\tau) - \mathbf{q}^r(hT+\tau)$. In order to write the control law in terms of the error, note that $\mathbf{q} = \mathbf{e} + \mathbf{q}^r$, hence (5.14) becomes

$$\mathbf{u}(hT+\tau) = \mathbf{A}_{\mathbf{e}(hT)+\mathbf{q}^r(hT)}(-k\mathbf{e}(hT) + \dot{\mathbf{q}}^r(hT+\tau)),$$

[1] Even $\mathbf{q}^r(hT+\tau)$ is continuously available, but it is convenient to pick it at instants hT in order to make the error $\mathbf{e}(hT)$ appear in the expression.

and the evolution of $\mathbf{q}(\,\cdot\,)$ is

$$\dot{\mathbf{q}}(hT+\tau) = \mathbf{A}^{-1}_{\mathbf{q}(hT+\tau)}\mathbf{u}(hT+\tau) = \mathbf{A}^{-1}_{\mathbf{e}(hT+\tau)+\mathbf{q}^r(hT+\tau)}\mathbf{u}(hT+\tau)$$

while the evolution of the error $\dot{\mathbf{e}} = \dot{\mathbf{q}} - \dot{\mathbf{q}}^r$ becomes

$$\begin{aligned}\dot{\mathbf{e}}(hT+\tau) = &-k\mathbf{A}^{-1}_{\mathbf{e}(hT+\tau)+\mathbf{q}^r(hT+\tau)}\mathbf{A}_{\mathbf{e}(hT)+\mathbf{q}^r(hT)}\mathbf{e}(hT)+\\ &+\left(\mathbf{A}^{-1}_{\mathbf{e}(hT+\tau)+\mathbf{q}^r(hT+\tau)}\mathbf{A}_{\mathbf{e}(hT)+\mathbf{q}^r(hT)} - \mathbf{I}\right)\dot{\mathbf{q}}^r(hT+\tau)\end{aligned}$$

Proposition 5.3.2. *The sampled control strategy of (5.14) does not always guarantee perfect tracking, i.e.*

$$\mathbf{q}(t) := \mathbf{q}^r(t) \not\Rightarrow \dot{\mathbf{e}}(t) = 0, \forall t \geq 0$$

Proof. If $\mathbf{q}(t) := \mathbf{q}^r(t)$ for all $t \geq 0$, then

$$\dot{\mathbf{e}}(hT+\tau) = \left(\mathbf{A}^{-1}_{\mathbf{q}^r(hT+\tau)}\mathbf{A}_{\mathbf{q}^r(hT)} - \mathbf{I}\right)\dot{\mathbf{q}}^r(hT+\tau)$$

which is in general not identically zero for all $h \in \mathbb{N}$ and $\tau \in [0,T)$ since $\mathbf{A}_{\mathbf{q}^r(hT)} \neq \mathbf{A}_{\mathbf{q}^r(hT+\tau)}$, unless $\mathbf{q}^r(t) = \mathbf{q}^r$, i.e. it is constant. ■

The previous result shows that if we simply sample the continuous-time law (5.5) in order to perform trajectory tracking, then this is not enough to guarantee perfect tracking.

Starting from this observation, we propose to slightly modify the feed-forward term in order to guarantee perfect (asymptotic) tracking. As before, we assume that a desired trajectory $\mathbf{q}^r(\,\cdot\,)$, $\dot{\mathbf{q}}^r(\,\cdot\,)$ is given as reference to the system and we choose a new control law for the SIKM controller:

$$\mathbf{u}(hT+\tau) = -k\mathbf{A}_{\mathbf{q}(hT)}(\mathbf{q}(hT) - \mathbf{q}^r(hT)) + \mathbf{A}_{\mathbf{q}^r(hT+\tau)}\dot{\mathbf{q}}^r(hT+\tau) \tag{5.15}$$

Notice that the difference from (5.14) is that the Jacobian of the feed-forward term is computed on $\mathbf{q}^r(hT+\tau)$ instead of $\mathbf{q}(hT)$. According to this choice, the error dynamics is

$$\begin{aligned}\dot{\mathbf{e}}(hT+\tau) &= -k\mathbf{A}^{-1}_{\mathbf{e}(hT+\tau)+\mathbf{q}^r(hT+\tau)}\mathbf{A}_{\mathbf{e}(hT)+\mathbf{q}^r(hT)}\mathbf{e}(hT)+\\ &\quad+\left(\mathbf{A}^{-1}_{\mathbf{e}(hT+\tau)+\mathbf{q}^r(hT+\tau)}\mathbf{A}_{\mathbf{q}^r(hT+\tau)} - \mathbf{I}\right)\dot{\mathbf{q}}^r(hT+\tau)\\ &\quad= \mathbf{f}(\mathbf{e}(hT+\tau), \mathbf{q}^r(hT+\tau), \dot{\mathbf{q}}^r(hT+\tau))\end{aligned} \tag{5.16}$$

This allows to prove the following property of the modified feedforward strategy:

Proposition 5.3.3. *Under the sampled control strategy of (5.15) the reference trajectory $\mathbf{q}^r(t)$ is an equilibrium trajectory i.e.*

$$\mathbf{q}(t) := \mathbf{q}^r(t) \Rightarrow \dot{\mathbf{e}}(t) = 0, \forall t \geq 0$$

Proof. Under the assumption that $\mathbf{q}(t) := \mathbf{q}^r(t)$ then $\mathbf{e}(t) = 0$ and the error dynamics

flow becomes:

$$\dot{\mathbf{e}}(hT+\tau) = -k\mathbf{A}^{-1}_{\mathbf{q}^r(hT+\tau)}\mathbf{A}_{\mathbf{q}^r(hT)}\underbrace{\mathbf{e}(hT)}_{=0} + \\ + \Big(\underbrace{\mathbf{A}^{-1}_{\mathbf{q}^r(hT+\tau)}\mathbf{A}_{\mathbf{q}^r(hT+\tau)}}_{=0} - \mathbf{I}\Big)\dot{\mathbf{q}}^r(hT+\tau) = 0,$$

for all $\tau \in [0,T)$ and $h \in \mathbb{N}$. ■

The previous proposition states that if $\mathbf{q}(0) = \mathbf{q}^r(0)$, i.e. if we start from a point on the reference trajectory, then $\mathbf{q}(t) = \mathbf{q}^r(t), \forall t > 0$, i.e. will perfectly track the trajectory. However, this does not guarantee that if $\mathbf{q}(0) \neq \mathbf{q}^r(0)$ then $\lim_{t\to\infty} \mathbf{q}(t) - \mathbf{q}^r(t) = 0$, i.e. if the proposed strategy guarantees asymptotic tracking, and if so, under what conditions in terms of the sampling period T and the feedback gain k. To answer these questions we need to study the evolution of the dynamical systems given in (5.16). The flow $\mathbf{f}(\,\cdot\,)$ is not continuous since the feedback term which depends on $\mathbf{e}(hT)$ is reset at every sampling period $t = hT$, therefore existence of global solution based on standard Lipschitz continuity cannot be invoked. However, by observing that within the intervals s.t. $\tau \in [0,T)$ the flow is Lipschitz continuous from Assumption 5.2.1, then if we can show that a solution $\mathbf{e}(hT+\tau)$ exists for any $\tau \in [0,T)$ and that the limit $\lim_{\tau\to T} \mathbf{e}(hT+\tau)$ exists and it is finite starting from any $\mathbf{e}(hT)$ satisfying Assumption 5.2.1, then the global existence is guaranteed also globally by patching together these pieces of trajectories. As so we will study the evolution of the dynamics $\dot{\mathbf{e}}(t) = \mathbf{f}(\mathbf{e}(t), \mathbf{q}^r(t), \dot{\mathbf{q}}^r(t); \mathbf{e}(hT))$ for $t = hT+\tau, \tau \in [0,T)$. Note that with a little abuse of notation, we explicitly exposed the dependence on the "intial" condition $\mathbf{e}(hT)$. In order to prove asymptotic stability we will use the following notion of contraction of the trajectory:

Definition 5.3.1. *For a fixed choice of T and k, we say that the flow $\mathbf{f}(\mathbf{e}(t), \mathbf{q}^r(t), \dot{\mathbf{q}}^r(t); \mathbf{e}(hT))$ is* ρ-monotonically contractive *if, under Assumption 5.2.1, for any $\|\mathbf{e}(hT)\| < d$ we have*

1. $\|e(hT+\tau)\| \leq \|e(hT)\|, \quad \tau \in [0,T)$
2. $\lim_{\tau\to T} \|e(hT+\tau)\| \leq \rho\|e(hT)\|, \ \rho \in [0,1)$

From this definition it easily follows the following lemma:

Lemma 5.3.1. *If the error flow defined in (5.16) is ρ-monotonically contractive and under Assumptions 5.2.1 then*

$$\|\mathbf{q}(t) - \mathbf{q}^r(t)\| \leq \rho^{\frac{t}{T}-1}\|\mathbf{q}(0) - \mathbf{q}^r(0)\|, \quad t \geq 0$$

The previous lemma guarantees exponential convergence of the system trajectory to the reference one. The property of ρ-monotonically contractivness is rather strong since it must hold for any segment of the trajectory of period T, however it allows to prove asymptoptic convergence.

We now provide two negative results in terms of ρ-monotonically contractivness

Proposition 5.3.4. *Under Assumption 5.2.1, there exists $k_1 > 0$ such that if at least one of following conditions are satisfied:*

1. $k < k_1$
2. $k > \frac{2}{T}$

then the error flow defined in (5.16) *is* ***not*** *ρ-monotonically contractive.*

Proof. 1) We prove the first statement by showing that the error norm initially increases for some initial conditions $\mathbf{e}(hT)$ and choice of reference trajectory $\mathbf{q}^r(t)$. Consider the following Lyapunov function

$$V(\tau) = \frac{1}{2}\|\mathbf{e}(hT+\tau)\|^2 \tag{5.17}$$

and its time derivative in $\tau = 0$:

$$\begin{aligned}
&\dot{V}(0) = \mathbf{e}(hT)^\top \dot{\mathbf{e}}(hT) = \\
&= \mathbf{e}(hT)^\top \Big(-k\mathbf{e}(hT) + (\mathbf{A}^{-1}_{\mathbf{e}(hT)+\mathbf{q}^r(hT)}\mathbf{A}_{\mathbf{q}^r(hT)} - \mathbf{I})\dot{\mathbf{q}}^r(hT)\Big) = \\
&= -k\|\mathbf{e}(hT)\|^2 + \mathbf{e}^\top(hT)(\mathbf{A}^{-1}_{\mathbf{e}(hT)+\mathbf{q}^r(hT)}\mathbf{A}_{\mathbf{q}^r(hT)} - \mathbf{I})\dot{\mathbf{q}}^r(hT)\Big)
\end{aligned}$$

Let us pick $\mathbf{e}(hT) \neq 0$ and $\mathbf{q}^r(hT)$ satisfying Assumption 5.2.1 for which

$$\|\mathbf{e}(hT)^\top(\mathbf{A}^{-1}_{\mathbf{e}(hT)+\mathbf{q}^r(hT)}\mathbf{A}_{\mathbf{q}^r(hT)} - \mathbf{I})\| =: a > 0$$

Such values clearly exists. Let us set

$$\dot{\mathbf{q}}^r(hT)^\top := \frac{\mathbf{e}(hT)^\top(\mathbf{A}^{-1}_{\mathbf{e}(hT)+\mathbf{q}^r(hT)}\mathbf{A}_{\mathbf{q}^r(hT)} - \mathbf{I})}{\|\mathbf{e}(hT)^\top(\mathbf{A}^{-1}_{\mathbf{e}(hT)+\mathbf{q}^r(hT)}\mathbf{A}_{\mathbf{q}^r(hT)} - \mathbf{I})\|} v_{max}$$

which clearly satisfies Assumption 5.2.1. Therefore

$$\dot{V}(0) = -k\|\mathbf{e}(hT)\|^2 + a v_{max}$$

If we define $k_1 := \frac{a v_{max}}{\|\mathbf{e}(hT)\|^2}$, then for any $k < k_1$ $\dot{V}(0) > 0$, hence there exists $0 < \bar{\tau} < T$ s.t. $\|\mathbf{e}(hT+\tau)\| > \|\mathbf{e}(hT)\|, \forall \tau \in (0, \bar{\tau})$, therefore it is not ρ-monotonically contractive.

2) In order to prove the second statement we first choose $\mathbf{q}^r(t) = \mathbf{q}^r, t \geq 0$ which satisfies Assumption 5.2.1, i.e. the reference trajectory is constant. As so the error dynamics reduces to

$$\dot{\mathbf{e}}(hT+\tau) = -k\mathbf{A}^{-1}_{\mathbf{e}(hT+\tau)+\mathbf{q}^r}\mathbf{A}_{\mathbf{e}(hT)+\mathbf{q}^r}\mathbf{e}(hT) := k\mathbf{g}(\mathbf{e}(hT+\tau))$$

The error trajectory can also be written in the form

$$\mathbf{e}(hT+\tau) = \mathbf{e}(hT) + k\int_{hT}^{hT+\tau} \mathbf{g}(\mathbf{e}(hT+\tau'))d\tau'$$

By using Taylor's theorem for multivariate functions with integral form of the remainder, it becomes

$$\begin{aligned}
&\mathbf{e}(hT+\tau) = \mathbf{e}(hT) + k\tau\,\mathbf{g}(\mathbf{e}(hT)) + \\
&+ k^2\tau^2 \int_0^1 (1-\varepsilon)\frac{\partial \mathbf{g}(\mathbf{e}(hT+\epsilon\tau))}{\partial \mathbf{e}}\,\mathbf{g}(\mathbf{e}(hT+\varepsilon\tau))d\varepsilon
\end{aligned}$$

$$= (1-k\tau)\mathbf{e}(hT) + k^2\tau^2\mathbf{r}(\mathbf{e}(hT+\tau);\mathbf{e}(hT)) \tag{5.18}$$

where $\mathbf{r}(\,\cdot\,)$ is the second-order reminder where we made explicit the dependence on $\mathbf{e}(hT)$. Under Assumption 5.2.1, the functions $\mathbf{g}$ and $\frac{\partial \mathbf{g}}{\partial \mathbf{e}}$ are continuously differentiable. Moreover their arguments are defined in a compact set and have the additional properties that $\mathbf{g}(\mathbf{e}(hT+\tau)) = 0, \frac{\partial \mathbf{g}(hT+\tau)}{\partial \mathbf{e}} = \mathbf{0},\ \forall \tau \in [0,T]$ if $\mathbf{e}(hT) = \mathbf{0}$. As a result, if we apply Lemma B.2.3 reported in Appendix, there must exist $\delta > 0$ such that

$$\|\mathbf{r}(\mathbf{e}(hT+\tau);\mathbf{e}(hT))\| \leq \delta \|\mathbf{e}(hT)\|^2, \quad \forall \tau \in [0,T), \forall \mathbf{e}(hT)$$

we now use the reverse triangle inequality($|\|\mathbf{x}\| - \|\mathbf{y}\|| \leq \|\mathbf{x}-\mathbf{y}\|$ and the previous inequality to get:

$$\begin{aligned}
\|\mathbf{e}(hT+\tau)\| &= \|(1-k\tau)\mathbf{e}(hT) + k^2\tau^2\mathbf{r}(\mathbf{e}(hT+\tau);\mathbf{e}(hT))\| \\
&= \|(1-k\tau)\mathbf{e}(hT) - (-k^2\tau^2\mathbf{r}(\mathbf{e}(hT+\tau);\mathbf{e}(hT)))\| \\
&\geq \Big|\|(1-k\tau)\mathbf{e}(hT)\| - k^2\tau^2\|\mathbf{r}(\mathbf{e}(hT+\tau);\mathbf{e}(hT))\|\Big| \\
&\geq \max\{0\,, |1-k\tau| - k^2\tau^2\delta\|\mathbf{e}(hT)\|\}\|\mathbf{e}(hT)\|
\end{aligned}$$

Now if $kT > 2$, there exist $\epsilon > 0$ and $\bar{\tau} \in (0,T)$ s.t. $k\bar{\tau} = 2+\epsilon$. Then we can choose $\mathbf{e}(hT)$ s.t. $\|\mathbf{e}(hT)\| = \frac{\epsilon}{2k\bar{\tau}^2\delta}$, hence

$$\|\mathbf{e}(hT+\bar{\tau})\| \geq (1+\epsilon-\frac{\epsilon}{2})\|\mathbf{e}(hT)\| \geq \|\mathbf{e}(hT)\|$$

implying the system not to be ρ-monotonically contractive. ■

The previous proposition basically states that there are choices of T and k for which the dynamics is surely not ρ-monotonically contractive. However, we can prove that there exist pairs of T and k for which the dynamics is ρ-monotonically contractive.

Proposition 5.3.5. *Under Assumption 5.2.1, there exists $k_2 > 0$ such that for any $k > k_2$ there exists $T_c(k)$ such that for all $T < T_c(k)$ the error flow defined in (5.16) is ρ-monotonically contractive.*

Proof. Let us consider the Lyapunov function defined in (5.17) and its time derivative in $\tau = 0$

$$\dot{V}(0) = -k\|\mathbf{e}(hT)\|^2 + \mathbf{e}^\top(hT)(\mathbf{A}^{-1}_{\mathbf{e}(hT)+\mathbf{q}^r(hT)}\mathbf{A}_{\mathbf{q}^r(hT)} - \mathbf{I})\dot{\mathbf{q}}^r(hT)$$

Note that the quantity $\mathbf{A}^{-1}_{\mathbf{e}(hT)+\mathbf{q}^r(hT)}\mathbf{A}_{\mathbf{q}^r(hT)} - \mathbf{I} = \mathbf{0}, \forall \mathbf{q}^r(hT)$ if $\mathbf{e}(hT) = 0$. Moreover being $\mathbf{e}(hT)$ and $\mathbf{q}^r(hT)$ in a compact set according to Assumption 5.2.1, we must have $\|\mathbf{A}^{-1}_{\mathbf{e}(hT)+\mathbf{q}^r(hT)}\mathbf{A}_{\mathbf{q}^r(hT)} - \mathbf{I}\| \leq \delta\|\mathbf{e}(hT)\|$ for some $\delta > 0$, therefore we have:

$$\begin{aligned}
\dot{V}(0) &\leq -k\|\mathbf{e}(hT)\|^2 + \delta\|\mathbf{e}(hT)\|^2 v_{max} = -(k-\delta v_{max})\|\mathbf{e}(hT)\|^2 \\
&= -2(k-\delta v_{max})V(0)
\end{aligned}$$

Let $k_2 := \delta v_{max}$, then for each $k > k_2$, there exists $\xi \in (0, k-\delta v_{max})$ and $T_c(k)$ s.t.

$$\begin{aligned}
&V(\tau) \leq e^{-2\xi\tau}V(0), \tau \in [0, T_c(k)) \\
&\implies \|\mathbf{e}(hT+\tau)\| \leq e^{-\xi\tau}\|\mathbf{e}(hT)\|,\ \tau \in [0, T_c(k))
\end{aligned}$$

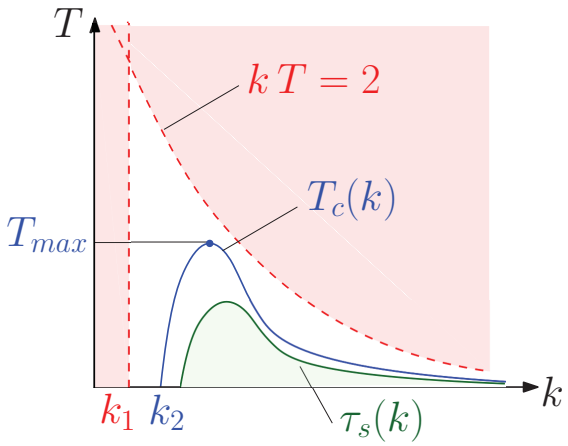


Figure 5.3. Representation of the curve $T_c(k)$ defined in Proposition 5.3.5 and $\tau_s(k)$ that is obtained using upper bounds on the error norm (see Proposition 5.4.1), hence it delimits a smaller stability region.

As a consequence for any $T < T_c(k)$ if we define $\rho := e^{-T\xi} < 1$ then we have

$$\|\mathbf{e}(hT+\tau)\| \leq \|\mathbf{e}(hT)\|,\ \tau \in [0,T)$$
$$\lim_{\tau \to T} \|\mathbf{e}(hT+\tau)\| \leq \rho \|\mathbf{e}(hT)\|$$

i.e. the systems is ρ-monotonically contractive. ■

The previous theorem guarantees the existence of a set of pairs T and k s.t. the system is ρ-monotonically contractive. By combining this theorem with the previous one, we can already see that $\lim_{k\to\infty} T_c(k) = 0$ and that there exists $T_{max} > 0$ such that $T_c(k) < T_{max}, \forall k > k_2$. In Figure 5.3 these quantities are depicted. From Proposition 5.3.4 we know that the error flow defined in (5.16) is **not** ρ-monotonically contractive for values of (k,T) in the red area. Instead, the curve $T_c(k)$ introduced in Proposition 5.3.5 delimits the area where the flow (5.16) is ρ-monotonically contractive. In the next Section we show how to obtain an estimate of $T_c(k)$. Actually, since our analysis is based on upper bounds on the tracking error norm, then we provide an under-estimate that is represented by the green line $\tau_s(k)$.

5.4 Stability and convergence rate bounds

The theorems introduced in the previous Section are of existential type, i.e. do not provide explicit computation of the set of the pairs k and T for which the system is ρ-monotonically contractive nor the convergence rate ρ. In this section we want to find explicit bounds for the "stability" set (green area in Figure 5.3 delimited by $\tau_s(k)$) and for the convergence rate, that is the region where the error norm decreases and for which gains and sampling times it goes fastest to zero.

Remark 5.4.1. When we talk about stability, we actually imply that the flow (5.16) is ρ-monotonically contractive.

Now we want to obtain an explicit expression for $\tau_s(k)$:

Proposition 5.4.1. *Given the system (5.16), where $\|\mathbf{e}(\,\cdot\,)\|$ is the tracking error and T the sampling time, it is possible to find an explicit upper bound of the decreasing norm of the error:*

$$\|\mathbf{e}(hT+\tau)\| \leq z\Big(k,\tau;\mu,\alpha,\gamma_1,\gamma_2\Big)\,\cdot\,\|\mathbf{e}(hT)\| \tag{5.19}$$

for $h\in\mathbb{N}$, $\tau\in(0,\tau_s(k))$, and

$$z\Big(k,\tau;\mu,\alpha,\gamma_1,\gamma_2\Big)=|1-k\tau|+\tau\alpha+\tau^2(k^2\mu+k\gamma_1+\gamma_2) \tag{5.20}$$

where $\alpha,\mu,\gamma_1,\gamma_2\geq 0$ are defined in the Proof and can be estimated following the procedure described in Section 5.4.1.

Proof. Note that the solution of (5.16) can be written as

$$\mathbf{e}(hT+\tau)=\mathbf{e}(hT)+\int_{hT}^{hT+\tau}\mathbf{f}(\mathbf{e}(\,\cdot\,),\mathbf{q}^r(\,\cdot\,),\dot{\mathbf{q}}^r(\,\cdot\,))\Big|_{hT+\tau'}d\tau'$$

By using Taylor's theorem for multivariate functions with integral form of the remainder, it becomes

$$\begin{aligned}
&\mathbf{e}(hT+\tau)=\mathbf{e}(hT)+\tau\,\mathbf{f}(\mathbf{e}(hT),\mathbf{q}^r(hT),\dot{\mathbf{q}}^r(hT))+\\
&+\tau^2\int_0^1(1-\varepsilon)\cdot\left[\frac{\partial\mathbf{f}(\mathbf{e}(\,\cdot\,),\mathbf{q}^r(\,\cdot\,),\dot{\mathbf{q}}^r(\,\cdot\,))}{\partial\mathbf{e}(\,\cdot\,)}\,\mathbf{f}(\mathbf{e}(\,\cdot\,),\mathbf{q}^r(\,\cdot\,),\dot{\mathbf{q}}^r(\,\cdot\,))+\right.\\
&+\frac{\partial\mathbf{f}(\mathbf{e}(\,\cdot\,),\mathbf{q}^r(\,\cdot\,),\dot{\mathbf{q}}^r(\,\cdot\,))}{\partial\mathbf{q}^r(\,\cdot\,)}\,\dot{\mathbf{q}}^r(\,\cdot\,)+\\
&\left.+\frac{\partial\mathbf{f}(\mathbf{e}(\,\cdot\,),\mathbf{q}^r(\,\cdot\,),\dot{\mathbf{q}}^r(\,\cdot\,))}{\partial\dot{\mathbf{q}}^r(\,\cdot\,)}\,\ddot{\mathbf{q}}^r(\,\cdot\,)\right]_{hT+\varepsilon\tau}d\varepsilon
\end{aligned} \tag{5.21}$$

This form of the solution allows to easily find an upper bound of $\|\mathbf{e}(hT+\tau)\|$, $\tau\in(0,\tau_s(k))$ by acting on the single terms of (5.21). Hence we can compute a more precise estimate of the convergence rate, that is how much quickly $\|\mathbf{e}(hT+\tau)\|$, $\tau\in(0,\tau_s(k))$ decreases with respect to $\|\mathbf{e}(hT)\|$. Moreover, the following computations will be useful to find for which values of (k,T) the system is stable and the tracking error converges to zero. First, we observe that

$$\begin{aligned}
&\mathbf{f}(\mathbf{e}(hT),\mathbf{q}^r(hT),\dot{\mathbf{q}}^r(hT))=\\
&=-k\mathbf{e}(hT)+\left(\mathbf{A}^{-1}_{\mathbf{e}(hT)+\mathbf{q}^r(hT)}\mathbf{A}_{\mathbf{q}^r(hT)}-\mathbf{I}\right)\,\dot{\mathbf{q}}^r(hT)
\end{aligned} \tag{5.22}$$

Notice that, by using Lemma B.2.2 reported in Appendix we can derive an upper bound for the norm of the second term:

$$\left\|\left(\mathbf{A}^{-1}_{\mathbf{e}(hT)+\mathbf{q}^r(hT)}\mathbf{A}_{\mathbf{q}^r(hT)}-\mathbf{I}\right)\,\dot{\mathbf{q}}^r(hT)\right\|\leq\alpha\|\mathbf{e}(hT)\|$$

where $\alpha:=a\,\cdot\,v_{max}$. Moreover, from (5.16),

$$\begin{aligned}
&\|\mathbf{f}(\mathbf{e}(hT+\tau),\mathbf{q}^r(hT+\tau),\dot{\mathbf{q}}^r(hT+\tau))\|\leq\\
&\leq\|-k\mathbf{A}^{-1}_{\mathbf{e}(hT+\tau)+\mathbf{q}^r(hT+\tau)}\mathbf{A}_{\mathbf{e}(hT)+\mathbf{q}^r(hT)}\mathbf{e}(hT)\|+
\end{aligned}$$

$$+\|\Big(\mathbf{A}^{-1}_{\mathbf{e}(hT+\tau)+\mathbf{q}^r(hT+\tau)}\mathbf{A}_{\mathbf{q}^r(hT+\tau)}-\mathbf{I}\Big)\dot{\mathbf{q}}^r(hT+\tau)\| \tag{5.23}$$

Now, observe that both the terms in (5.23) are continuously differentiable functions on a compact set; in addition, they are equal to zero respectively if $\mathbf{e}(hT)=\mathbf{0}$ and $\mathbf{e}(hT+\tau)=\mathbf{0}$, then we can apply Lemmas B.2.1-B.2.2 reported in Appendix and obtain $\|k\mathbf{A}^{-1}_{\mathbf{e}(hT+\tau)+\mathbf{q}^r(hT+\tau)}\mathbf{A}_{\mathbf{e}(hT)+\mathbf{q}^r(hT)}\mathbf{e}(hT)\| \leq k\cdot b\|\mathbf{e}(hT)\|,\ b>0$ and $\|\Big(\mathbf{A}^{-1}_{\mathbf{e}(hT+\tau)+\mathbf{q}^r(hT+\tau)}\mathbf{A}_{\mathbf{q}^r(hT+\tau)}-\mathbf{I}\Big)\dot{\mathbf{q}}^r(hT+\tau)\| \leq c v_{max}\|\mathbf{e}(hT)\|,\ c>0$. Notice that we used Proposition 5.3.5 and point 1) from Definition 5.3.1. Given these observations it is possible to rewrite (5.23):

$$\|\mathbf{f}(\mathbf{e}(\ \cdot\ ),\mathbf{q}^r(\ \cdot\ ),\dot{\mathbf{q}}^r(\ \cdot\ ))\|_{hT+\tau} \leq (kb+cv_{max})\|\mathbf{e}(hT)\| \tag{5.24}$$

In addition, since $\mathbf{f}(\mathbf{e}(\ \cdot\ ),\mathbf{q}^r(\ \cdot\ ),\dot{\mathbf{q}}^r(\ \cdot\ ))$ is a differentiable function with respect to all its variables on the compact set $\mathcal{B}_r(\mathbf{e}(\ \cdot\ ))$, then we can derive the following properties based on (5.22) :

$$\begin{aligned}
\left\|\frac{\partial\mathbf{f}(\mathbf{e},\mathbf{q}^r,\dot{\mathbf{q}}^r)}{\partial\mathbf{e}}\right\|_{(hT+\tau)} &\leq (k\,d+g\,v_{max})\|\mathbf{e}(hT)\| \\
\left\|\frac{\partial\mathbf{f}(\mathbf{e},\mathbf{q}^r,\dot{\mathbf{q}}^r)}{\partial\mathbf{q}^r}\right\|_{(hT+\tau)} &\leq (k\,l+m\,v_{max})\|\mathbf{e}(hT)\| \\
\left\|\frac{\partial\mathbf{f}(\mathbf{e},\dot{\mathbf{q}}^r,\dot{\mathbf{q}}^r)}{\partial\dot{\mathbf{q}}^r}\right\|_{(hT+\tau)} &\leq kn\|\mathbf{e}(hT)\|
\end{aligned} \tag{5.25}$$

where $d,l,g,m,n>0$.

Given the previous properties, we can easily derive the following upper bound for the tracking error:

$$\begin{aligned}
&\|\mathbf{e}(hT+\tau)\| \leq \\
&\leq (|1-k\tau|+\tau\alpha)\|\mathbf{e}(hT)\|+\tau^2\int_0^1\Big[(1-\varepsilon)(kd+gv_{max})(kb+cv_{max})\|\mathbf{e}(hT)\|^2+ \\
&+(kl+mv_{max})v_{max}\|\mathbf{e}(hT)\|+kna_{max}\|\mathbf{e}(hT)\|\Big]d\varepsilon \\
&=(|1-k\tau|+\tau\alpha)\|\mathbf{e}(hT)\|+\tau^2\int_0^1\Big[(1-\varepsilon)(k^2db+kdcv_{max}+kgbv_{max}+gcv^2_{max})\|\mathbf{e}(hT)\|^2+ \\
&+(klv_{max}+mv^2_{max})\|\mathbf{e}(hT)\|+kna_{max}\|\mathbf{e}(hT)\|\Big]d\varepsilon= \\
&=(|1-k\tau|+\tau\alpha)\|\mathbf{e}(hT)\|+\tau^2\int_0^1(1-\varepsilon)d\varepsilon\Big[k^2db\|\mathbf{e}(hT)\|+kdcv_{max}\|\mathbf{e}(hT)\|+ \\
&+kgbv_{max}\|\mathbf{e}(hT)\|+gcv^2_{max}\|\mathbf{e}(hT)\|+klv_{max}+mv^2_{max}+kna_{max}\Big]\|\mathbf{e}(hT)\|d\varepsilon= \\
&=(|1-k\tau|+\tau\alpha)\|\mathbf{e}(hT)\|+ \\
&+\frac{1}{2}\tau^2\Big[k^2db\|\mathbf{e}(hT)\|+k\Big(v_{max}(dc\|\mathbf{e}(hT)\|+gb\|\mathbf{e}(hT)\|+l)+na_{max}\Big)+ \\
&+gcv^2_{max}\|\mathbf{e}(hT)\|+mv^2_{max}\Big]\|\mathbf{e}(hT)\|= \\
&=\Big[|1-k\tau|+\tau\alpha+\tau^2(k^2\mu+k\gamma_1+\gamma_2)\Big]\|\mathbf{e}(hT)\|
\end{aligned}$$

where

$$\begin{aligned}\alpha &:= a\, v_{max}\\ \mu &:= db\|\mathbf{e}(hT)\|\\ \gamma_1 &:= v_{max}(dc\,\|\mathbf{e}(hT)\| + gb\,\|\mathbf{e}(hT)\| + l) + n a_{max}\\ \gamma_2 &:= (gc\|\mathbf{e}(hT)\| + m)v_{max}^2\end{aligned}$$

Finally, we can write that

$$\|\mathbf{e}(hT+\tau)\| \leq z\Big(k,\tau;\mu,\gamma_1,\gamma_2\Big)\,\|\mathbf{e}(hT)\| \tag{5.26}$$

Given the properties obtained in (5.22),(5.24) and (5.25), we derive the overall bound defined in (5.19):

$$\|\mathbf{e}(hT+\tau)\| \leq \Big(|1-k\tau| + \alpha\tau + \tau^2(k^2\mu + k\gamma_1 + \gamma_2)\Big)\|\mathbf{e}(hT)\|$$

where, given $a > 0$ defined in the proof of Proposition 5.3.4,

$$\begin{aligned}\alpha &:= a\, v_{max}\\ \mu &:= b\|\mathbf{e}(hT)\|\\ \gamma_1 &:= (c+gb)v_{max}\|\mathbf{e}(hT)\| + n\, a_{max}\\ \gamma_2 &:= g\, c\, v_{max}\|\mathbf{e}(hT)\| + m\, v_{max}\end{aligned}$$

■

Notice that we obtained an expression for the convergence rate but the value of the parameters $\mu, \alpha, \gamma_1, \gamma_2$ is not known yet. Thus, in the next Section we provide a numerical procedure to find an estimate of those parameters.

5.4.1 Estimate of the parameters $\mu, \alpha, \gamma_1, \gamma_2$

In the previous Section we defined the function $z(k,\tau;\mu,\alpha,\gamma_1,\gamma_2)$ that describes the convergence rate of the tracking error. Now we would like to estimate this rate and choose the values (k,T) which guarantee the fastest one. To do that we need to know the value of the parameters $\theta := (\mu,\alpha,\gamma_1,\gamma_2)$, hence in the following we provide a numerical procedure to compute them. Proposition 5.4.1 implies that the following set is not empty:

$$\Theta := \Big[\theta > 0\,|\,\|\mathbf{e}(hT+\tau)\| \leq z\Big(k,\tau;\theta\Big)\|\mathbf{e}(hT)\|, \forall \mathbf{q}^r(\,\cdot\,), \mathbf{q}(0)\Big]$$

where the inequality is to be intended componentwise. Ideally, we would like to pick the smallest possible values for the entries of the vector θ in order to get the largest possible set of feasible pairs (k,T) described in the next section. For example, one possible way is to choose θ as follows

$$\vartheta := \arg\min_{\theta\in\Theta} \|\theta\|$$

although other (possibly weighed) norms, such as $\|\theta\|_1$ or $\|\theta\|_\infty$, could be adopted. Note that such ϑ exists since $\mathbf{q}^r(\,\cdot\,)$ and $\mathbf{q}(0)$ belongs to a compact set, however it cannot

numerically be computed since one would have to check it for all pairs (k,T) and points $\mathbf{q}^r(t)$ and $\mathbf{q}(0)$. We will then propose an approximate strategy to estimate ϑ by sampling $\mathbf{q}^r(t)$ and $\mathbf{q}(0)$ from their compact set for different values of (k,T) and run simulations to get a set of samples $\{(k_i, T_i, \mathbf{e}^i_{h+1}, \mathbf{e}^i_h)\}_{i=1}^N$, where N is the sample set size. Let us define the following quantities:

$$\begin{aligned} y_i &:= \|\mathbf{e}^i_{h+1}\|, \\ a_i^T &= [T_i^2 \;\; k_i^2 \;\; T_i \;\; T_i^2 k_i \;\; T_i^2] \|\mathbf{e}^i_h\| \\ b_i &= |1 - k_i T_i| \cdot \|\mathbf{e}^i_h\| \end{aligned}$$

then the inequality $\|\mathbf{e}^i_{h+1}\| \leq z\Big(k_i, T_i; \theta\Big)\|\mathbf{e}^i_h\|$ can be rewritten as $y_i \leq a_i^T \theta + b_i$. Based on these sampled trajectories we then propose to solve the following quadratic programming (QP) problem:

$$\begin{aligned} \widehat{\theta}_N := \arg\min_{\theta} \quad & \|\theta\|^2 \\ s.t. \quad & \theta \geq 0, \\ & y_i \leq A_i \theta + b_i, \quad i = 1, \ldots, N \end{aligned} \tag{5.27}$$

Unfortunately it is possible that $\widehat{\theta}_N \notin \Theta$ since we are checking the inequality $\|\mathbf{e}(hT + \tau)\| \leq z\Big(k, \tau; \theta\Big)\|\mathbf{e}(hT)\|$ only over a finite number of points, however we expect that $\lim_{N\to\infty} \widehat{\theta}_N = \vartheta$ if we sample enough different $\mathbf{q}^r(\,\cdot\,)$ and $\mathbf{q}(0)$. A formal proof of this claim is, however, not trivial. Moreover there might be alternative numerical strategies to compute better estimates of ϑ or alternative parameter choices in the set Θ. These problems go beyond the scope of this work, however we will show in the simulation section as the strategy proposed here is nonetheless effective.

5.4.2 Analysis of the function $z\Big(k, \tau; \mu, \gamma_1, \gamma_2\Big)$

Recall that the function $z\Big(k, \tau; \mu, \gamma_1, \gamma_2\Big)$ in (5.19) describes for which (k, τ) and how fast the norm of the tracking error decreases starting from the last sampled measurement. In this Section we analyse the function in order to compute the convergence rate of the error and to define the stability region (green area in Figure 5.3), that is where $z\Big(k, \tau; \mu, \gamma_1, \gamma_2\Big) < 1$. Notice that if we fix k, then there exists a value of τ where the system reaches the fastest convergence rate and vice-versa. These values will be called optimal time $\tau_o(k)$ or $k_o(\tau)$ and they obviously lie in the green area in Figure Figure 5.3. Notice that (5.20) can be rewritten as[2]:

$$z\Big(k, \tau\Big) = \begin{cases} z^-(k, \tau) & \text{if } k\tau < 1 \\ z^+(k, \tau) & \text{if } k\tau > 1 \end{cases}$$

where

$$z^-(k, \tau) = 1 + \tau(\alpha - k) + \tau^2(k^2\mu + k\gamma_1 + \gamma_2)$$

[2]In the following, we will use the short notation $z\Big(k, \tau\Big) := z\Big(k, \tau; \mu, \gamma_1, \gamma_2\Big)$

$$z^+(k,\tau) = -1 + \tau(\alpha + k) + \tau^2(k^2\mu + k\gamma_1 + \gamma_2)$$

We want to find the stability set, that is

$$\mathcal{U} := \{(k,\tau) | z(k,\tau;\mu,\alpha,\gamma_1,\gamma_2\Big)) < 1\}$$

thast is defined in the following Proposition.

Proposition 5.4.2. *The stability set of* $z\Big(k,\tau;\mu,\gamma_1,\gamma_2\Big)$ *is*

$$\mathcal{U} = \{\alpha < k < +\infty, 0 < \tau < \tau_s(k)\}$$

where $\alpha = k_{min}$ *was defined in Proposition 5.4.1 and*

$$\tau_s(k) = \begin{cases} \tau_{s_1}(k) & \alpha < k < \bar{k} \quad \text{if } \mu > 1 \\ \tau_{s_1}(k) & \alpha < k < \bar{k} \quad \text{if } \mu < 1 \\ \tau_{s_2}(k) & k > \bar{k} \quad \text{if } \mu < 1 \end{cases}$$

where $\tau_{s_1} := \frac{k-\alpha}{k^2\mu + k\gamma_1 + \gamma_2}$, $\tau_{s_2} := \frac{-(\alpha+k) + \sqrt{(\alpha+k)^2 + 8(k^2\mu + k\gamma_1 + \gamma_2)}}{2(k^2\mu + k\gamma_1 + \gamma_2)}$, $\bar{k} = \frac{\alpha + \gamma_1 + \sqrt{(\alpha+\gamma_1)^2 + 4\gamma_2(1-\mu)}}{2(1-\mu)}$.

The proof is reported in Appendix B.4.1.

Moreover, if we fix k, then we define respectively the time and convergence rate as functions of k and of the parameters $\mu, \alpha, \gamma_1, \gamma_2$:

$$\tau_o(k;\mu,\alpha,\gamma_1,\gamma_2) := \text{argmin}_\tau z(k,\tau(k);\mu,\alpha,\gamma_1,\gamma_2) \tag{5.28}$$

$$\rho_o(k;\mu,\alpha,\gamma_1,\gamma_2) := z(k,\tau_o(k);\mu,\alpha,\gamma_1,\gamma_2) \tag{5.29}$$

where τ_o is the time when $\|\mathbf{e}(hT+\tau)\|$, $0 \le \tau \le T$ is closest to the origin, hence it corresponds to the fastest convergence rate ρ_o of the error.

Proposition 5.4.3. *Let us assume to fix* k*, then the optimal time* $\tau_o(k) := \tau_o(k;\mu,\alpha,\gamma_1,\gamma_2)$ *defined in (5.28) can be computed as:*

$$\tau_o(k) = \begin{cases} \tau_{o_1}(k) & \text{for } \alpha < k < \bar{\bar{k}} \quad \text{if } \mu > \frac{1}{2} \\ \tau_{o_1}(k) & \text{for } \alpha < k < \bar{\bar{k}} \quad \text{if } \mu < \frac{1}{2} \\ \frac{1}{k} & \text{for } k > \bar{\bar{k}} \quad \text{if } \mu < \frac{1}{2} \end{cases}$$

where $\tau_{o_1}(k) := \frac{k-\alpha}{2(k^2\mu + k\gamma_1 + \gamma_2)}$ *and* $\bar{\bar{k}} = \frac{\alpha + 2\gamma_1 + \sqrt{(\alpha+2\gamma_1)^2 + 8\gamma_2(1-2\mu)}}{2(1-2\mu)}$.

The corresponding convergence rate $\rho_o(k) := \rho_o(k;\mu,\alpha,\gamma_1,\gamma_2)$ *is:*

$$\rho_o(k) = \begin{cases} \rho_{k_1}(k) & & \text{if } \mu > \frac{1}{2} \\ \rho_{k_1}(k) & \text{for } \alpha < k < \bar{\bar{k}} & \text{if } \mu < \frac{1}{2} \\ \rho_{k_2}(k) & \text{for } k > \bar{\bar{k}} & \text{if } \mu < \frac{1}{2} \end{cases}$$

where $\rho_{k_1}(k) := 1 - \frac{(\alpha-k)^2}{4(k^2\mu + k\gamma_1 + \gamma_2)}$ *and* $\rho_{k_2}(k) := \mu + \frac{1}{k}(\alpha + \gamma_1) + \frac{\gamma_2}{k^2}$.

The proof is reported in Appendix B.4.2.

The same quantities can be found as functions of τ:

$$k_o(\tau;\mu,\alpha,\gamma_1,\gamma_2) := \text{argmin}_k z(k(\tau),\tau;\mu,\alpha,\gamma_1,\gamma_2) \tag{5.30}$$

$$\rho_o(\tau;\mu,\alpha,\gamma_1,\gamma_2) := z(k_o(\tau),\tau;\mu,\alpha,\gamma_1,\gamma_2) \tag{5.31}$$

Proposition 5.4.4. *Let us assume to fix τ, then the optimal $k_o(\tau) := k_o(\tau;\mu,\alpha,\gamma_1,\gamma_2)$, defined in (5.30), can be computed as:*

$$k_o(\tau) = \begin{cases} \frac{1-\tau\gamma_1}{2\tau\mu} & \text{for } \tau < \frac{1}{\gamma_1} \quad \text{if } \mu > \frac{1}{2} \\ \frac{1-\tau\gamma_1}{2\tau\mu} & \text{for } \tau_{m_k} < \tau < \tau_{M_k} \quad \text{if } \mu < \frac{1}{2} \\ \frac{1}{\tau} & \text{for } 0 < \tau < \tau_{m_k} \quad \text{if } \mu < \frac{1}{2} \end{cases}$$

where $\tau_{m_k} := \frac{1-2\mu}{\gamma_1}$, $\tau_{M_k} := \frac{1}{\gamma_1}$ and the corresponding convergence rate $\rho_o(\tau) := \rho_o(\tau;\mu,\alpha,\gamma_1,\gamma_2)$ is:

$$\rho_o(\tau) = \begin{cases} \rho_{\tau_1}(\tau) & \text{for } \tau < \tau_{v_2}^- \quad \text{if } \mu > \frac{1}{2} \\ \rho_{\tau_1}(\tau) & \text{for } \tau_{m_\tau} < \tau < \tau_{M_\tau} \quad \text{if } \mu < \frac{1}{2} \\ \rho_{\tau_2}(\tau) & \text{for } 0 < \tau < \tau_{m_\tau} \quad \text{if } \mu < \frac{1}{2} \end{cases}$$

where $\tau_{m_\tau} := \min\{\frac{1-2\mu}{\gamma_1}, \tau_{v_2}^-\}$, $\tau_{M_\tau} := \max\{\frac{1-2\mu}{\gamma_1}, \tau_{v_2}^-\}$, $\rho_{\tau_1}(\tau) := \frac{(-\gamma_1^2+4\gamma_2\mu)\tau^2+2(\gamma_1+2\alpha\mu)\tau+4\mu-1}{4\mu}$, $\rho_{\tau_2}(\tau) := \gamma_2\tau^2+(\alpha+\gamma_1)\tau+\mu$ and $\tau_{v_2}^- = \frac{-(\gamma_1+\alpha\mu)+\sqrt{(\gamma_1+\alpha\mu)^2+(-\gamma_1^2+4\gamma_2\mu)}}{-\gamma_1^2+4\gamma_2\mu}$.

The proof is reported in Appendix B.4.3.

Figures 5.4a, 5.4b and 5.4c depict the quantities that we defined above: in particular, $\tau_s(k)$ is defined in Proposition 5.4.2; then $\tau_o(k)$ in Proposition 5.4.3 and $\tau(k_o)$ was obtained by deriving τ from the definition of $k_o(\tau)$ in Proposition 5.4.4). Depending on the parameter that we fix, we can choose the optimal gain or sampling time in order to obtain the fastest convergence rate. We chose three examples which correspond to $\mu < \frac{1}{2}$, $\frac{1}{2} < \mu < 1$ and $\mu > 1$ in order to represent the three different scenarios of the definitions given in the previous propositions. Note that $\tau = \frac{1}{k}$ represents the value where $z^-(k,\tau) = z^+(k,\tau)$. Figure 5.5 instead represents the convergence rate $\rho_o(k)$ as a function of k (figure above) and time τ (figure below). Observe that the convergence rate is always smaller than one, hence

$$\|\mathbf{e}(hT+\tau)\| \leq \rho_o\|\mathbf{e}(hT)\| \leq \|\mathbf{e}(hT)\|, \ \tau \in [0,T]$$

Observe that the instant τ_{CR} is the critical time corresponding to the maximum of $\tau_s(k)$ and it corresponds to the slowest convergence rate.

Remark. Consider (5.19) and suppose that a constant reference is assigned, that is $\frac{d^{(n)}\mathbf{q}^r(t)}{dt^{(n)}} \equiv \mathbf{0}$ for $n \geq 1$. Then $\alpha = \gamma_1 = \gamma_2 = 0$. As a consequence $z(k,\tau;\mu)$ is exactly the function $g(\tau;\mu)\Big|_{\tau=k\tau}$ that we found in Chapter 3. In addition, in that case for any sampling time $T > 0$ we could find a gain small enough to make the system converge to the desired configuration. In this scenario, instead, there exists a maximum time $\tau_{CR} > 0$ that can be chosen.

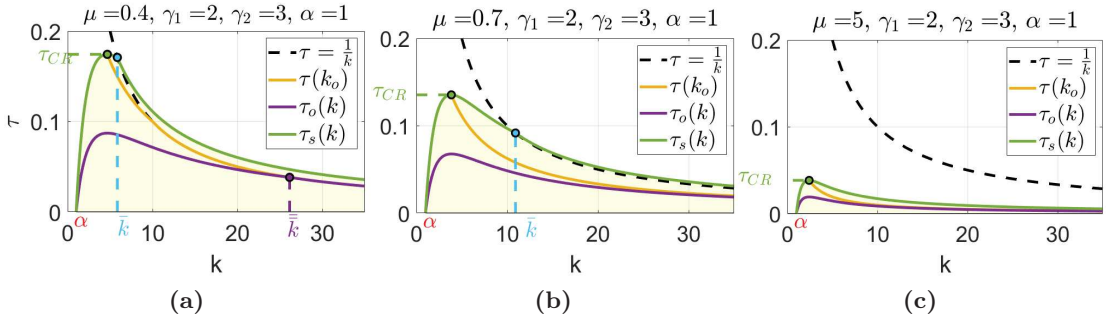


Figure 5.4. Representation of the quantities $\tau_s(k)$, $\tau_o(k)$, $\tau(k_o)$ defined in Proposition 5.4.2-5.4.4.

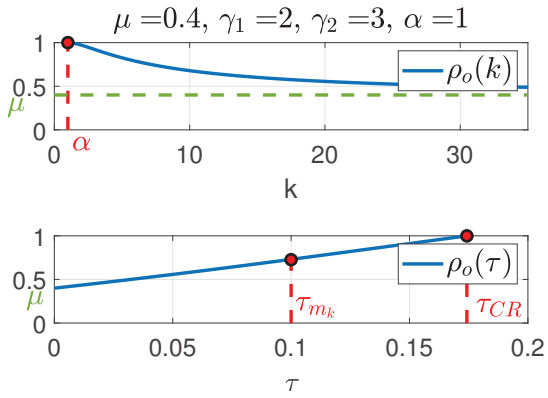


Figure 5.5. Representation of the convergence rate as a function of the gain k and the time τ.

5.5 Simulation Results

In this section we compare the different techniques described before to perform trajectory tracking. Recalling the SIKM controller law $\mathbf{u} = \mathbf{u}_k + \mathbf{u}_{ff}$, in the following we refer to the different strategies used in simulation with the numbers $1)-4)$ as shown in Tab.5.1. They differ both in the computation (off-line (k_{OFF}) or on-line (k_{ON})) of the gain in $\mathbf{u}_k(hT)$ and in the point where the Jacobian in $\mathbf{u}_{ff}(hT+\tau)$ is evaluated: or on the last received measurement $\mathbf{q}_{hT}$ or on the desired configuration $\mathbf{q}^r_{hT+\tau}$. In particular our analysis started with point-stabilization control (Section 5.3.2) with off-line computation of the gain and we saw that it does not guarantee zero-tracking error, even though it is possible to find an upper bound to it. We also noticed in the second part of Section 5.3.1 that a better performance could be intuitively obtained by designing a sequence of time-varying gains k_h (strategy 4). Even if we cannot derive theoretical results in this

$\mathbf{u}_{hT+\tau} = \mathbf{u}_k(hT) + \mathbf{u}_{ff(hT+\tau)}$	$\mathbf{u}_k(hT)$	$\mathbf{u}_{ff}(hT+\tau)$
1)	$-k_{ON}\mathbf{A}_{\mathbf{q}_h}(\mathbf{q}_h - \mathbf{q}^r_h)$	$\mathbf{A}_{\mathbf{q}^r_{hT+\tau}}\dot{\mathbf{q}}^r_{hT+\tau}$
2)	$-k_{OFF}\mathbf{A}_{\mathbf{q}_h}(\mathbf{q}_h - \mathbf{q}^r_h)$	$\mathbf{A}_{\mathbf{q}^r_{hT+\tau}}\dot{\mathbf{q}}^r_{hT+\tau}$
3)	$-k_{ON}\mathbf{A}_{\mathbf{q}_h}(\mathbf{q}_h - \mathbf{q}^r_h)$	$\mathbf{A}_{\mathbf{q}_{hT}}\dot{\mathbf{q}}^r_{hT+\tau}$
4)	$-k_{ON}\mathbf{A}_{\mathbf{q}_h}(\mathbf{q}_h - \mathbf{q}^r_h)$	$\mathbf{0}$

Table 5.1. Description of the four strategies used in simulation.

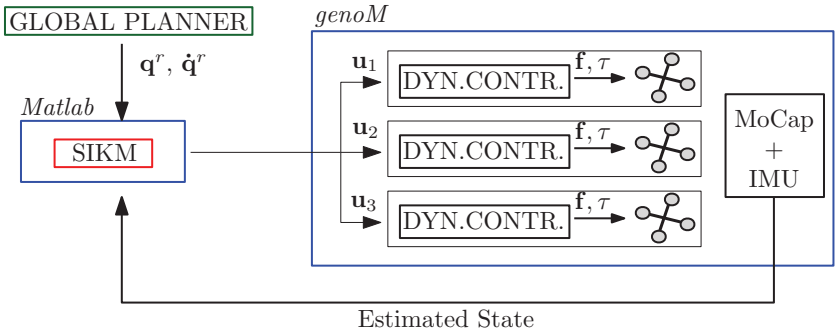


Figure 5.6. Architecture used to perform simulations:a global planner generates the desired trajectory $\mathbf{q}^r, \dot{\mathbf{q}}^r$, sends it to the local planner which generates the desired robots velocities. The blue rectangle on the right represents a realistic environment where the robotic system is simulated.

case, we show practically in the next plots that a bounded tracking error can be achieved. In the simulations we also plot the results of the sampled Jacobian technique, introduced in the first part of Section 5.3.3 (strategy 3), where the gain is computed on-line following the same reasoning of point-stabilization. Finally, we present our solution to the problem: in particular, we adopt the second technique of Section 5.3.3. In one case we compute the optimal gain off-line (strategy 2), following the procedure of Section 5.4 where stability guarantees were given, then we exploit the same technique but k is computed on-line (5.12) (strategy 1) to check if this allows to reach a faster convergence with respect to the off-line computation.

We simulate the four techniques on the Fly-Crane: it consists of three Quadrotor UAVs connected by six cables to a platform, depicted in Figure 3.1a. The dynamical model of the system has been developed in Gazebo and the communication part among the system, the planner and the sensing part was implemented by using a middle-ware called Pocolibs and the software framework genoM Koenig and Howard (2004). These tools allow to represent in a very detailed manner the real system that is described in Sanalitro et al. (2020). The architecture that we used is represented in Figure 5.6: a global planner is assumed to generate off-line the desired trajectory $(\mathbf{q}^r, \dot{\mathbf{q}}^r)$ and this information is available to the SIKM controller, implemented in Matlab-Simulink. This generates the desired robots velocities $\mathbf{u}_i$ which are sent to the robots every T seconds. Then, the low-level-dynamical controller of each robot converts these velocities into thrust and torque $(\mathbf{f}, \tau)$ for the quad-rotors. Then, an unscented Kalman filter, running at 1 [kHz], fuses the Motion Capture (MoCap) System measurements (at 120 [Hz]) with the IMU measurements (at 1 [kHz]). The estimated system state is then sent to the SIKM controller.

Of course, when implementing the architecture on a real system, the blue rectangle on the right in Figure 5.6 is no more simulated but real sensors and robots are exploited. Moreover, the desired velocities would be sent via wireless.

Next, we show two different results: firstly the four tracking strategies described above are compared when $T = 1.5$ [s]. In this first part you will see the benefits of the feed-forward term with respect to to the point-stabilization technique, which is slower in tracking the desired trajectory. In the second part we compare the four strategies using different sampling times and we plot the mean of the tracking error norms obtained in

	1	2	3	4
$T = 0.5[s]$	k_{on}	$k_{off} = 2$	k_{on}	k_{on}
$T = 0.75[s]$	k_{on}	$k_{off} = 1.28$	k_{on}	k_{on}
$T = 1.5[s]$	k_{on}	$k_{off} = 0.67$	k_{on}	k_{on}
$T = 4[s]$	k_{on}	$\times$	k_{on}	$\times$

Table 5.2. Couples (k,T) of gains and sampling times used to compare the four strategies 1), 2), 3), 4). The norm of the tracking error is depicted in Figure 5.9b.

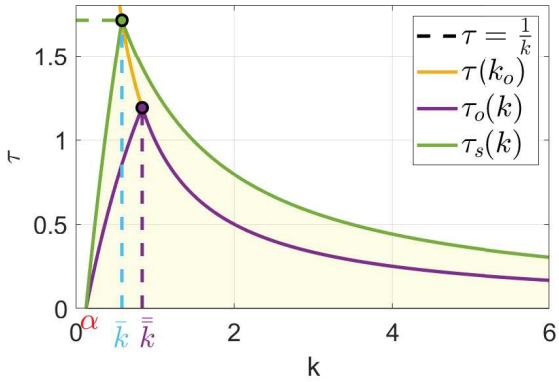


Figure 5.7. Representation of the stability region, the optimal gain k_o and sampling time τ_o, obtained by estimating the parameters $\mu, \alpha, \gamma_1, \gamma_2$ along the trajectory depicted in Figure 5.8 and for different couples (k,T).

the simulations. Notice that, in order to apply strategy 2, we estimated the parameters $[\mu, \alpha, \gamma_1, \gamma_2] = [0.02, 0.13, 0.1, 0.2]$ along the desired trajectory depicted in Figure 5.8 for different couples of (k,T) as described in Section 5.4.1, obtaining the stability region depicted in Figure 5.7. Then, for each value of T, we chose the gain k which gave the maximum convergence rate while keeping the system stable. To do that we only needed to chose the gain on the curve $\tau(k_o)$ in the picture which corresponds to the sampling time T. Table 5.2 indicates which simulations have been performed: in three cases we could test all the strategies; then, for larger values of T (last row) strategy 2 is no more feasible because $T > \tau_{CR}$, hence stability cannot be guaranteed. Finally for even bigger T (last row), strategy 4 is no more interesting because the error norm increases too much.

Let us consider now the first result: in Figure 5.8 the variables $\mathbf{q}(t)$ are depicted: the desired trajectory was generated in order to stress all the components of $\mathbf{q}$, except for the y-translation since the system is symmetric with respect to $y = x$, hence we forced only translations along the x-axis. We assume that the sampling time is $T = 1.5$ [s]. In this picture you can notice that strategy 4 has clearly a delay with respect to the feed-forward techniques, as emphasized in the zoom plot of the component z_L: this is due to the fact that the desired velocity $\mathbf{u}$ is updated only when a new measurement arrives. As far as the feed-forward techniques are concerned, strategies 1 and 2 gives slightly better performance with respect to the other two: this can be intuitive by observing the zoomed plot (z_L), where strategy 3 has an overshoot, but is even more evident observing Figure 5.9a, where the Euclidean norm of the tracking error is depicted. Anyway, this comparison is related to a single value of T: a more interesting and general result is depicted in Figure 5.9b where the strategies are compared for different values of the sampling time, as resumed in Table 5.2. In particular, for each strategy, we computed the

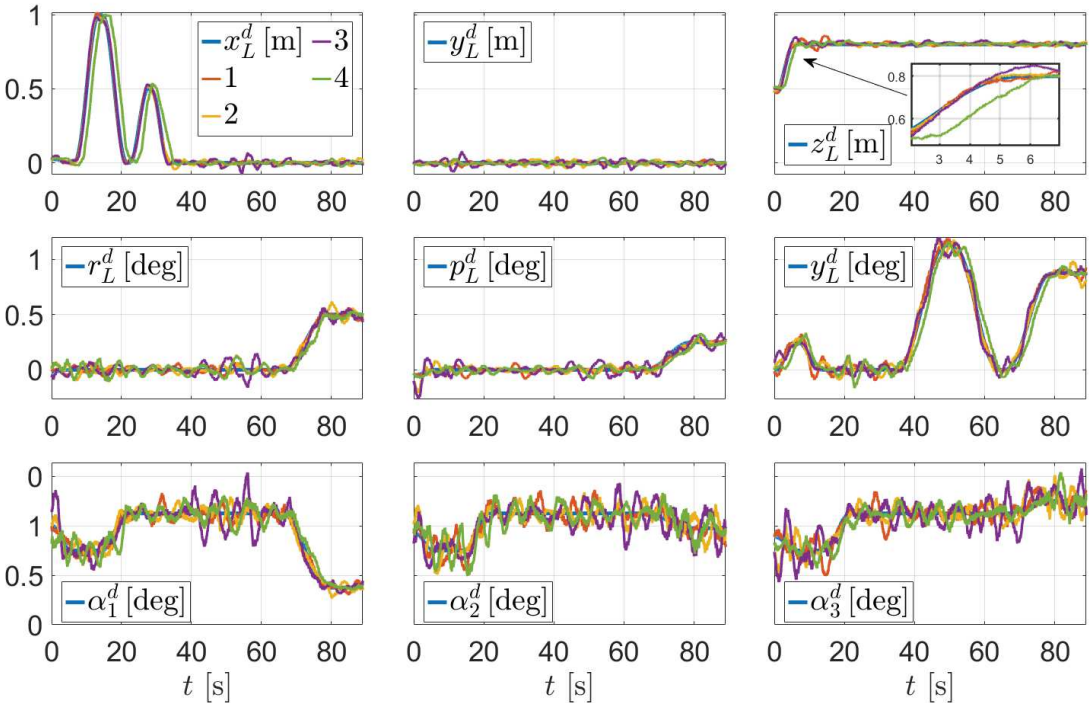
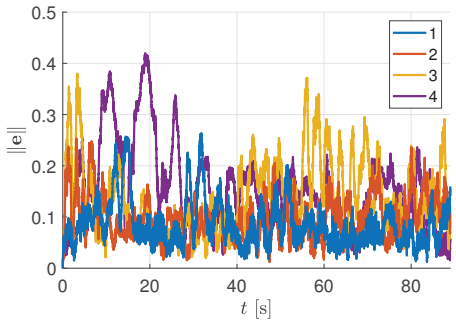
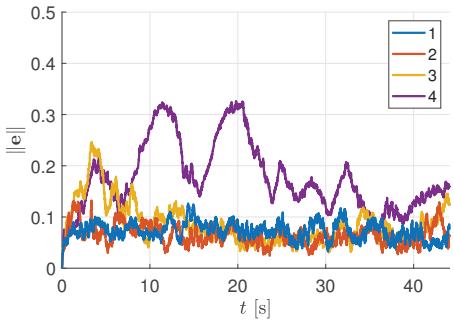


Figure 5.8. Comparison of the variables $\mathbf{q}(t)$ in three different simulations, where the tracking strategies 1), 2), 4) described in Tab.5.1 are depicted. The used sampling time is $T = 1.5$ [s]. The first two rows represent respectively the position x_l, y_L, z_L and orientation r_L, p_L, y_L (roll, pitch and yaw) of the load. On the last row the angles $\alpha_i := q_i$, $i = 1,2,3$ between the cables and the load are depicted.

mean of the tracking error norm of different simulations, characterized by different values of T. For instance let us call $\|\mathbf{e}^i_{T_j}\|$ the norm of vector $\mathbf{e}^i_{T_j}$ that is the tracking error obtained by applying the i-th strategy ($i = 1,\dots 4$) with $T = T_j$, $j = 1,\dots,N_j$. Now, each line of the plot is related to one of the four strategies and is obtained by computing the mean $\frac{1}{N_j}\sum_{j=1}^{N_j}\|\mathbf{e}^i_{T_j}\|$. This plot allows to appreciate the benefit of the feed-forward term in (5.5), indeed strategy 4 is characterized by a very larger tracking error; moreover, it is evident that strategy 3, besides the fact that it does not give any stability guarantees, it is slower in achieving zero-tracking error, indeed the Jacobian of the feed-forward term is sampled, hence updated every T seconds. Finally, strategies 1 and 2 are comparable, indeed we proved that 1 was faster only in the point-stabilization scenario. Anyway, from the plot we can see a slight better performance of 1, where the gain is updated on-line, with respect to strategy 1, especially during the first seconds. Finally, we show in Figure 5.9 the behavior of the system state when increasing the noise variance of the sensor that collects data for the MoCap system: σ_2 is the variance used for the previous simulations, while $\sigma_1 = 3\sigma_2$. You can notice that, as expected, the performance become worse when increasing the noise variance and the oscillations around the reference trajectory become bigger. This result shows that our strategy can handle uncertain measurements of the state $\mathbf{q}$ while guaranteeing stability of the system.



(a) *Comparison of the error tracking norm* $\mathbf{e}(t) = \mathbf{q}(t) - \mathbf{q}^r(t)$ *when using the four tracking strategies described in Tab.5.1 are depicted. The used sampling time is* $T = 1.5$ *[s]*



(b) *Each line in the plot is obtained by computing the mean of the tracking error norm for the same strategy applied with different sampling times* T*.*

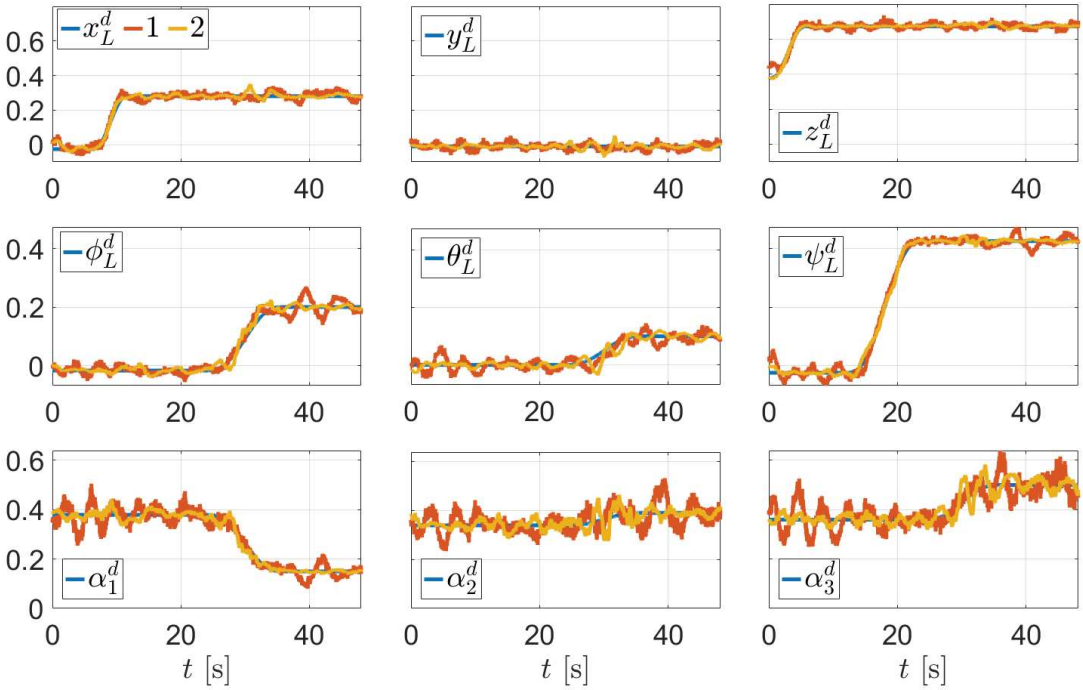


Figure 5.9. Comparison of the variables $\mathbf{q}(t)$ for two different values $\sigma_1 = 0.01$ and $\sigma_2 = 0.003$ of noise variance of the sensor that collects measurements fro the MoCap system: the tracking strategy 2) is exploited with $(k, T) = (1.28, 0.75)$. The used sampling time is $T = 0.75$ [s]. The first two rows represent respectively the position x_l, y_L, z_L and orientation ϕ_L, θ_L, ψ_L (roll, pitch and yaw) of the load. On the last row the angles $\alpha_i := q_i$, $i = 1, 2, 3$ between the cables and the load are depicted.

6

Conclusions and future works

The global goal of this thesis was to advance in the control and motion planning for aerial robots interacting with the environment. In particular, we studied different types of tethered aerial vehicles and derived novel control laws which make the systems perform demanding tasks in challenging scenarios.

6.1 Summary

In Chapter 2 we consider an unmanned aerial vehicle tethered to the ground through a cable. For this system we design a real-time control strategy to drive it from an initial to a final configuration. In particular, our goal is to design a NMPC based solution, able to handle external disturbances. Indeed, until now, solutions based on purely NMPC are considered not trustable for real applications, since they are model-based, hence do not consider a number of non idealities. As a consequence, these solutions are delegated to research. In this work we wanted to prove that it is possible to improve the controller performance when a low level controller, working at high frequency, is added into the loop. Indeed, it is not ensured the NMPC can work at high frequencies (kHz), for example when the model is particularly complicated and the optimization problem to be solved at each iteration is computationally demanding. As a consequence, if external disturbances affect the system at these frequencies, they cannot be handled by the controller. Thus, we propose a two-layers control strategy where the NMPC is used at a lower rate to generate feasible reference trajectories for the low level local controller, which runs at a higher frequency. Thanks to this approach the low-level controller design results to be simple and the NMPC can handle complex constraints and control objectives. Thus, the trajectory generated by NMPC adapts on-line to possible disturbances or modeling errors. A fast implementation of NMPC is used to achieve real time performance in presence of nonlinear constraints. The effectiveness of the proposed control strategy is proved by simulations where a challenging maneuver is performed. In particular, the effect of the wind on the system is analysed, showing that the multi-layer strategy guarantees a better rejection to un-modeled disturbances with respect to the use of purely NMPC.

In Chapter 3, we defined a class of multi-agent systems whose kinematic model is characterized by a square and invertible Jacobian matrix. The considered class includes different types of systems, from ground to aerial one, hence the result we obtain is pretty general. In particular, we are interested to the case where communication is implemented via wireless. This subject is usually not considered in robotics, where communication is assumed to be continuous or is not employed. We face the problem of steering a multi-agent system to a desired configuration by means of a feedback-based planner that

exploits the sparse structure of the system and provides the desired vehicles velocities to be tracked by the system. As we discussed in the Chapter, a decentralized approach is commonly adopted in robotics for several reasons, like the fact that it guarantees major robustness and flexibility with respect to a centralized approach. The choice of exploiting communication is useful in scenarios where the exact position and orientation of an object of interest (load to be transported, object to be manipulated, robot to be escorted and so on) cannot be retrieved exactly from the knowledge of the system structure. For these reasons we design and compare two different strategies to make the system accomplish the task: the offline strategy converges exponentially fast and exploits a static control gain, while the second method is even faster, but computationally demanding since the gain must be computed online. Furthermore, if the second technique can be implemented only in a centralized fashion, the first one can be fulfilled also in a distributed way. Anyway, both the techniques do not require any communication among the robots. We validate the proposed strategies with simulations on the kinematic model of the system; then, we also report the results obtained on a real system. In detail, instead of assigning a constant reference configuration, we considered the more realistic scenario where a path must be followed. The experiment confirmed the theoretical results.

In Chapter 4 we faced a similar problem as in the previous chapter, but we considered a more general scenario, where the Jacobian matrix of the kinematic model is not square. This aspect corresponds to have an over-actuated or under-actuated system. In particular, we focused on the first case: notice that the dimension of the input to the system is greater than the degrees of freedom, hence one could assign unfeasible inputs which cannot be handled by the system. To solve this problem, we adapt the architecture introduced in Chapter 3 where a global planner is assumed to generate off-line the reference configurations and a local planner generates on-line the robots velocities in order to make the configuration error converge to zero as fast as possible. We still propose two strategies to generate the gain of the local planner, with different computational requirements and performance. These results are proved both for Euclidean and infinite norm. Anyway, the most interesting contribution is the solution adopted to handle the case where the reference velocities for the robots are not feasible for the system. In this scenario, we propose a technique to project these values on a feasible space. We also show, through a multi-scale time approach, that this choice is consistent with what happens when applying such unfeasible inputs to a dynamical system. This allows to validate the theoretical results obtained in the Chapter. A further contribution is the simulation with the dynamical model of a non-square system under far-from-ideal conditions: we simulate a wireless network to test our strategies when the packet loss probability is greater than zero.

In Chapter 5 we propose a Sampled communication-aware Inverse-Kinematics controller for Multi-robot systems. The goal is to make a multi-robot system track a desired trajectory: in detail, the controller we design is assumed to know the reference configurations while it receives the measurements from the system at sampled time instants. Based on this knowledge, it computes the desired robots velocities to complete the task. We provide two strategies to compute the controller gain. In one case the gain is computed off-line and we provide stability and convergence guarantees, while in the second case the gain is updated on-line, allowing a faster convergence rate, but only practical guarantees

CH. 2	ONLINE TRAJECTORY GENERATOR NMPC	LOCAL CONTROLLER Feedback Linearization
CH. 3-4	OFFLINE PATH GENERATOR	LOCAL PLANNER Inverse Kinematics
CH. 5	OFFLINE TRAJECTORY GENERATOR	LOCAL CONTROLLER Inverse Kinematics

Figure 6.1. Presence of a multi-layer structure across the thesis.

can be shown. Moreover, we compare these techniques with the point-stabilization strategy proposed in Chapter 3: it turns out that it does not guarantee the tracking error to converge to zero, while the new strategy, where a feed-forward term is added to the control law, allows to track a reference trajectory with a smaller tracking error. We validate our results by implementing the strategies to cooperatively manipulate a cable-suspended load.

6.2 Discussion

Notice that the whole work deals with tethered aerial solutions (even if we showed that our strategies can be adapted to different types of systems) for which we propose a two-layer architecture. What we learned is that this choice can be very helpful to deal with model uncertaintes and external disturbances. This observation comes mainly from Chapter 2 where we simulated the wind disturbing the system and handled un-modeled parts of the system. In addition, in Chapters 3-5 we found different solutions to drive a multi-agent system from a configuration to an other one or to track a trajectory. In these cases we could exploit the kinematic structure of the system to generate reference velocities for the agents. We assumed the presence of an off-line path/trajectory generator, that avoids singular configurations and eventual obstacles in the path. In these chapters, our strategies allowed to solve the problem of driving the multi-agent system even in presence of sampled communication, i.e. when the state measurements are available at sampling instants. The multi-layer architecture allows to intermediate between the high-level planner and the system, by generating on-line the reference velocities for the agents by guaranteeing stability and convergence to the desired configuration. In Figure 6.1 we reported a resume of the proposed techniques, highlighting the two levels in each problem.

6.3 Future works

As far as the solution proposed in Chapter 2 is concerned, there is space for improvements. For instance, we considered a simplified 2D model of the system, assuming that it can move only on a plane. An intuitive extension is to consider the 3D model to both formulate the NMPC problem and design the low-level controller. In Tognon et al. (2016a) the authors study this scenario where the 3D model is considered. Anyway, in that case the optimal trajectory to be tracked by the system is generated off-line. As a consequence, any disturbance or in general any un-predicted event cannot be handled by the controlled system. In conclusion, the next step would be a merge between that work and the one presented in this thesis, where the optimal trajectory is generated on-line and the 3D model is considered. An other important aspect that would give more relevance to our work is to test our multi-layer strategy on a real system. An other interesting future work is the theoretical analysis of the system stability during the maneuver. Dealing with Chapter 3, the main aspect to develop is the design of the global planner; indeed, the sequence of points to be reached by the system is generated off-line and we assume that singular configurations are avoided in this process. The natural development is the design of the planner, that is the study of a cost function with the related constraints in order to generate effectively the sequence of configurations. In addition, we could implement a NMPC based solution in order to generate on-line the points: this approach is helpful to handle external disturbances by generating a new sequence if the system state is different from the desired one; moreover, additional parameters as the inter-sample interval can be taken into account. Further developments of this work are already included in the following chapter of the thesis, where we study systems characterized by non-square Jacobian matrix in Chapter 4 and then we focus on trajectory tracking instead of point-stabilization control in Chapter 5. Notice that a fault-tolerance analysis would be an interesting development: if for example one cable breaks the system could not be able to sustain the load anymore: in this case the system would not be square, but we have seen that there exist rectangular configurations which can transport the load, see Chapter 4. It would be interesting to analyse for which of these configurations the system would be still able to perform the task. An other important aspect is collision avoidance: indeed, during the maneuver a robot may collide against an other robot. This can be avoided by adding a constraint in the off-line trajectory generator. Anyway, this could be not enough if the collision happens between two sampling times; in this case the use of cameras or other sensors could help to avoid collisions.

In Chapter 5 we propose a Sampled communication-aware Inverse-Kinematics controller for Multi-robot systems. In this case two main aspects could be developed: the trajectory generation could be implemented by exploiting NMPC based solutions, as we already explained in the previous lines; then, experiments with a real system would be a natural step forward[1]. In this thesis we analysed scenarios where communication was simplified as a sampled transmission. In particular we provided theoretical analysis only in this case. Of course a more realistic communication would be an interesting aspect to be

[1] Actually, the experiments were scheduled, but then canceled because of the restrictions related to COVID-19

considered, taking into consideration possible delays, packet losses and synchronization. Actually, in Chapter 4 and A we provided the results obtained by simulating a non-square system in a realistic scenario, exploiting a tool called *Truetime*: it allowed to simulate a real wireless network, with all the related characteristics mentioned above. Of course, it would be interesting, as future work, to provide theoretical analysis of this more complete scenario.

Related to this aspect, an other aspect could be taken into consideration: event-triggered and self-triggered control systems Heemels, Johansson, and Tabuada (2012). They consist of two elements: a feedback controller computing the control input and a triggering mechanism determining when the control input has to be updated again. In the first case a triggering condition is based on current measurements and is continuously monitored: when it is violated, an event is triggered. In the latter, the next update time is pre-computed at a control update time and is based on predictions using previously received data and knowledge on the dynamical model of the plant. We could apply these strategies for instance to avoid periodic communication and choosing to exchange information only under certain conditions.

A

Supplementary material

A.1 Additional simulation with a non-square system

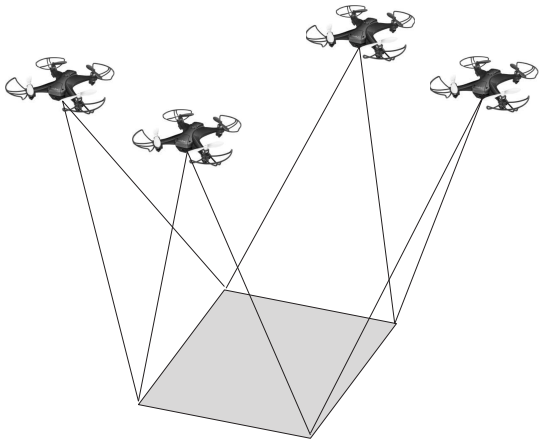


Figure A.1. Example of a rectangular multi-agent system where four aerial robots transport a payload.

In this section we show the behavior of the system depicted in Figure A.1 when the strategies introduced in Chapter 4 are exploited. Notice that it is a non-square system, hence the Jacobian matrix of the kinematic model is rectangular, indeed it has $n = 3N$ rows, with $N = 4$ the number of robots, and $m = 10$ columns (which correspond to four angles of the cables and the six degrees of freedom of the payload). In order to simulate the system in a far-from ideal environment, we exploit a Matlab/Simulink-based toolbox called *Truetime*. We already detailed its characteristics in Section 4.6.1.

In this simulation we consider the packet loss probability equal to 20%. Figures A.2 and A.3 represent the behavior of the Lagrangian variables $\mathbf{q}$ starting from an initial configuration $\mathbf{q}^0 = [80^\circ\ 80^\circ\ 80^\circ\ 80^\circ\ 0[m]\ 0[m]\ 0[m]\ 0^\circ\ 0^\circ\ 0^\circ]^\top$ to the final one $\mathbf{q}^r = [60^\circ\ 60^\circ\ 60^\circ\ 60^\circ\ 0[m]\ 0[m]\ 0[m]\ 0^\circ\ 0^\circ\ 0^\circ]^\top$. In the on-line scenario the load measures the whole vector $\mathbf{q}$, then it computes the desired velocities $\mathbf{u}$ and sends them to the robots every $1.5[s]$ (see last plot of Figure A.4). If they receive the message, then they try to

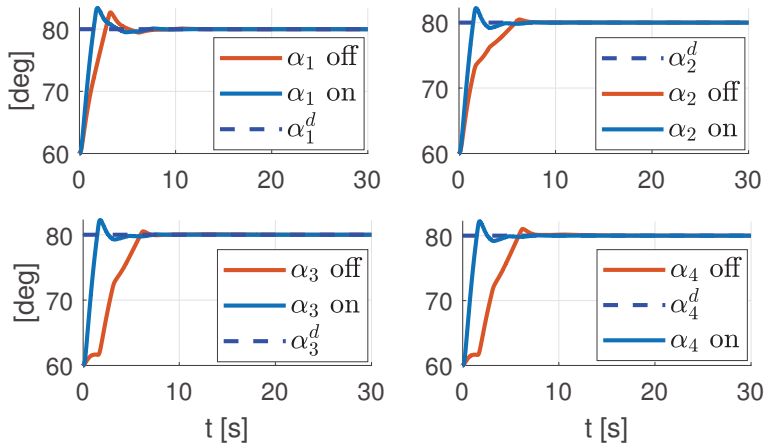


Figure A.2. Behavior of the variables $\mathbf{q}_i = \alpha_i$ which are the angles between the cables and the load. The off-line and on-line strategies are compared when the wireless packet loss probability is set to 20% and the transmission period to $T = 1.5\,[s]$.

track the desired velocity, otherwise the desired velocity is set to zero until next message arrives. Moreover a faster periodic task establishes if a message was lost by checking every $0.2[s]$ how much time has passed since the last message (see plots R_1, R_2, R_3, R_4 in Figure A.4). In the off-line case, the load pose is sent every $1.5[s]$ (see last plot of Figure A.5) and the reference velocities $\mathbf{u}$ to be tracked are computed by the robots. If a packet loss occurs, then the latest value of $\mathbf{u}$ is kept as reference. This is possible thanks to the smaller value of the planner gain, whereas the on-line strategy aims at a faster convergence rate, by exploiting a bigger value of the gain: as a consequence it would be not safe to keep the last reference velocity constant until the new reference arrives (see the comparison between the gains in Figure A.6).

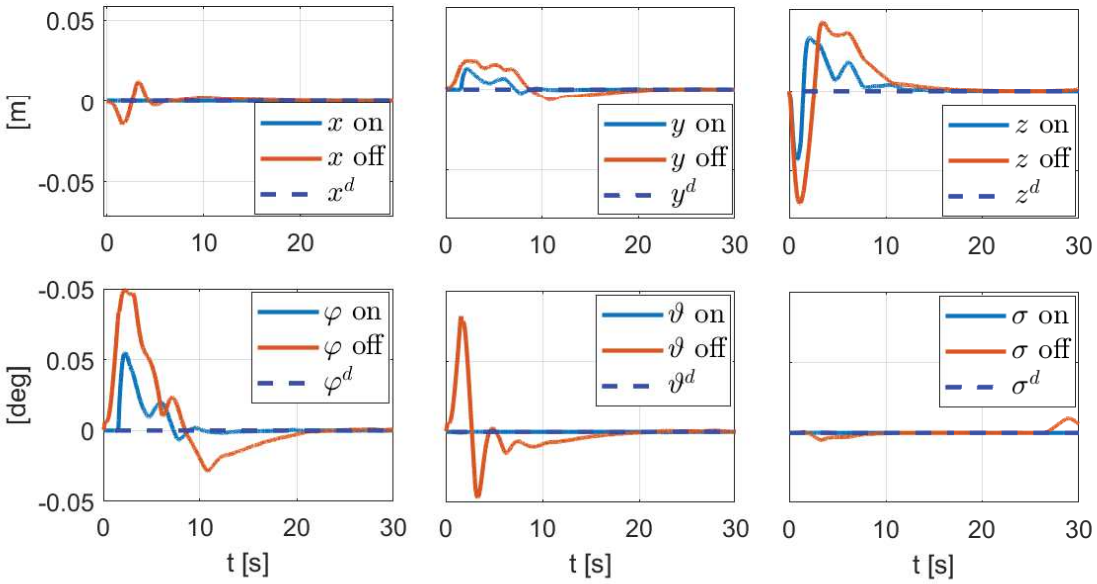


Figure A.3. Behavior of the load position and orientation $\mathbf{q}_V$: x, y, z are the coordinates on the three axes and ϕ, θ, σ the Euler angles. The off-line and on-line strategies are compared when the wireless packet loss probability is set to 20% and the transmission period to $T = 1.5\,[s]$.

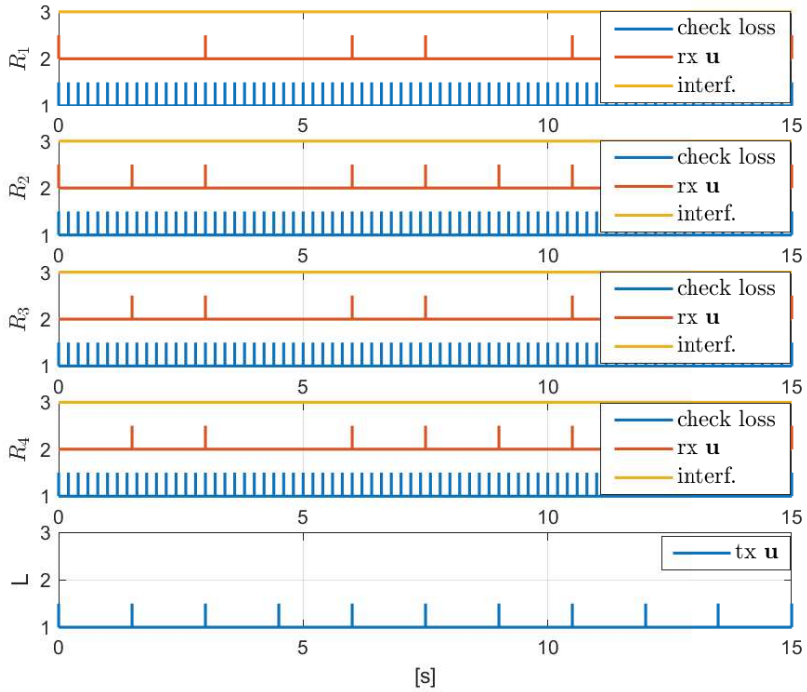


Figure A.4. On-line scenario: close-up of schedules showing the allocation of common resources: robots (top) and platform node (bottom). A high signal means sending or executing and a low signal means idle. The platform sends the reference velocity vector **u** to the robots every $T = 1.5[s]$. Every $0.2[s]$ each robot checks if a packet loss occurred (that is if last message arrived more than $1.5[s]$ before). In this case the desired velocity is set to zero until next packet arrives.

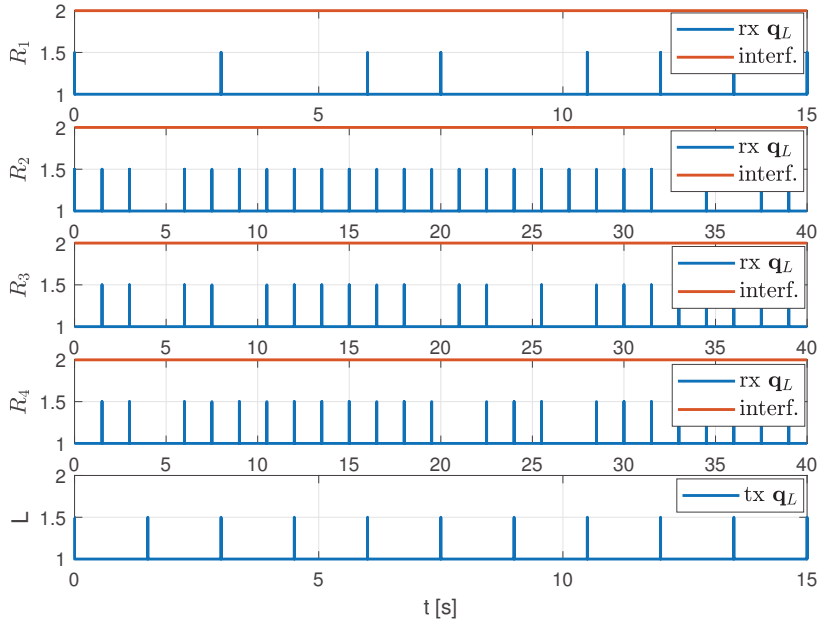


Figure A.5. Off-line scenario: close-up of schedules showing the allocation of common resources: robots (top) and platform node (bottom). A high signal means sending or executing and a low signal means idle. The platform sends its pose to the robots every $T = 1.5[s]$. The robots receive the packets unless a loss occurred. When a packet arrives, then u is computed.

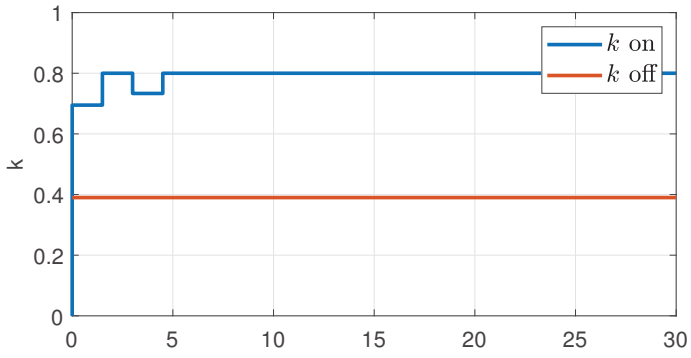


Figure A.6. Comparison between the off-line and on-line gains: note that the faster convergence rate achieved by the on-line strategy is possible thanks to a bigger value of the gain.

B

Appendix

B.1 Derivation of the Fly-crane model

B.1.1 Kinematic model

In this section the system, reported in Fig.B.1, is introduced and described. We refer to it as Fly-crane and it consists of a load (referred as V in the following), that is a flat platform, that can move in the 3D space thanks to three flying robots which carry it. Each robot is linked, through two cables, to two points of the platform. In Fig.B.2, we represent the reference frames which are define to derive the robots and load positions in the world reference frame. Some assumptions are taken with respect to the system variables:

- The cables are tight, hence their length is fixed, equal to l_i
- The dimensions of the load are given
- the cables are considered mass-less
- We assume that the robots center of mass corresponds to the point where the cable is attached

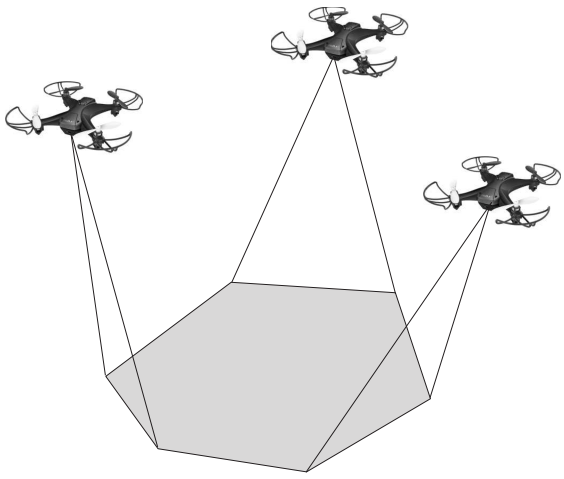


Figure B.1. Representation of the fly-crane.

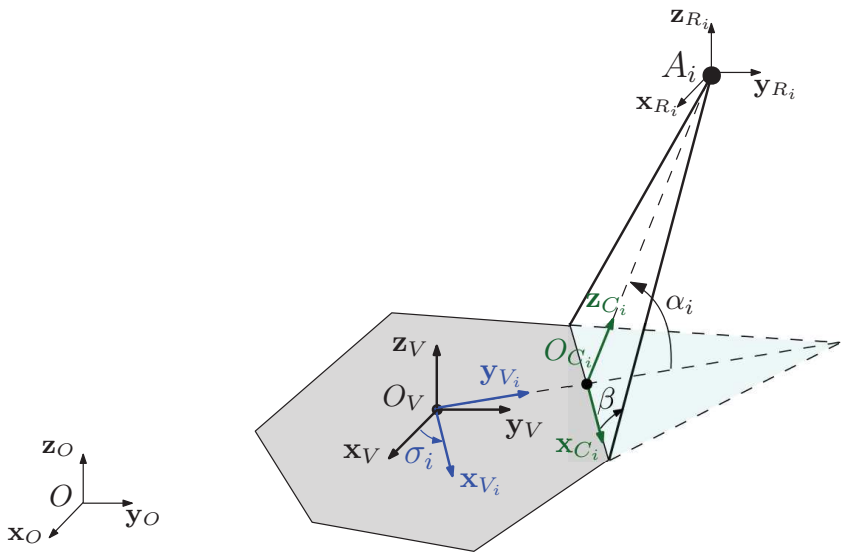


Figure B.2. Definition of the reference frames.

Let $\mathcal{F}_O = \{O, \mathbf{x}_O, \mathbf{y}_O, \mathbf{z}_O\}$ be the fixed reference frame and $\mathcal{F}_V = \{O_V, \mathbf{x}_V, \mathbf{y}_V, \mathbf{z}_V\}$ the moving one, centered on the Center of Mass (CoM) of the load. From $\mathcal{F}_V$, it is immediate to define the reference frames $\mathcal{F}_{V_i} = \{O_V, \mathbf{x}_{V_i}, \mathbf{y}_{V_i}, \mathbf{z}_{V_i}\}$, $i = 1,2,3$ which are centered on O_V and rotated about the axis $\mathbf{z}_V$ of σ_i: each $\mathbf{y}_{V_i}$ points to the corresponding O_{C_i}, which divides each load edge in two equal parts. Each robot can rotate about the edge to which it is attached: to this aim we define the reference frames $\mathcal{F}_{C_i} = \{O_{C_i}, \mathbf{x}_{C_i}, \mathbf{y}_{C_i}, \mathbf{z}_{C_i}\}$, $i = 1,2,3$. Thus, the position of the i-th robot is described by a rotation about $\mathbf{x}_{C_i}$ of α_i with respect to $\mathcal{F}_{O_{C_i}}$. Moreover, we call β the constant angle between each cable and the axis $\mathbf{x}_i$.

In the following, we define the unitary vectors $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3 \in \mathbb{R}^3$ and we use the notation $\mathbf{p}_H^K$ to indicate the position of a point H with respect to a reference frame $\mathcal{F}_K = \{O_K, \mathbf{x}_K, \mathbf{y}_K, \mathbf{z}_K\}$. When the reference frame is not indicated, then we mean that it is the fixed reference frame $\mathcal{F}_O$. Moreover, we define l_{C_i} the distance between O_{C_i} and O_V. Thus we can derive the expressions for the following position vectors

- $\mathbf{p}_{A_i}^{C_i} = l_i \, sin(\beta) \, \mathbf{e}_3$ is the position of the point A_i with respect to $\mathcal{F}_{C_i}$;
- $\mathbf{p}_{C_i}^{V_i} = l_{C_i} \mathbf{e}_2$ is the position of the point O_{C_i} with respect to $\mathcal{F}_{V_i}$;
- $\mathbf{p}_V^O$ is the position of the load CoM with respect to $\mathcal{F}_O$.

The generalized coordinates of the system are chosen as $\mathbf{q} = \left[\mathbf{q}_\alpha{}^T; \mathbf{q}_V^T\right] \in \mathbb{R}^9$ where $\mathbf{q}_\alpha = [\alpha_1 \ \alpha_2 \ \alpha_3]^\top \in \mathbb{R}^3$ and $\mathbf{q}_V = [x_V \ y_V \ z_V \ \phi_V \ \theta_V \ \psi_V]^\top \in \mathbb{R}^6$, which are the position and orientation (expressed with Euler angles roll, pitch and yaw) of the load. The rotation matrix that allows to transform vectors from $\mathcal{F}_V$ to $\mathcal{F}_O$ and vice-versa is

$$R_V^O = R_z(\phi_V) R_y(\theta_V) R_x(\psi_V) \tag{B.1}$$

Other useful rotation matrices are described in the following:

- $R_{V_i}^V$, $i = 1,2,3$ represents the orientation of $\mathcal{F}_{V_i}$ with respect to $\mathcal{F}_V$, obtained by rotating $\mathcal{F}_V$ about $\mathbf{z}_V$ of σ_i;

- $R_{C_i}^{V_i}$ represents the orientation of $\mathcal{F}_{C_i}$ with respect to $\mathcal{F}_{V_i}$, obtained by rotating $\mathcal{F}_{V_i}$ about $\mathbf{x}_{V_i}$ of $-(\frac{\pi}{2}-\alpha_i)$ (the minus is due to the clock-wise rotation and $\frac{\pi}{2}$ appears because the rotation angle is actually the complementary angle of α_i);
- $R_{C_i}^{O} = R_V^O\, R_{V_i}^V\, R_{C_i}^{V_i}$ is the rotation matrix from $\mathcal{F}_{C_i}$ to $\mathcal{F}_O$.

At this point, the position of the i-th robot with respect to $\mathcal{F}_O$ is:

$$\mathbf{p}_{A_i}^O = \mathbf{p}_V^O + R_V^O\, R_{V_i}^V\, (\mathbf{p}_{C_i}^{V_i} + R_{C_i}^{V_i}\, \mathbf{p}_{A_i}^{C_i});$$

Hence

$$\mathbf{p} = \begin{bmatrix} \mathbf{p}_{A_1}^O \\ \mathbf{p}_{A_2}^O \\ \mathbf{p}_{A_3}^O \end{bmatrix} = \mathbf{h}(\mathbf{q}) = \begin{bmatrix} \mathbf{h}^{(1)}(\alpha_1, \mathbf{q}_V) \\ \mathbf{h}^{(2)}(\alpha_2, \mathbf{q}_V) \\ \mathbf{h}^{(3)}(\alpha_3, \mathbf{q}_V) \end{bmatrix} \tag{B.2}$$

where $\mathbf{p} = [\mathbf{p}_{A_1}^\top\ \mathbf{p}_{A_2}^\top\ \mathbf{p}_{A_3}^\top]^\top \in \mathbb{R}^{9\times 1}$ with $\mathbf{p}_{A_i} \in \mathbb{R}^{3\times 1}$, $\mathbf{q} = [\alpha_1\ \alpha_2\ \alpha_3\ \mathbf{q}_V^\top]^\top \in \mathbb{R}^{9\times 1}$, with $\alpha_i \in \mathbb{R}$ and $\mathbf{q}_V \in \mathbb{R}^{6\times 1}$.

The differential kinematics of the system is:

$$\dot{\mathbf{p}} = \frac{\partial \mathbf{h}(\mathbf{q})}{\partial \mathbf{q}}\, \dot{\mathbf{q}} = \mathbf{A}_\mathbf{q}\, \dot{\mathbf{q}},$$

where $\mathbf{A}_\mathbf{q}$ is the Jacobian matrix.

B.1.2 Dynamical model

In this section we show how to derive the dynamical model of the Fly-crane. We adopt the Lagrangian-Euler method, hence we need to compute the kinetic K and potential U energies of the system. Recall that we assume mass-less cables and that they are linked to the robots with a passive joint at their CoM. In the following we use the subscript R_i to refer to the robots reference frames and positions, hence $O_{R_i} := O_{A_i}$ and $\mathbf{p}_{R_i} = \mathbf{p}_{A_i}$.

The kinetic energy of the load is

$$K_V = \frac{1}{2}\, m_V\, \mathbf{p}_V^\top\, \mathbf{p}_V + \frac{1}{2} \boldsymbol{\omega}_V^\top\, \mathbf{I}_V\, \boldsymbol{\omega}_V$$

where $\mathbf{I}_V \in \mathbb{R}^{3\times 3}$ is the positive definite inertia matrix of the load, expressed in $\mathcal{F}_V$ and relative to O_V, m_V is the load mass and $\boldsymbol{\omega}_V = [\dot{\phi}_V\ \dot{\theta}_V\ \dot{\psi}_V]^\top \in \mathbb{R}^3$ the angular velocity of $\mathcal{F}_V$ with respect to $\mathcal{F}_O$, expressed in $\mathcal{F}_V$.

The potential energy of the load is

$$U_V = m_V\, g\, \mathbf{e}_3^\top\, \mathbf{p}_V$$

where g is the gravitational constant.

The reference frame centered on robot i is defined as $\mathcal{F}_{R_i} = \{O_{R_i}, \mathbf{x}_{R_i}, \mathbf{y}_{R_i}, \mathbf{z}_{R_i}\}$. The orientation of the robot in the fixed reference frame is expressed by the roll, pitch and yaw angles $\mathbf{q}_{R_i} = [\phi_{R_i}, \theta_{R_i}, \psi_{R_i}]^\top$. Thus $R_{R_i}^O = R_z(\phi_{R_i}) R_y(\theta_{R_i}) R_x(\psi_{R_i})$ represents the rotation from $\mathcal{F}_{R_i}$ to $\mathcal{F}_O$.

Thus, the kinetic energy of the robots is

$$K_{R_i} = \frac{1}{2} m_R \dot{\mathbf{p}}_{R_i}^\top \dot{\mathbf{p}}_{R_i} + \frac{1}{2} \boldsymbol{\omega}_{R_i}^\top \mathbf{I}_R \boldsymbol{\omega}_{R_i},\ i = 1,2,3$$

where $\mathbf{I}_R \in \mathbb{R}^{3\times 3}$ is the positive definite inertia matrix of each robot, expressed in $\mathcal{F}_{R_i}$ and relative to O_{R_i}, while m_R is the mass and $\boldsymbol{\omega}_{R_i} = [\dot{\phi}_{R_i}\ \dot{\theta}_{R_i}\ \dot{\psi}_{R_i}]^\top \in \mathbb{R}^3$ is the angular velocity of $\mathcal{F}_{R_i}$ with respect to $\mathcal{F}_O$, expressed in $\mathcal{F}_{R_i}$.

The potential energy of each robot is

$$U_{R_i} = m_R g \mathbf{e}_3^\top \mathbf{p}_{R_i},\ i = 1,2,3$$

The total kinetic energy of the system is then

$$K(\mathbf{q}, \dot{\mathbf{q}}) = K_V + K_{R_1} + K_{R_2} + K_{R_3}$$

and the potential one

$$U(\mathbf{q}) = U_V + U_{R_1} + U_{R_2} + U_{R_3}$$

Now, let us compute the generalized forces and torques. The motion of a quad-rotor is controlled by the coordinated action of four control inputs: $f_{R_i} \in \mathbb{R}$ is the intensity of the total thrust applied in O_{R_i} such that $\mathbf{f}_{R_i} = f_{R_i}\mathbf{z}_{R_i}$, which generates translational motion, and $\boldsymbol{\tau}_{R_i} = [\tau_{x_{R_i}}\ \tau_{y_{R_i}}\ \tau_{zR_i}]^\top \in \mathbb{R}^3$ is the total moment applied to $\mathcal{F}_{R_i}$ and expressed in $\mathcal{F}_{R_i}$, which generates rotational motion. Thus we can compute the vector of generalized forces $\mathbf{Q}_{R_i,f}(\mathbf{f}_{R_i}, \mathbf{q}) = \mathbf{f}_{R_i}^\top \frac{d\mathbf{p}_{R_i}}{d\mathbf{q}}$ and torques $\mathbf{Q}_{R_i,\tau}(\boldsymbol{\tau}_{R_i}, \mathbf{q}) = \boldsymbol{\tau}_{R_i}^\top \frac{d\mathbf{q}_{R_i}}{d\mathbf{q}}$ Then, the total vector of generalized forces and torques is given by

$$\mathbf{Q}(\mathbf{f}, \mathbf{q}) = \sum_{i=1}^{3} \mathbf{Q}_{R_i,f}(\mathbf{f}_{R_i}, \mathbf{q}) + \mathbf{Q}_{R_i,\tau}(\boldsymbol{\tau}_{R_i}, \mathbf{q})$$

At this point we compute the Lagrangian function

$$L(\mathbf{q}, \dot{\mathbf{q}}) = K(\mathbf{q}, \dot{\mathbf{q}}) - U(\mathbf{q})$$

and then the corresponding Lagrangian equation

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\mathbf{q}}} - \frac{\partial L}{\partial \mathbf{q}} = \mathbf{Q}$$

The resulting equations of motion can be then written in the form:

$$\mathbf{M}(\mathbf{q}) + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{Q}$$

where $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ contains the centrifugal and Coriolis terms, while $\mathbf{g}(\mathbf{q})$ contains the gravitational terms, and $\mathbf{Q}$ the generalized forces.

Notice that the same reasoning can be used when more then three robots transport the load, like in Chapter 4 where we considered four robots. In addition, if external forces $\mathbf{f}_i \in \mathbb{R}^3$ are applied at the end of the cables instead of using aerial robots, then the model simplifies and the terms $\mathbf{Q}_{R_i,\tau}(\boldsymbol{\tau}_{R_i}, \mathbf{q})$ disappear, while $\mathbf{f}_i$ take the place of the thrust forces $\mathbf{f}_{R_i}$.

B.2 Useful properties of continuously differentiable functions

In this section we derive some useful Lemmas related to continuously differentiable functions for which we found upper bounds on the norms.

Lemma B.2.1. *Given* $\mathbf{x} \in \mathcal{X}$*, where* $\mathcal{X} \subset \mathbb{R}^n$ *is a compact set and given a continuously differentiable function* $\mathbf{f}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^m$ *s.t.* $\mathbf{f}(\mathbf{0}) = \mathbf{0}$*, then there exists* $\alpha > 0$ *s.t.:*

$$\|\mathbf{f}(\mathbf{x})\| \leq \alpha \|\mathbf{x}\| \ \forall \mathbf{x} \in \mathcal{X}$$

Lemma B.2.2. *Given* $\mathbf{x} \in \mathcal{X}$ *and* $\mathbf{y} \in \mathcal{Y}$*, where* $\mathcal{X} \subset \mathbb{R}^n$*,* $\mathcal{Y} \subset \mathbb{R}^p$ *are compact sets and given a function that is continuously differentiable on the second argument* $\mathbf{f}(\mathbf{x}, \mathbf{y}) : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^m$ *s.t.* $\mathbf{f}(\mathbf{x}, \mathbf{0}) = \mathbf{0} \forall \mathbf{x} \in \mathcal{X}$*, then there exists* $\alpha > 0$ *s.t.:*

$$\|\mathbf{f}(\mathbf{x}, \mathbf{y})\| \leq \alpha \|\mathbf{y}\| \ \forall (\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y}$$

Lemma B.2.3. *Given* $\mathbf{x} \in \mathcal{X}$*, where* $\mathcal{X} \subset \mathbb{R}^n$ *is a compact set and given two continuously differentiable functions* $\mathbf{f}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^m$*,* $\mathbf{g}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^m$ *s.t.* $\mathbf{f}(\mathbf{0}) = \mathbf{g}(\mathbf{0}) = \mathbf{0}$*, then there exists* $\alpha > 0$ *s.t.:*

$$\|\mathbf{f}(\mathbf{x}) \cdot \mathbf{g}(\mathbf{x})\| \leq \alpha \|\mathbf{x}\|^2 \ \forall \mathbf{x} \in \mathcal{X}$$

B.3 Proof of Theorem 4.4.1

The proof of Theorem 4.4.1 follows from a direct application of Theorem 11.4 of Khalil (2002). In particular we show that the model in (4.12) satisfies the Theorem hypotheses.

In order to do that we firstly rewrite the dynamics of the system in state-space form; to do that let us define the state variables

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix} := \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix}$$

As a consequence, (4.12) when (3.8) is applied can be rewritten in state-space form as

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{z} \\ \varepsilon \dot{\mathbf{z}} = -{\mathbf{M}_\mathbf{x}}^{-1} {\mathbf{A}_\mathbf{x}}^\top \mathbf{A}_\mathbf{x} \mathbf{z} \\ \qquad\qquad -k{\mathbf{M}_\mathbf{x}}^{-1} {\mathbf{A}_\mathbf{x}}^\top \mathbf{A}_\mathbf{x} (\mathbf{x} - \mathbf{x}^r) - \varepsilon {\mathbf{M}_\mathbf{x}}^{-1} \mathbf{C}_{\mathbf{x},\mathbf{z}}\, \mathbf{z} \end{cases} \tag{B.3}$$

where $\mathbf{x}^r := \mathbf{q}^r$. System in (B.3) can be analyzed in a multitime-scale scenario Khalil (2002), where the dynamical system is characterized by slow and fast transients as far as the response to external stimuli is concerned. In particular, (B.3) can be written as

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{z}) \\ \varepsilon \dot{\mathbf{z}} = \mathbf{g}(\mathbf{x}, \mathbf{z}, \varepsilon) \end{cases}$$

that is, in the form introduced in Theorem 11.4 of Khalil (2002), except for the fact that some dependences do not appear (the function $\mathbf{f}$ for example depends only on $\mathbf{z}$ and does not depend on $(t, \mathbf{x}, \mathbf{z})$). To make the proof as self-contained as possible, we next state Theorem 11.4. Then, we will show that the hypotheses of this Theorem are satisfied by system in (B.3).

Theorem B.3.1. *Consider the singularity perturbed system*

$$\dot{x} = f(t, x, z, \varepsilon) \tag{B.4}$$

$$\varepsilon \dot{z} = g(t, x, z, \varepsilon) \tag{B.5}$$

Assume that the following assumptions are satisfied for all

$$(t, x, z, \varepsilon) \in [0, \infty) \times B_r \times [0, \varepsilon_0]$$

1. $f(t, 0, 0, \varepsilon) = 0$ *and* $g(t, 0, 0, \varepsilon) = 0$.

2. *The equation*

   $$0 = g(t, x, z, 0)$$

   has an isolated root $z = h(t, x)$ *such that* $h(t, 0) = 0$.

3. *The functions* f, g, h *and their partial derivatives up to the second order are bounded for* $y = z - h(t, x) \in B_\rho$.

4. *The origin of the reduced system*

$$\dot{x} = f(t, x, h(t, x), 0)$$

is exponentially stable.

5. *The origin of the boundary-layer system*

$$\frac{dy}{d\tau} = g(t, x, y + h(t, x), 0)$$

is exponentially stable, uniformly in (t, x).

Then, there exists $\varepsilon^* > 0$ *such that for all* $\varepsilon < \varepsilon^*$, *the origin of* (B.4)-(B.5) *is exponentially stable.*

Now we show that all the assumptions of the previous Theorem are satisfied for the system of interest. As far as the function $\mathbf{g}$ is concerned, we apply a change of coordinates defining

$$\mathbf{e_x} := \mathbf{x} - \mathbf{x}^r$$

Hence

$$\mathbf{g}(\mathbf{e_x}, \mathbf{z}, \varepsilon) = {\mathbf{M_{e_x}}}^{-1}[-(\mathbf{A}_{\mathbf{e_x}}^{\top}\mathbf{A_{e_x}})\mathbf{z} - k({\mathbf{A_{e_x}}}^{\top}\mathbf{A_{e_x}})\mathbf{e_x} - \varepsilon\mathbf{C_{e_x,z}}\,\mathbf{z}]$$

where $\mathbf{A_{e_x}}$, $\mathbf{M_{e_x}}$, $\mathbf{C_{e_x,z}}$ are compact notations for $\mathbf{A_{e_x+x^r}}$, $\mathbf{M_{e_x+x^r}}$, $\mathbf{C_{e_x+x^r,z}}$ and describe the dependence of these matrices on $\mathbf{e_x}$; we will see that our reasoning holds independently of this notation.

Assumption 1: In our scenario it is easy to verify that $\mathbf{f}(\mathbf{0}) = \mathbf{0}$ and $\mathbf{g}(\mathbf{0}, \mathbf{0}, \varepsilon) = \mathbf{0}$.

Assumption 2: In our scenario we have the equation $\mathbf{0} = \mathbf{g}(\mathbf{e_x}, \mathbf{z}, 0) = -{\mathbf{M_{e_x}}}^{-1}[(\mathbf{A}_{\mathbf{e_x}}^{\top}\mathbf{A_{e_x}})\mathbf{z} + k({\mathbf{A_{e_x}}}^{\top}\mathbf{A_{e_x}})\mathbf{e_x}]$. Since ${\mathbf{A_{e_x}}}^{\top}\mathbf{A_{e_x}}$ and $\mathbf{M_{e_x}}$ are strictly p.d. matrices by assumption, then the equation is equivalent to $\mathbf{z} + k\mathbf{e_x} = \mathbf{0} \Leftrightarrow \mathbf{z} = -k\mathbf{e_x} = \mathbf{h}(\mathbf{e_x})$ that is an isolated root. Moreover note that $\mathbf{h}(\mathbf{0}) = \mathbf{0}$.

Assumption 3: This condition is guaranteed as soon as $\mathbf{q}(t)$ belongs to $\mathcal{B}_*(\mathbf{q}^r)$.

Assumption 4: In our scenario the reduced system is

$$\dot{\mathbf{e}}_\mathbf{x} = \mathbf{f}(\mathbf{h}(\mathbf{e_x})) = -k\mathbf{e_x}$$

that is exponentially stable since $k > 0$.

Assumption 5: In our scenario the boundary-layer system is

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}\tau} = \mathbf{g}(\mathbf{e_x}, \mathbf{y} + \mathbf{h}(\mathbf{e_x}), \mathbf{0})$$

where

$$\begin{aligned}\mathbf{g}(\mathbf{e_x}, \mathbf{y} + \mathbf{h}(\mathbf{e_x}), \mathbf{0}) &= \mathbf{g}(\mathbf{e_x}, \mathbf{y} - k\mathbf{e_x}, 0) \\ &= -{\mathbf{M}_{\mathbf{e_x}}}^{-1}(\mathbf{A}_{\mathbf{e_x}}^{\top}\mathbf{A}_{\mathbf{e_x}})[(\mathbf{y} - k\mathbf{e_x}) + k\mathbf{e_x}] \\ &= -{\mathbf{M}_{\mathbf{e_x}}}^{-1}(\mathbf{A}_{\mathbf{e_x}}^{\top}\mathbf{A}_{\mathbf{e_x}})\mathbf{y}\end{aligned}$$

From now on, to keep the notation lighter, we omit the subscripts of the matrices $\mathbf{M}$ and $\mathbf{A}$. Note that $\mathbf{M}$ is positive definite, hence $\mathbf{M}^{-\frac{1}{2}} = (\mathbf{M}^{-\frac{1}{2}})^{\top}$ and

$$\begin{aligned}\mathbf{M^{-1}A}^{\top}\mathbf{A} &= \mathbf{M}^{-\frac{1}{2}}\mathbf{M}^{-\frac{1}{2}}\mathbf{A}^{\top}\mathbf{A} = \mathbf{M}^{-\frac{1}{2}}(\mathbf{M}^{-\frac{1}{2}}\mathbf{A}^{\top}\mathbf{A}\mathbf{M}^{-\frac{1}{2}})\mathbf{M}^{\frac{1}{2}} \\ &= \mathbf{M}^{-\frac{1}{2}}(\mathbf{A}\mathbf{M}^{-\frac{1}{2}})^{\top}(\mathbf{A}\mathbf{M}^{-\frac{1}{2}})\mathbf{M}^{\frac{1}{2}}\end{aligned}$$

Thus $\mathbf{M^{-1}A}^{\top}\mathbf{A}$ is similar to $(\mathbf{A}\mathbf{M}^{-\frac{1}{2}})^{\top}(\mathbf{A}\mathbf{M}^{-\frac{1}{2}})$ and, in turn, they have the same eigenvalues. Since $(\mathbf{A}\mathbf{M}^{-\frac{1}{2}})^{\top}(\mathbf{A}\mathbf{M}^{-\frac{1}{2}})$ is definite positive, then the eigenvalues of $\mathbf{M^{-1}A}^{\top}\mathbf{A}$ are positive real numbers thus implying that the matrix $-{\mathbf{M}_{\mathbf{e_x}}}^{-1}(\mathbf{A}_{\mathbf{e_x}}^{\top}\mathbf{A}_{\mathbf{e_x}})$ is Hurwitz . It follows that the boundary-layer system is exponentially stable. Since $\mathcal{B}_*(\mathbf{q}^r)$ is compact, the exponential stability property is uniform in $\mathbf{e_x}$.

The fact that the assumptions of Theorem B.3.1 are satisfied implies the existence of a Lyapunov function $V(\mathbf{e_x})$ for the reduced system and a Lyapunov function $W(\mathbf{e_x}, \mathbf{z})$ for the boundary system. Consider the function

$$\nu(\mathbf{e}_x, \mathbf{z}) = V(\mathbf{e_x}) + W(\mathbf{e_x}, \mathbf{z}).$$

Then, exploiting the properties of $\mathbf{f}$ and $\mathbf{g}$, and following the lines of proof of Theorem 11.4 of Khalil (2002), we can conclude that there exists $\bar{\epsilon} > 0$ and a neighborhood of $\mathbf{q}^r$, $\bar{\mathcal{B}}(\mathbf{q}^r) \subseteq \mathcal{B}_*(\mathbf{q}^r)$, such that for any $0 < \epsilon < \bar{\epsilon}$ and $\mathbf{q}(0) \in \bar{\mathcal{B}}(\mathbf{q}^r)$ we have that $\mathbf{q}(t) \in \mathcal{B}_*(\mathbf{q}^r)$ for all $t > 0$ and $\dot{\nu} < 0$ for all $t > 0$. This concludes the proof.

B.4 Proofs of Propositions Chapter 5

This section contains the derivation and analysis of quantities related to the stability and rate of convergence of the class of systems introduced in Chapter 5. In particular, we report the proof of Propositions 5.4.2-5.4.4. Before each proof, we recall the statement of the corresponding proposition, to facilitate the reading.

B.4.1 Proof of Proposition 5.4.2

Proposition. *The stability set of* $z\big(k,\tau;\mu,\gamma_1,\gamma_2\big)$ *is*

$$\mathcal{U} = \{\alpha < k < +\infty, 0 < \tau < \tau_s(k)\}$$

where $\alpha = k_{min}$ *was defined in Proposition 5.4.1 and*

$$\tau_s(k) = \begin{cases} \tau_{s_1}(k) & \alpha < k < \bar{k} \quad \text{if } \mu > 1 \\ \tau_{s_1}(k) & \alpha < k < \bar{k} \quad \text{if } \mu < 1 \\ \tau_{s_2}(k) & k > \bar{k} \quad \text{if } \mu < 1 \end{cases}$$

where $\tau_{s_1} := \frac{k-\alpha}{k^2\mu+k\gamma_1+\gamma_2}$, $\tau_{s_2} := \frac{-(\alpha+k)+\sqrt{(\alpha+k)^2+8(k^2\mu+k\gamma_1+\gamma_2)}}{2(k^2\mu+k\gamma_1+\gamma_2)}$, $\bar{k} = \frac{\alpha+\gamma_1+\sqrt{(\alpha+\gamma_1)^2+4\gamma_2(1-\mu)}}{2(1-\mu)}$.

Proof. In the following, we will indicate $z\big(k,\tau\big) = z\big(k,\tau;\mu,\gamma_1,\gamma_2\big)$ for convenience.
Recall that the stability time $\tau_s(k)$ is s.t. $z(k,\tau) < 1$ for $k > \alpha, 0 < \tau < \tau_s(k)$. Hence the goal now is to find an expression for $\tau_s(k)$ by studying the inequality

$$z(k,\tau) < 1$$

From the definition of the function $z(k,\tau)$, we can distinguish three cases: $k\tau < 1$, $k\tau = 1$, $k\tau > 1$.

1. $k\tau < 1$: In this case

$$z(k,\tau) = z^-(k,\tau) = 1 + \tau(\alpha - k) + \tau^2(k^2\mu + k\gamma_1 + \gamma_2)$$

We are interested to understand when $z^-(k,\tau) < 1$, that is to find the maximum τ for which $\|\mathbf{e}(hT+\tau)\| < \|\mathbf{e}(hT)\|$.

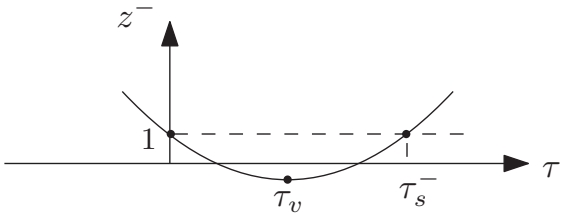


Figure B.3. Representation of $z(k,\tau)$ in the case $k\tau < 1$ and $k > \alpha$.

$$z^-(k,\tau) < 1$$

$$\Leftrightarrow 1+\tau(\alpha-k)+\tau^2(k^2\mu+k\gamma_1+\gamma_2)<1$$
$$\Leftrightarrow p_s^-(k):=\tau(\alpha-k)+\tau^2(k^2\mu+k\gamma_1+\gamma_2)<0$$

$p_s^-(k)$ represents an upward parabola with solutions

$$\tau_{s_1}^-=0,\quad \tau_{s_2}^-=\frac{k-\alpha}{k^2\mu+k\gamma_1+\gamma_2}$$

Hence $p_s^-(k)<0$ if $0<\tau<\tau_{s_2}^-$; as a consequence $z^-(k,\tau)<1$ if $0<\tau<\tau_{s_2}^-$.
Let us define

$$\tau_s^-:=\tau_{s_2}^-$$

Now let us check when $\tau_s^-<\frac{1}{k}$, that is

$$\frac{k-\alpha}{k^2\mu+k\gamma_1+\gamma_2}<\frac{1}{k}$$
$$\Leftrightarrow p_{s_2}^-(k):=k^2(1-\mu)-k(\alpha+\gamma_1)-\gamma_2<0$$

The solutions are

$$k_{s_{1,2}}^-=\frac{\alpha+\gamma_1\pm\sqrt{(\alpha+\gamma_1)^2+4\gamma_2(1-\mu)}}{2(1-\mu)}$$

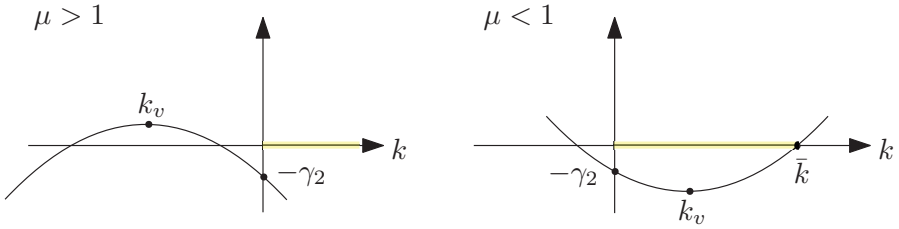


Figure B.4. Representation of $p_{s_2}^-(k)$.

- if $\mu<1$, then $p_{s_2}^-(k)$ is an upward parabola such that the solutions and the vertex k_v satisfy

$$\sum \text{sol}=\frac{\alpha+\gamma_1}{1-\mu}>0,\quad \prod \text{sol}=\frac{-\gamma_2}{1-\mu}<0$$

$$k_v=\frac{\alpha+\gamma_1}{2(1-\mu)}>0$$

Hence, if $\mu<1$, then $\tau_s^-<\frac{1}{k}$ for $\alpha<k<\bar{k}$, where

$$\bar{k}=\max\{k_{s_1}^-,k_{s_2}^-\}=$$
$$\begin{cases}\frac{\alpha+\gamma_1+\sqrt{(\alpha+\gamma_1)^2+4\gamma_2(1-\mu)}}{2(1-\mu)} & \text{if } \mu<1\\ \frac{\alpha+\gamma_1-\sqrt{(\alpha+\gamma_1)^2+4\gamma_2(1-\mu)}}{2(1-\mu)} & \text{if } \mu>1\end{cases} \tag{B.6}$$

In this case $\mu<1$, hence $\bar{k}=\frac{\alpha+\gamma_1+\sqrt{(\alpha+\gamma_1)^2+4\gamma_2(1-\mu)}}{2(1-\mu)}$.

- if $\mu > 1$, then $p^-_{s_2}(k)$ is a downward parabola such that

$$\sum \text{sol} < 0, \quad \prod \text{sol} > 0$$

$$k_v < 0$$

Hence, if $\mu > 1$, then $\tau_s^- < \frac{1}{k} \ \ \forall \ \ k > \alpha$.

In conclusion,

$$z^-(k,\tau) < 1 \text{ if } 0 < \tau < \tau_s^-$$

$$\tau_s^- < \frac{1}{k} \text{ if } \begin{cases} \alpha < k < \bar{k} & \text{if } \mu < 1 \\ k > \alpha & \text{if } \mu > 1 \end{cases}$$

2. $k\tau = 1$: In this case

$$z\left(k, \frac{1}{k}\right) = \mu + \frac{1}{k}(\alpha + \gamma_1) + \frac{\gamma_2}{k^2}$$

For which values of k is $z\left(k, \frac{1}{k}\right) < 1$?

$$\mu + \frac{1}{k}(\alpha + \gamma_1) + \frac{\gamma_2}{k^2} < 1$$
$$p_s^=(k) := k^2(\mu - 1) + k(\alpha + \gamma_1) + \gamma_2 < 0 \tag{B.7}$$

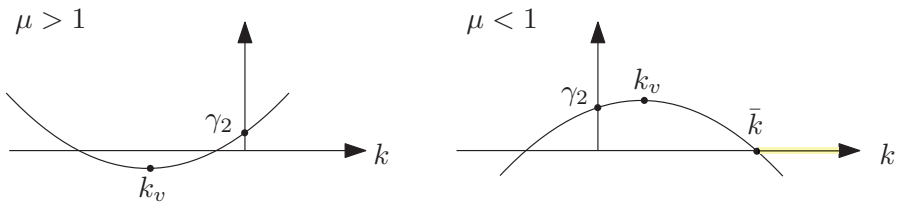


Figure B.5. Representation of $p_s^=$.

- $\mu < 1$: $p_s^=$ is a concave downward parabola. In $k = 0$ the parabola intersects the y-axis in $\gamma_2 > 0$. Now the vertex of the parabola is in $k_v^= > 0$, the sum of the solutions is $\sum \text{sol} > 0$ and product $\prod \text{sol} < 0$. Finally, $z\left(k, \frac{1}{k}\right) < 1$ where $p_s^=(k) < 0$, that is for $k > \bar{k}$, where $\bar{k} := \max\{k_{s_1}^=, k_{s_2}^=\}$ has the same value that we have found in (B.6).
- $\mu > 1$: $p_s^=$ is a concave upward parabola. In $k = 0$ the parabola intersects the y-axis in $\gamma_2 > 0$. The vertex of the parabola is in

$$k_v^= = \frac{\alpha + \gamma_1}{2(1-\mu)} < 0$$

The sum and product of the solutions are

$$\sum \text{sol} = \frac{\alpha + \gamma_1}{1 - \mu} < 0, \quad \prod \text{sol} = \frac{\gamma_2}{\mu - 1} > 0$$

hence the solutions are both negative and $p_s^=(k)$ is always positive, so $z\left(k, \frac{1}{k}\right) >$

$1\ \forall k > \alpha$ if $\mu > 1$.

In conclusion:

$$z(k,1/k) < 1 \text{ for } \begin{cases} \nexists\ k > \alpha & \text{if } \mu > 1 \\ k > \bar{k} & \text{if } \mu < 1 \end{cases}$$

3. $k\tau > 1$: In this case

$$z\Big(k,\tau\Big) = z^{+}\Big(k,\tau\Big) = -1 + \tau(\alpha + k) + \tau^2(k^2\gamma_1 + k\gamma_2 + \gamma_3)$$

This is an upward parabola with vertex

$$\tau_v^{+} = -\frac{k+\alpha}{2(k^2\gamma_1 + k\gamma_2 + \gamma_3)} < 0$$

Notice that $z^{+}(k,0) = -1 < 0$ and

$$\sum \text{sol} = -\frac{\alpha + k}{k^2\mu + k\gamma_1 + \gamma_2} < 0$$
$$\prod \text{sol} = -\frac{1}{k^2\mu + k\gamma_1 + \gamma_2} < 0$$

For which values of k is $z^{+}(k,\tau) < 1$?

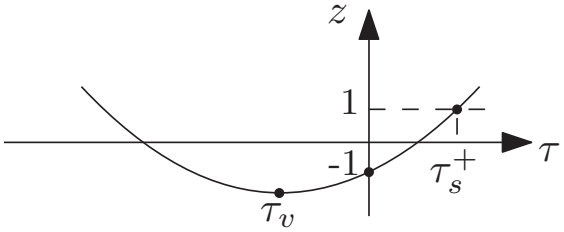


Figure B.6. Representation of $z(k,\tau)$ in the case $k\tau > 1$.

$$\begin{aligned} & z^{+}(k,\tau) < 1 \\ & \Leftrightarrow -1 + \tau(\alpha + k) + \tau^2(k^2\mu + k\gamma_1 + \gamma_2) < 1 \\ & \Leftrightarrow p_s^{+}(k) := \tau^2(k^2\mu + k\gamma_1 + \gamma_2) + \tau(\alpha + k) - 2 < 0 \end{aligned}$$

$p_s^{+}(k)$ represents an upward parabola with solutions s.t.

$$\sum \text{sol} = -\frac{\alpha + k}{k^2\mu + k\gamma_1 + \gamma_2} < 0$$
$$\prod \text{sol} = -\frac{2}{k^2\mu + k\gamma_1 + \gamma_2} < 0$$

$$\tau_{s_1}^{+} = \frac{-(\alpha + k) - \sqrt{(\alpha + k)^2 + 8(k^2\mu + k\gamma_1 + \gamma_2)}}{2(k^2\mu + k\gamma_1 + \gamma_2)}$$
$$\tau_{s_2}^{+} = \frac{-(\alpha + k) + \sqrt{(\alpha + k)^2 + 8(k^2\mu + k\gamma_1 + \gamma_2)}}{2(k^2\mu + k\gamma_1 + \gamma_2)}$$

We define

$$\tau_s^+ := \max\{\tau_{s_{1,2}}^+\} = \tau_{s_2}^+$$

and $z^+(k,\tau) < 1$ for $0 < \tau < \tau_s^+$. Notice that in this case we have to check for which k it holds $\tau_s^+ > \frac{1}{k}$ since we are analyzing the case $k\tau > 1$:

$$\begin{aligned}
&\tau_s^+ > \frac{1}{k} \\
&\frac{-(\alpha+k)+\sqrt{(\alpha+k)^2+8(k^2\mu+k\gamma_1+\gamma_2)}}{2(k^2\mu+k\gamma_1+\gamma_2)} > \frac{1}{k} \\
&-k(\alpha+k)+k\sqrt{(\alpha+k)^2+8(k^2\mu+k\gamma_1+\gamma_2)} > \\
&> 2(k^2\mu+k\gamma_1+\gamma_2) \\
&k^2((\alpha+k)^2+8(k^2\mu+k\gamma_1+\gamma_2)) > k^2(\alpha+k)^2+ \\
&+4(k^2\mu+k\gamma_1+\gamma_2)^2+4k(\alpha+k)(k^2\mu+k\gamma_1+\gamma_2) \\
&8k^2(k^2\mu+k\gamma_1+\gamma_2) > 4(k^2\mu+k\gamma_1+\gamma_2)^2+ \\
&+4k(\alpha+k)(k^2\mu+k\gamma_1+\gamma_2) \\
&2k^2 > (k^2\mu+k\gamma_1+\gamma_2)+k(\alpha+k) \\
&k^2(1-\mu)-k(\alpha+\gamma_1)-\gamma_2 > 0 \\
&p_{\tau_s^+}(k) := k^2(\mu-1)+k(\alpha+\gamma_1)+\gamma_2 < 0
\end{aligned}$$

The solutions are

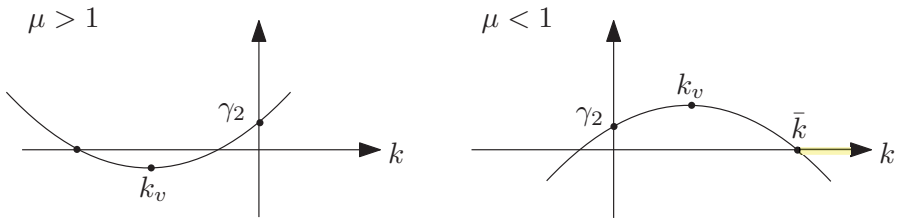


Figure B.7. Representation of $p_{\tau_s^+}(k)$.

$$k_\pm = \frac{\alpha+\gamma_1 \pm \sqrt{(\alpha+\gamma_1)^2+4\gamma_2(1-\mu)}}{2(1-\mu)}$$

- $\mu > 1$: $p_{\tau_s^+}(k)$ is an upward parabola s.t. for $k = 0$ it intersects the y-axis in γ_2 and

  $$\sum \text{sol} = -\frac{\alpha+\gamma_1}{\mu-1} < 0, \quad \prod \text{sol} = \frac{\gamma_2}{\mu-1} < 0$$

  As a consequence, this parabola is never negative for $k > 0$, hence $\tau_s^+ < \frac{1}{k}\ \forall k > 0$.

- $\mu < 1$: $p_{\tau_s^+}(k)$ is a downward parabola and

  $$\sum \text{sol} > 0, \quad \prod \text{sol} > 0$$

  As a consequence, it is negative for $k > \bar{k}$, hence $\tau_s^+ > \frac{1}{k}$ for $k > \bar{k} = \max\{k_\pm\}$, that is the same value found in (B.6).

In conclusion,

$$z^+(k,\tau) < 1 \text{ if } 0 < \tau < \tau_s^+$$

$$\tau_s^+ \begin{cases} < \frac{1}{k} & \text{for } k > \alpha \quad \text{if } \mu > 1 \\ > \frac{1}{k} & \text{for } k > \bar{k} \quad \text{if } \mu < 1 \end{cases}$$

Notice that only if $\mu < 1$, then τ_s^+ belongs to the region $k\tau > 1$ that we are considering. Otherwise, we have to consider τ_s^- instead.

Hence $\tau_s = \tau_s^-$ if $\mu > 1$ or if $\mu < 1$ and $\alpha < k < \bar{k}$, whereas $\tau_s = \tau_s^+$ if $\mu < 1$ and $k > \bar{k}$, that is

$$\tau_s(k;\mu,\gamma_1,\gamma_2) = \frac{k-\alpha}{k^2\mu + k\gamma_1 + \gamma_2} \text{ if } \mu > 1$$

$$\tau_s(k;\mu,\gamma_1,\gamma_2) =$$

$$= \begin{cases} \frac{k-\alpha}{k^2\mu+k\gamma_1+\gamma_2} & \text{for } \alpha < k < \bar{k} \\ \frac{-(\alpha+k)+\sqrt{(\alpha+k)^2+8(k^2\mu+k\gamma_1+\gamma_2)}}{2(k^2\mu+k\gamma_1+\gamma_2)} & \text{for } k > \bar{k} \end{cases} \quad \text{if } \mu < 1$$

■

B.4.2 Proof of Proposition 5.4.3

Proposition. *Let us assume to fix k, then the optimal time $\tau_o(k) := \tau_o(k;\mu,\alpha,\gamma_1,\gamma_2)$ defined in (5.28) can be computed as:*

$$\tau_o(k) = \begin{cases} \tau_{o_1}(k) & \text{for } \alpha < k < \bar{\bar{k}} \quad \text{if } \mu > \frac{1}{2} \\ \tau_{o_1}(k) & \text{for } \alpha < k < \bar{\bar{k}} \quad \text{if } \mu < \frac{1}{2} \\ \frac{1}{k} & \text{for } k > \bar{\bar{k}} \quad \text{if } \mu < \frac{1}{2} \end{cases}$$

where $\tau_{o_1}(k) := \frac{k-\alpha}{2(k^2\mu+k\gamma_1+\gamma_2)}$ *and* $\bar{\bar{k}} = \frac{\alpha+2\gamma_1+\sqrt{(\alpha+2\gamma_1)^2+8\gamma_2(1-2\mu)}}{2(1-2\mu)}$.

The corresponding convergence rate $\rho_o(k) := \rho_o(k;\mu,\alpha,\gamma_1,\gamma_2)$ is:

$$\rho_o(k) = \begin{cases} \rho_{k_1}(k) & & \text{if } \mu > \frac{1}{2} \\ \rho_{k_1}(k) & \text{for } \alpha < k < \bar{\bar{k}} & \text{if } \mu < \frac{1}{2} \\ \rho_{k_2}(k) & \text{for } k > \bar{\bar{k}} & \text{if } \mu < \frac{1}{2} \end{cases}$$

where $\rho_{k_1}(k) := 1 - \frac{(\alpha-k)^2}{4(k^2\mu+k\gamma_1+\gamma_2)}$ *and* $\rho_{k_2}(k) := \mu + \frac{1}{k}(\alpha+\gamma_1) + \frac{\gamma_2}{k^2}$.

Proof. In the following, we will indicate $z(k,\tau) = z(k,\tau;\mu,\gamma_1,\gamma_2)$ for convenience.
Recall that the optimal time $\tau_o(k)$ is s.t. $z(k,\tau_o(k))$ reaches its minimum value. Hence the goal now is to find an expression for $\tau_o(k)$.
From the definition of the function $z(k,\tau)$, we can distinguish two cases: $k\tau < 1$, $k\tau > 1$.

1. $k\tau < 1$: In this case

$$z(k,\tau) = z^-(k,\tau) = 1 + \tau(\alpha - k) + \tau^2(k^2\mu + k\gamma_1 + \gamma_2)$$

$z(k,\tau)$ is an upward parabola (see Fig.B.3) with vertex at

$$\tau_v^- = \frac{k-\alpha}{2(k^2\mu + k\gamma_1 + \gamma_2)} \tag{B.8}$$

Notice that $\tau_v^- > 0$ since $k > \alpha$ (Prop.**??**). The value of the function at the vertex corresponds to its minimum:

$$\begin{aligned} z(k,\tau_v) &= 1 - \frac{(k-\alpha)^2}{2(k^2\mu + k\gamma_1 + \gamma_2)} + \\ &+ \frac{(k-\alpha)^2}{4(k^2\mu + k\gamma_1 + \gamma_2)^{\cancel{2}}} \cancel{(k^2\mu + k\gamma_1 + \gamma_2)} = \\ &= 1 - \frac{1}{4}\frac{(k-\alpha)^2}{(k^2\mu + k\gamma_1 + \gamma_2)} < 1 \end{aligned}$$

Now we want to check if $\tau_v^- < \frac{1}{k}$, that is if the minimum value belongs to the region we are analyzing:

$$\begin{aligned} \tau_v^- &= \frac{k-\alpha}{2(k^2\mu + k\gamma_1 + \gamma_2)} < \frac{1}{k} \\ &\Leftrightarrow k^2 - k\alpha < 2\mu k^2 + 2\gamma_1 k + 2\gamma_2 \\ &\Leftrightarrow p_v^-(k) := k^2(1-2\mu) - k(\alpha + 2\gamma_1) - 2\gamma_2 < 0 \end{aligned} \tag{B.9}$$

where the solutions of $p_v^-(k)$

$$k_\pm = \frac{\alpha + 2\gamma_1 \pm \sqrt{(\alpha+2\gamma_1)^2 + 8\gamma_2(1-2\mu)}}{2(1-2\mu)}$$

are s.t.

$$\sum \text{sol} = \frac{\alpha + 2\gamma_1}{1-2\mu}, \quad \prod \text{sol} = \frac{-2\gamma_2}{1-2\mu}$$

where there exist two different solutions if $(\alpha+2\gamma_1)^2 + 8\gamma_2(1-2\mu) > 0$, that is

$$\mu < \frac{(\alpha+2\gamma_1)^2}{16\gamma_2} + \frac{1}{2}$$

otherwise τ_v^- is always greater than $1/k$.

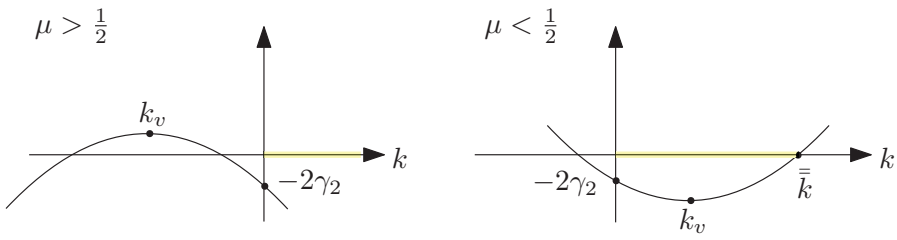


Figure B.8. Representation of $p_v^-(k)$.

- if $1-2\mu > 0$, that is $\mu < \frac{1}{2}$, then $p_v^-(k)$ is an upward parabola with solutions

$k_\pm$ s.t. $\sum \text{sol} > 0$, $\prod \text{sol} < 0$ and $\tau_v < \frac{1}{k}$ if $\alpha < k < \bar{\bar{k}}$, where

$$\bar{\bar{k}} := \max\{k_-, k_+\} = \\ = \begin{cases} \frac{\alpha+2\gamma_1+\sqrt{(\alpha+2\gamma_1)^2+8\gamma_2(1-2\mu)}}{2(1-2\mu)} & \text{if} \quad \mu < \frac{1}{2} \\ \frac{\alpha+2\gamma_1>\sqrt{(\alpha+2\gamma_1)^2+8\gamma_2(1-2\mu)}}{2(1-2\mu)} & \text{if} \quad \mu > \frac{1}{2} \end{cases} \tag{B.10}$$

- if $1-2\mu < 0$, that is $\mu > \frac{1}{2}$, then $p_v^-(k)$ is a downward parabola with solutions $k_\pm$ s.t. $\sum \text{sol} < 0$, $\prod \text{sol} > 0$ and $\tau_v < \frac{1}{k}$ if $k > \alpha$.

In conclusion,

$$\tau_v^- < \frac{1}{k} \text{ if } \begin{cases} \alpha < k < \bar{\bar{k}} & \text{if} \quad \mu < \frac{1}{2} \\ k > \alpha & \text{if} \quad \mu > \frac{1}{2} \end{cases}$$

So τ_v^- represents the optimal time in the case $k\tau < 1$ for $\alpha < k < \bar{\bar{k}}$ if $\mu < \frac{1}{2}$ and for $k > \alpha$ if $\mu > \frac{1}{2}$.

2. $k\tau > 1$. In this case

$$z\Big(k,\tau\Big) = z^+\Big(k,\tau\Big) = -1+\tau(\alpha+k)+\tau^2(k^2\gamma_1+k\gamma_2+\gamma_3)$$

we still have an upward parabola with vertex

$$\tau_v^+ = -\frac{k+\alpha}{2(k^2\gamma_1+k\gamma_2+\gamma_3)} < 0$$

Since $\tau_v^+ < 0$ it cannot be a valid optimal time and, since $z(k,\tau)$ is a monotonically increasing function in k and τ, then the minimum can be detected in $\tau = 1/k$.

Finally $\tau_o(k) = \tau_v^-(k)$ for $\alpha < k < \bar{\bar{k}}$ if $\mu < \frac{1}{2}$ and for $k > \alpha$ if $\mu > \frac{1}{2}$, whereas $\tau_o(k) = 1/k$ for $k > \bar{\bar{k}}$ if $\mu < \frac{1}{2}$, that is

$$\tau_o(k;\mu,\gamma_1,\gamma_2) = \frac{k-\alpha}{2(k^2\mu+k\gamma_1+\gamma_2)} \text{ if } \mu > \frac{1}{2}$$

$$\tau_o(k;\mu,\gamma_1,\gamma_2) = \begin{cases} \frac{k-\alpha}{2(k^2\mu+k\gamma_1+\gamma_2)} & \text{for } \alpha < k < \bar{\bar{k}} \\ \frac{1}{k} & \text{for } k > \bar{\bar{k}} \end{cases} \quad \text{if } \mu < \frac{1}{2}$$

Computation of the convergence rate

The convergence rate, defined in (5.29), corresponds to the function $z(k,\tau)$ evaluated in $\tau = \tau_o(k)$:

$$\rho(k) = z(k,\tau_o)$$

We have already seen that, depending on the value of the parameter μ, then $\tau_o(k)$ has different values and $z(k,\tau_o(k))$ too:

- $0 < \mu < \frac{1}{2}$:

$$\tau_o = \begin{cases} \tau_v^- & \text{if } \alpha < k < \bar{\bar{k}} \\ \frac{1}{k} & \text{if } k > \bar{\bar{k}} \end{cases}$$

and

$$\rho(k) = z^-(k, \tau_o) = \\ = \begin{cases} z^-(k, \tau_v^-) = 1 - \frac{(\alpha-k)^2}{4(k^2\mu + k\gamma_1 + \gamma_2)} & \text{if } \alpha < k < \bar{\bar{k}} \\ z(k, \frac{1}{k}) = \mu + \frac{1}{k}(\alpha + \gamma_1) + \frac{1}{k^2}\gamma_2 & \text{if } k > \bar{\bar{k}} \end{cases}$$

Notice that $z^-(k, \tau_o) < 1$ always and $z(k, \frac{1}{k}) < 1$ in the case it is considered ($\mu < \frac{1}{2}$). Moreover, note that for $\gamma_1 = \gamma_2 = 0$, $z^-(k, \frac{1}{k}) = \mu + \frac{\alpha}{k}$ that tends to μ for large values of k:

$$\rho(k) \to \begin{cases} 1 - \frac{1}{4\mu} & \text{if } \alpha < k < \bar{\bar{k}} \\ \mu & \text{if } k > \bar{\bar{k}} \end{cases}$$

- $\mu > \frac{1}{2}$: in this case

$$\rho = z^-(k, \tau_v^-) = 1 - \frac{(\alpha - k)^2}{4(k^2\mu + k\gamma_1 + \gamma_2)} \quad \forall \quad k > \alpha$$

and for $\gamma_1 = \gamma_2 = 0$ and big values of k:

$$\rho(k) \to 1 - \frac{1}{4\mu} \quad \forall \quad k > \alpha$$

■

B.4.3 Proof of Proposition 5.4.4

Proposition. *Let us assume to fix τ, then the optimal $k_o(\tau) := k_o(\tau; \mu, \alpha, \gamma_1, \gamma_2)$, defined in (5.30), can be computed as:*

$$k_o(\tau) = \begin{cases} \frac{1-\tau\gamma_1}{2\tau\mu} & \text{for } \tau < \frac{1}{\gamma_1} & \text{if } \mu > \frac{1}{2} \\ \frac{1-\tau\gamma_1}{2\tau\mu} & \text{for } \tau_{m_k} < \tau < \tau_{M_k} & \text{if } \mu < \frac{1}{2} \\ \frac{1}{\tau} & \text{for } 0 < \tau < \tau_{m_k} & \text{if } \mu < \frac{1}{2} \end{cases}$$

where $\tau_{m_k} := \frac{1-2\mu}{\gamma_1}$, $\tau_{M_k} := \frac{1}{\gamma_1}$ and the corresponding convergence rate $\rho_o(\tau) := \rho_o(\tau; \mu, \alpha, \gamma_1, \gamma_2)$ is:

$$\rho_o(\tau) = \begin{cases} \rho_{\tau_1}(\tau) & \text{for } \tau < \tau_{v_2}^- & \text{if } \mu > \frac{1}{2} \\ \rho_{\tau_1}(\tau) & \text{for } \tau_{m_\tau} < \tau < \tau_{M_\tau} & \text{if } \mu < \frac{1}{2} \\ \rho_{\tau_2}(\tau) & \text{for } 0 < \tau < \tau_{m_\tau} & \text{if } \mu < \frac{1}{2} \end{cases}$$

where $\tau_{m_\tau} := \min\{\frac{1-2\mu}{\gamma_1}, \tau_{v_2}^-\}$, $\tau_{M_\tau} := \max\{\frac{1-2\mu}{\gamma_1}, \tau_{v_2}^-\}$, $\rho_{\tau_1}(\tau) := \frac{(-\gamma_1^2 + 4\gamma_2\mu)\tau^2 + 2(\gamma_1 + 2\alpha\mu)\tau + 4\mu - 1}{4\mu}$, $\rho_{\tau_2}(\tau) := \gamma_2\tau^2 + (\alpha + \gamma_1)\tau + \mu$ and $\tau_{v_2}^- = \frac{-(\gamma_1 + \alpha\mu) + \sqrt{(\gamma_1 + \alpha\mu)^2 + (-\gamma_1^2 + 4\gamma_2\mu)}}{-\gamma_1^2 + 4\gamma_2\mu}$.

Proof. 1. $k\tau < 1$:

Let us rewrite the function $z^-(k, \tau)$ s.t. it depends on k, since τ is assumed to be

fixed now.

$$z^-(k,\tau) = \tau^2\mu k^2 + \tau(\tau\gamma_1 - 1)k + 1 + \tau\alpha + \tau^2\gamma_2 \tag{B.11}$$

The function represents an upward parabola with vertex

$$k_v^- = \frac{1-\tau\gamma_1}{2\tau\mu} > 0 \text{ if } \tau < \frac{1}{\gamma_1}$$

Let us check when $k_v^- < \frac{1}{\tau}$:

$$\frac{1-\tau\gamma_1}{2\tau\mu} < \frac{1}{\tau} \quad \Leftrightarrow \quad \tau > \frac{1-2\mu}{\gamma_1}$$

Observe that if $\mu > \frac{1}{2}$, then $\frac{1-2\mu}{\gamma_1} < 0$, hence $\tau > \frac{1-2\mu}{\gamma_1}$ always and $k_v^- < \frac{1}{\tau}$.
In conclusion,

$$k_v^- < \frac{1}{\tau} \text{ if } \tau > \frac{1-2\mu}{\gamma_1}$$

Now we find out when $z^-(k_v^-,\tau) < 1$. First of all we compute the expression $z^-(k_v^-,\tau)$:

$$\begin{aligned}
z^-(k_v^-,\tau) &= \cancel{\tau^2}\cancel{\mu}\frac{(1-\tau\gamma_1)^2}{4\cancel{\tau^2}\mu^{\cancel{2}}} + \cancel{\tau}(\tau\gamma_1 - 1)\frac{1-\tau\gamma_1}{2\cancel{\tau}\mu} + \tau^2\gamma_2 + \\
&+ \tau\alpha + 1 = \\
&= \frac{(1-\tau\gamma_1)^2}{4\mu} - \frac{(1-\tau\gamma_1)^2}{2\mu} + \tau^2\gamma_2 + \tau\alpha + 1 \\
&= -\frac{(1-\tau\gamma_1)^2}{4\mu} + \tau^2\gamma_2 + \tau\alpha + 1 \\
&= \frac{(-\gamma_1^2 + 4\gamma_2\mu)\tau^2 + 2(\gamma_1 + 2\alpha\mu)\tau + 4\mu - 1}{4\mu}
\end{aligned}$$

Now, under which conditions does it hold $z^-(k_v^-,\tau) < 1$?

$$\begin{aligned}
&\frac{(-\gamma_1^2 + 4\gamma_2\mu)\tau^2 + 2(\gamma_1 + 2\alpha\mu)\tau + 4\mu - 1}{4\mu} < 1 \\
&\Leftrightarrow (-\gamma_1^2 + 4\gamma_2\mu)\tau^2 + 2(\gamma_1 + 2\alpha\mu)\tau + 4\mu - 1 < 4\mu \\
&p_v^-(\tau) := (-\gamma_1^2 + 4\gamma_2\mu)\tau^2 + 2(\gamma_1 + 2\alpha\mu)\tau - 1 < 0
\end{aligned} \tag{B.12}$$

$p_v^-(\tau)$ is a parabola in τ with vertex in

$$\tau_v^- = -\frac{\gamma_1 + 2\alpha\mu}{-\gamma_1^2 + 4\gamma_2\mu} > 0 \quad \text{if} \quad \mu < \frac{\gamma_1^2}{4\gamma_2}$$

The solutions of (B.12) are

$$\tau_{v_1}^- = \frac{-(\gamma_1 + \alpha\mu) - \sqrt{(\gamma_1 + \alpha\mu)^2 + (-\gamma_1^2 + 4\gamma_2\mu)}}{-\gamma_1^2 + 4\gamma_2\mu}$$

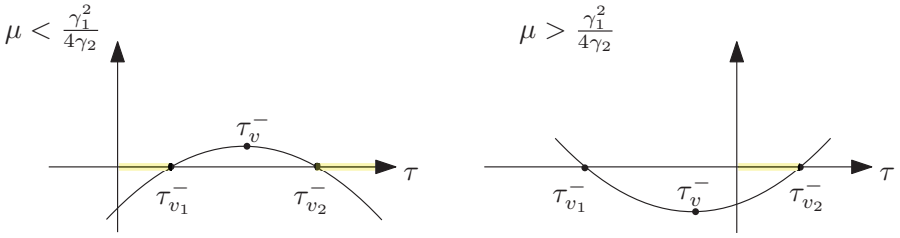


Figure B.9. Representation of $p_v^-(\tau)$.

$$\tau_{v_2}^- = \frac{-(\gamma_1+\alpha\mu)+\sqrt{(\gamma_1+\alpha\mu)^2+(-\gamma_1^2+4\gamma_2\mu)}}{-\gamma_1^2+4\gamma_2\mu}$$

Notice that

$$\sum \text{sol} = \frac{\gamma_1+\alpha\mu}{\gamma_1^2-4\gamma_2\mu} > 0 \quad \text{if} \quad \mu < \frac{\gamma_1^2}{4\gamma_2}$$
$$\prod \text{sol} = \frac{1}{\gamma_1^2-4\gamma_2\mu} > 0 \quad \text{if} \quad \mu < \frac{\gamma_1^2}{4\gamma_2}$$

Moreover the argument of the square root is always positive and $\tau_{v_1}^- < \tau_{v_2}^-$ if $\mu > \frac{\gamma_1^2}{4\gamma_2}$. Finally, the concavity of $p_v^-(\tau)$ is upward if $\mu > \frac{\gamma_1^2}{4\gamma_2}$, otherwise it is downward. You can see these results in Fig.B.9.
In conclusion, $z^-(k_v^-,\tau)$:

$$\begin{cases} <1 \quad \text{for} \quad 0<\tau<\tau_{v_2}^- \ \vee \ \tau>\tau_{v_1}^- & \text{if } \mu < \frac{\gamma_1^2}{4\gamma_2} \\ <1 \quad \text{for} \quad 0<\tau<\tau_{v_2}^- & \text{if } \mu > \frac{\gamma_1^2}{4\gamma_2} \end{cases}$$

2. $k\tau = 1$
Even if the stability analysis has been performed in the scenario where k was fixed, we will analyze the stability on the curve $k = \frac{1}{\tau}$ because the result will be useful when we will compute the convergence rate.

$$z(\frac{1}{\tau},\tau) = \gamma_2\tau^2 + (\alpha+\gamma_1)\tau + \mu$$

Let us check when $z(\frac{1}{\tau},\tau) < 1$:

$$\gamma_2\tau^2 + (\alpha+\gamma_1)\tau + \mu < 1$$
$$\Leftrightarrow p^=(\tau) := \gamma_2\tau^2 + (\alpha+\gamma_1)\tau + \mu - 1 < 0$$

$p^=(\tau)$ is an upward parabola with vertex

$$\tau_v^= = \frac{-(\alpha+\gamma_1)}{2\gamma_2} < 0$$

and solutions

$$\tau_{s_{1,2}}^= = \frac{-(\alpha+\gamma_1) \pm \sqrt{(\alpha+\gamma_1)^2-4\gamma_2(\mu-1)}}{2\gamma_2}$$

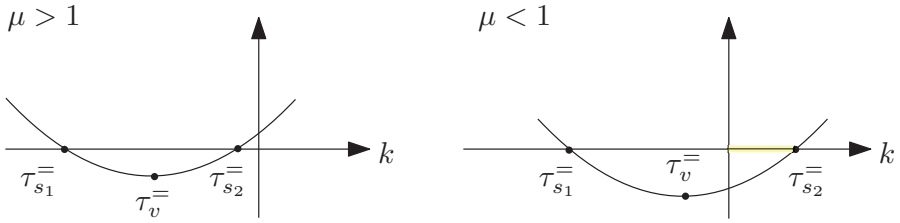


Figure B.10. Representation of $p_s^{=}(k,\tau)$.

s.t.

$$\sum \text{sol} = -\frac{\alpha+\gamma_1}{\gamma_2} < 0$$
$$\prod \text{sol} = \frac{\mu-1}{\gamma_2} > 0 \quad \text{if} \quad \mu > 1$$

In conclusion:

$$z\Big(\frac{1}{\tau},\tau\Big) < 1 \text{ for } \begin{cases} \nexists\, \tau > 0 & \text{if } \mu > 1 \\ 0 < \tau < \tau_{s_2}^{=} & \text{if } \mu < 1 \end{cases}$$

3. $k\tau > 1$:
   Let us rewrite the function $z^+(k,\tau)$ s.t. it depends on k, since τ is assumed to be fixed now.

$$z^+(k,\tau) = \tau^2\mu k^2 + \tau(\tau\gamma_1+1)k + \tau^2\gamma_2 k + \tau\alpha - 1 \tag{B.13}$$

   The function represents an upward parabola with vertex

$$k_v^+ = -\frac{1+\tau\gamma_1}{2\tau\mu} < 0 \text{ for } \tau > 0$$

   Hence k_v^+ never belongs to the region $k\tau > 1$.

In conclusion $k_o(\tau) = k_v^-$ for $0 < \tau < \frac{1}{\gamma_1}$ if $\mu > \frac{1}{2}$ and for $\frac{1-2\mu}{\gamma_1} < \tau < \frac{1}{\gamma_1}$ if $\mu < \frac{1}{2}$; instead, $k_o(\tau) = \frac{1}{\tau}$ for $0 < \tau < \frac{1-2\mu}{\gamma_1}$ if $\mu < \frac{1}{2}$, that is

$$k_o(\tau;\mu,\gamma_1,\gamma_2) = \frac{1-\tau\gamma_1}{2\tau\mu} \quad \text{for } \tau < \frac{1}{\gamma_1} \quad \text{if } \mu > \frac{1}{2}$$
$$k_o(\tau;\mu,\gamma_1,\gamma_2) - = \begin{cases} \frac{1-\tau\gamma_1}{2\tau\mu} & \text{for } \frac{1-2\mu}{\gamma_1} < \tau < \frac{1}{\gamma_1} \\ \frac{1}{\tau} & \text{for } 0 < \tau < \frac{1-2\mu}{\gamma_1} \end{cases} \quad \text{if } \mu < \frac{1}{2}$$

Computation of the convergence rate

The convergence rate, defined in (5.31), corresponds to the function $z(k,\tau)$ evaluated in $\tau = \tau_o(k)$:

$$\rho(\tau) = z(k_o,\tau)$$

We have already seen that, depending on the value of the parameter μ, then $k_o(\tau)$ has different values and $z(k_o(\tau),\tau)$ too:

- $0 < \mu < \frac{1}{2}$:

$$k_o = \begin{cases} k_v^- & \text{for } \frac{1-2\mu}{\gamma_1} < \tau < \frac{1}{\gamma_1} \\ \frac{1}{\tau} & \text{for } 0 < \tau < \frac{1-2\mu}{\gamma_1} \end{cases}$$

and

$$\begin{aligned} &\rho(\tau) = z^-(k_o, \tau) = \\ &= \begin{cases} z^-(k_v^-, \tau) & \text{for } \frac{1-2\mu}{\gamma_1} < \tau < \frac{1}{\gamma_1} \\ z(\frac{1}{\tau}, \tau) & \text{for } 0 < \tau < \frac{1-2\mu}{\gamma_1} \end{cases} \\ &= \begin{cases} \frac{(-\gamma_1^2 + 4\gamma_2\mu)\tau^2 + 2(\gamma_1 + 2\alpha\mu)\tau + 4\mu - 1}{4\mu} & \text{for } \frac{1-2\mu}{\gamma_1} < \tau < \frac{1}{\gamma_1} \\ \gamma_2\tau^2 + (\alpha + \gamma_1)\tau + \mu & \text{for } 0 < \tau < \frac{1-2\mu}{\gamma_1} \end{cases} \end{aligned}$$

- $\mu > \frac{1}{2}$: in this case

$$\begin{aligned} &\rho(\tau) = z^-(k_v^-, \tau) = \\ &= \frac{(-\gamma_1^2 + 4\gamma_2\mu)\tau^2 + 2(\gamma_1 + 2\alpha\mu)\tau + 4\mu - 1}{4\mu} \\ &\text{for } \ 0 < \tau < \frac{1}{\gamma_1} \end{aligned}$$

■

Bibliography

Elistair | the tethered drone company. https://elistair.com/. (Accessed on 07/13/2020).

Abdolhosseini M., Zhang Y. M., and Rabbath C. A. An efficient model predictive control scheme for an unmanned quadrotor helicopter. *Jour. of Intelligent & Robotic Systems*, 70(1):27–38, Apr 2013. ISSN 1573-0409.

Alexis K., Nikolakopoulos G., and Tzes A. Model predictive quadrotor control: attitude, altitude and position experimental studies. *IET Control Theory Applications*, 6(12):1812–1827, 2012.

Alexis K., Nikolakopoulos G., and Tzes A. On trajectory tracking model predictive control of an unmanned quadrotor helicopter subject to aerodynamic disturbances. *Asian Journal of Control*, 16(1):209–224, 2012.

Baizid K., Caccavale F., Chiaverini S., Giglio G., and Pierri F. Safety in coordinated control of multiple unmanned aerial vehicle manipulator systems: Case of obstacle avoidance. In *22nd Mediterranean Conference on Control and Automation*, pages 1299–1304, 2014.

Bicego D. *Design and Control of Multi-Directional Thrust Multi-Rotor Aerial Vehicles with applications to Aerial Physical Interaction Tasks*. Theses, INSA de Toulouse, September 2019. URL https://hal.laas.fr/tel-02433940.

Bock H. G. and Plitt K.-J. A multiple shooting algorithm for direct solution of optimal control problems. In *Proceedings of the IFAC World Congress*, 1984.

Caccavale F., Giglio G., Muscio G., and Pierri F. Cooperative impedance control for multiple uavs with a robotic arm. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2366–2371, 2015.

Chen Y., Bruschetta M., Cuccato D., and Beghi A. An adaptive partial sensitivity updating scheme for fast nonlinear model predictive control. *IEEE Trans. on Automatic Control*, 2018.

Chen Y., Bruschetta M., Picotti E., and Beghi A. Matmpc-a matlab based toolbox for real-time nonlinear model predictive control. In *European Control Conf. 2019, Accepted*, 2019.

Chen Y., Cuccato D., Bruschetta M., and Beghi A. An inexact sensitivity updating scheme for fast nonlinear model predictive control based on a curvature-like measure of nonlinearity. In *2017 IEEE 56th CDC*, pages 4382–4387. IEEE, 2017.

Conti R., Meli E., Ridolfi A., and Allotta B. An innovative decentralized strategy for i-auvs cooperative manipulation tasks. *Robotics and autonomous systems*, 72: 261–276, 2015.

Diehl M., Bock H. G., Schlöder J. P., Findeisen R., Nagy Z., and Allgöwer F. Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. *Jour. of Process Control*, 12(4):577–585, 2002.

Farivarnejad H. and Berman S. Stability and convergence analysis of a decentralized proportional-integral control strategy for collective transport. In *2018 Annual American Control Conference (ACC)*, pages 2794–2801, June 2018a.

Farivarnejad H. and Berman S. Stability and convergence analysis of a decentralized proportional-integral control strategy for collective transport. In *2018 Annual American Control Conference (ACC)*, pages 2794–2801. IEEE, 2018b.

Ferreau H., Kirches C., Potschka A., Bock H., and Diehl M. qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, 6(4):327–363, 2014.

Franchi A. and Giordano P. R. Online leader selection for improved collective tracking and formation maintenance. *IEEE Transactions on Control of Network Systems*, 5(1):3–13, 2018.

Franchi A. and Mallet A. Adaptive closed-loop speed control of bldc motors with applications to multi-rotor aerial vehicles. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5203–5208, 2017.

Franchi A., Petitti A., and Rizzo A. Distributed estimation of state and parameters in multi-agent cooperative load manipulation. *IEEE Trans. on Control of Network Systems*, 6(2):690–701, 2019.

Franchi A., Masone C., Grabe V., Ryll M., Bülthoff H. H., and Giordano P. R. Modeling and control of uav bearing formations with bilateral high-level steering. *The International Journal of Robotics Research*, 31(12):1504–1525, 2012. URL https://doi.org/10.1177/0278364912462493.

Gabellieri C., Tognon M., Sanalitro D., Palottino L., and Franchi A. A study on force-based collaboration in swarms. *Swarm Intelligence*, 14:57–82, 2020.

Heemels W. P. M. H., Johansson K. H., and Tabuada P. An introduction to event-triggered and self-triggered control. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 3270–3285, 2012.

Kendoul F. Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems. *Journal of Field Robotics*, 29(2):315–378, 2012. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.20414.

Khalil H. *Nonlinear Systems*. Pearson Education. Prentice Hall, 2002. ISBN 9780130673893.

Koenig N. and Howard A. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 3, pages 2149–2154 vol.3, 2004.

Kumar V. and Michael N. Opportunities and challenges with autonomous micro aerial vehicles. *The International Journal of Robotics Research*, 31(11):1279–1291, 2012. URL https://doi.org/10.1177/0278364912455954.

Lim Y., Kwon S., Kim K., and Ahn H. Implementation of load transportation using multiple quadcopters. In *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 639–644, July 2017.

Lupashin S. and D'Andrea R. Stabilization of a flying vehicle on a taut tether using inertial sensing. *2013 IEEE/RSJ International Conf. on Intelligent Robots and Systems*, pages 2432–2438, 2013.

Manubens M., Devaurs D., Ros L., and Cortés J. Motion planning for 6-D manipulation with aerial towed-cable systems. In *2013 Robotics: Science and Systems*, Berlin, Germany, May 2013.

Marino A. Distributed adaptive control of networked cooperative mobile manipulators. *IEEE Transactions on Control Systems Technology*, 26(5):1646–1660, 2018.

Marino A. and Pierri F. A two stage approach for distributed cooperative manipulation of an unknown object without explicit communication and unknown number of robots. *Robotics and Autonomous Systems*, 103:122 – 133, 2018. ISSN 0921-8890. URL http://www.sciencedirect.com/science/article/pii/S0921889017307807.

Masone C., Bülthoff H. H., and Stegagno P. Cooperative transportation of a payload using quadrotors: A reconfigurable cable-driven parallel robot. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1623–1630, Oct 2016.

Maza I., Kondak K., Bernard M., and Ollero A. Multi-UAV cooperation and control for load transportation and deployment. *Journal of Intelligent & Robotics Systems*, 57(1-4):417–449, 2010a.

Maza I., Kondak K., Bernard M., and Ollero A. Multi-uav cooperation and control for load transportation and deployment. *J. Intell. Robotics Syst.*, 57(1-4): 417–449, January 2010b. ISSN 0921-0296.

Maza I., Kondak K., Bernard M., and Ollero A. Multi-uav cooperation and control for load transportation and deployment. *Journal of Intelligent & Robotic Systems*, 57(1-4):417, 2010c. URL https://app.dimensions.ai/details/publication/pub.1012942262.

Mellinger D., Shomin M., Michael N., and Kumar V. Cooperative grasping and transport using multiple quadrotors. *DARS*, pages 545–558, 2010.

Mellinger D., Shomin M., Michael N., and Kumar V. Cooperative grasping and transport using multiple quadrotors. In *Distributed autonomous robotic systems*, pages 545–558. Springer, 2013.

Michael N., Fink J., and Kumar V. Cooperative manipulation and transportation with aerial robots. *Autonomous Robots*, 30(1):73–86, Jan 2011. ISSN 1573-7527. URL https://doi.org/10.1007/s10514-010-9205-0.

Muttin F. Umbilical deployment modeling for tethered uav detecting oil pollution from ship. *Applied Ocean Research*, 33(4):332 – 343, 2011. ISSN 0141-1187.

Oh S.-R., Pathak K., Agrawal S. K., Pota H. R., and Garratt M. Approaches for a tether-guided landing of an autonomous helicopter. *IEEE Trans. on Robotics*, 22 (3), June 2006. ISSN 1552-3098.

Petitti A., Franchi A., Di Paola D., and Rizzo A. Decentralized motion control for cooperative manipulation with a team of networked mobile manipulators. In *IEEE Int. Conf. on Robotics and Automation*, pages 441–446, Stockholm, Sweden, May 2016.

Pinkney M. F. J., Hampel D., and DiPierro S. Unmanned aerial vehicle (uav) communications relay. In *Conf. on Military Communications, IEEE*, volume 1, pages 47–51 vol.1, 1996.

Ritz R. and D'Andrea R. Carrying a flexible payload with multiple flying vehicles. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3465–3471, 2013.

Rossi E., Bruschetta M., Carli R., Chen Y., and Farina M. Online nonlinear model predictive control for tethered uavs to perform a safe and constrained maneuver. In *2019 18th European Control Conference (ECC)*, pages 3996–4001, 2019.

Rossi E., Tognon M., Carli R., Cortés L. J., and Franchi A. Discrete-time control of parallel kinematic systems. Technical report, University of Padova, March 2019. URL http://automatica.dei.unipd.it/tl_files/utenti2/rossi/tech_report.pdf.

Sanalitro D., Savino H. J., Tognon M., Cortés J., and Franchi A. Full-pose manipulation control of a cable-suspended load with multiple uavs under uncertainties. *IEEE Robotics and Automation Letters*, 5:2185–2191, 01/2020 2020. ISSN 2377-3766.

Sandino L. A., Bejar M., Kondak K., and Ollero A. Advances in modeling and control of tethered unmanned helicopters to enhance hovering performance. *Jour. of Intelligent & Robotic Systems*, 73(1):3–18, Jan 2014. ISSN 1573-0409.

Sanz P. Robotics: Modeling, planning, and control (siciliano, b. et al; 2009) [on the shelf]. *Robotics & Automation Magazine, IEEE*, 16:101–101, 12 2009.

Sieber D. and Hirche S. Human-guided multirobot cooperative manipulation. *IEEE Transactions on Control Systems Technology*, 27(4):1492–1509, 2019.

Sreenath K. and Kumar V. Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots. In *Robotics: Science and Systems*, Berlin, Germany, June 2013a.

Sreenath K. and Kumar V. Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots. 06 2013b.

Tagliabue A., Kamel M., Siegwart R., and Nieto J. Robust collaborative object transportation using multiple mavs. *The International Journal of Robotics Research*, 38(9):1020–1044, 2019. URL https://doi.org/10.1177/0278364919854131.

Tagliabue A., Kamel M., Siegwart R., and Nieto J. I. Robust collaborative object transportation using multiple mavs. *CoRR*, abs/1711.08753, 2017.

Tognon M. and Franchi A. Nonlinear observer-based tracking control of link stress and elevation for a tethered aerial robot using inertial-only measurements. In *2015 IEEE International Conf. on Robotics and Automation*, pages 3994–3999, May 2015.

Tognon M., Gabellieri C., Pallottino L., and Franchi A. Aerial co-manipulation with cables: The role of internal force for equilibria, stability, and passivity. *IEEE Robotics and Automation Letters, Special Issue on Aerial Manipulation*, 3(3):2577 – 2583, 2018.

Tognon M., Testa A., Rossi E., and Franchi A. Takeoff and landing on slopes via inclined hovering with a tethered aerial robot. In *2016 IEEE/RSJ International Conf. on Intelligent Robots and Systems*, pages 1702–1707, 2016a.

Tognon M., Dash S. S., and Franchi A. Observer-based control of position and tension for an aerial robot tethered to a moving platform. *IEEE Robotics and Automation Letters*, 1:732–737, 2016b.

Tognon M., Testa A., Rossi E., and Franchi A. Takeoff and landing on slopes via inclined hovering with a tethered aerial robot. In *2016 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1702–1707, Daejeon, South Korea, 10/2016 2016c.

Tsiamis A., Verginis C. K., Bechlioulis C. P., and Kyriakopoulos K. J. Cooperative manipulation exploiting only implicit communication. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 864–869, Sep. 2015.

Tsiamis A., Verginis C. K., Bechlioulis C. P., and Kyriakopoulos K. J. Cooperative manipulation exploiting only implicit communication. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 864–869. IEEE, 2015.

Verginis C. K., Nikou A., and Dimarogonas D. V. Communication-based decentralized cooperative object transportation using nonlinear model predictive control. In *2018 European Control Conference (ECC)*, pages 733–738. IEEE, 2018.

Wang Z. and Schwager M. Force-amplifying n-robot transport system (force-ants) for cooperative planar manipulation without communication. *The International Journal of Robotics Research*, 35(13):1564–1586, 2016a.

Wang Z. and Schwager M. Multi-robot manipulation without communication. In **Chong N.-Y. and Cho Y.-J.**, editors, *Distributed Autonomous Robotic Systems*, pages 135–149, Tokyo, 2016b. Springer Japan. ISBN 978-4-431-55879-8.

Yang H. and Lee D. Hierarchical cooperative control framework of multiple quadrotor-manipulator systems. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4656–4662, 2015.

Yutao C. Matmpc. https://github.com/chenyutao36/MATMPC, 2017.