Optimization-based Estimation and Control Algorithms for Quadcopter Applications

Optimization-based Estimation and Control Algorithms for Quadcopter Applications

Alex Andriën

Optimization-based Estimation and Control Algorithms for Quadcopter Applications

Alex Andriën

The author has successfully completed the educational program of the Graduate School of the Dutch Institute of Systems and Control (DISC).

The work described in this thesis was carried out at the Eindhoven University of Technology (TU/e) in collaboration with Avular, an Eindhoven-based mobile robotics company. The work is part of the research project "Drone Safety Cluster", which is made possible by the European Union through the European Regional Development Fund (ERDF) as part of the OPZuid Program.

A catalogue record is available from the Eindhoven University of Technology Library.
ISBN: 978-90-386-5428-7

Reproduction: Ipskamp Printing, Enschede, the Netherlands

This thesis has been printed on recycled paper.

Optimization-based Estimation and Control Algorithms for Quadcopter Applications

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de rector magnificus prof.dr.ir. F.P.T. Baaijens, voor een commissie aangewezen door het College voor Promoties, in het openbaar te verdedigen op maandag 24 januari 2022 om 16.00 uur

door

Alex Rudolf Petrus Andrien

geboren te Heeze

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	prof.dr.ir. M. Steinbuch
promotor:	prof.dr.ir. W.P.M.H. Heemels
copromotor:	dr. D.J. Guerreiro Tomé Antunes
leden:	prof.dr.ir. N. van de Wouw
	prof.dr.ir. B. Jayawardhana (Rijksuniversiteit Groningen)
	dr. B. Houska (ShanghaiTech University)
	dr. I. Palunko (University of Dubrovnik)

Het onderzoek dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

DUM SPIRO SPERO

Contents

I Introduction

Part I

Introduction

Chapter 1

Introduction

In this chapter the contents of the thesis are introduced, put into historical perspective and connected to recent literature. First, a brief history of human aviation is provided, starting in antiquity and ending with the relatively recent introduction of quadcopters. Next, the components making up a quadcopter are explained, covering actuation, sensing, estimation, planning and control. The last three of these topics are discussed in more detail, since they are related to the main topics covered in the thesis. Thereafter, the contributions of the thesis are summarized, followed by an overview of the structure of the remainder of the thesis. The chapter concludes with a list of publications of the author.

1.1 A brief history of human aviation

I

The history of (human) aviation goes back thousands of years, starting with gliding-based devices such as kites that appeared as early as the 5th century BC [49]. Stories of tower jumping, where people would strap birdlike wings to themselves and would attempt to fly by jumping of a tower, have been around since antiquity with the Greek legend of Icarus [78] as one of the best known examples. It would not be until the 18th century that prolonged human flight was possible with the first successful hot-air balloon flight in 1783 by Jean-François Pilâtre de Rozier and François d'Arlandes [110]. The achievement of the first controlled, sustained flight of a heavier-than-air aircraft was achieved 120 years later by the famous flight of the Wright brothers on December 17th 1903 in Kitty Hawk, North-Carolina, USA [91]. From this short flight of just 12 seconds, powered aviation progressed quickly with series production of aircraft being achieved by 1907, when the Demoiselle No. 19 monoplane became the first heavier-than-air aircraft produced in series [84].

(a) Daedalus and Icarus, relief in the Villa Albani, Rome[1].

(b) The famous flight of the Wright brothers[2].

Figure 1.1: Illustrations of the long history of human aviation

[1]Image retrieved from https://en.wikipedia.org/wiki/File:Daedalus_und_Ikarus_MK1888.png, taken from page 409 of 1028, in Volume 4 of the German illustrated encyclopedia Meyers Konversationslexikon, 4th edition (1885-1890).

[2]Image taken from Library of Congress Prints and Photographs Division Washington, D.C. 20540 USA, http://loc.gov/pictures/resource/ppprs.00626/

1.1.1 Rotorcraft

Around the same time as the first airplanes were produced in series, the first human flight with a rotor wing aircraft was achieved in 1907 by the French brothers Jacques en Louis Breguet [155]. The aircraft they used was the Gyroplane No. 1, which might very well be the earliest example of a quadcopter. It consisted of a central frame from which four arms extended, each of them having a four blade biplane rotor at the end (see Figure 1.2a). Although this aircraft was able to lift a human of the ground (be it only 0.6 meters at the first flight, which later extended to 1.5 meters), it required a person at the end of each arm to stabilize the device. Paul Cornu performed the first free flight in the same year with the Cornu helicopter [155], which consisted of two counter-rotating rotors (see Figure 1.2b). Many other designs arose in subsequent years. However, it would take until the early 1920s for the first 'practical' rotorcraft to arrive with the invention of the autogyro by Juan de la Cierva [100] (see Figure 1.2c). Practical is meant here in the sense that it was able to fly distances that made it useful for traveling: it flew from London to Paris in 1928, crossing the English Channel. In the subsequent decades many different designs were developed, such as coaxial multirotors, twin rotors, tiltrotors, intermeshing rotors and tip-jets, see Figure 1.3. Although all these designs are/were capable of successful flights, at least to some extent, the most common configuration in use today is that of a single main rotor with a tail rotor to compensate the rotation caused by the main rotor, known commonly as the helicopter.

1.1.2 Modern Quadcopters: A revolution in the skies

Despite the growth of the airline industry in the second half of the 20th century, airplanes and rotorcraft were still very expensive devices and private use was only

[3] Image taken from http://www.aviastar.org/helicopters_eng/breguet_gyro.php.

[4] Image retrieved from https://commons.wikimedia.org/wiki/File:HE2G8.jpg, originally taken from http://www.centennialofflight.gov/essay/Rotary/early_20th_century/HE2G8.htm.

[5] Image taken from https://flickr.com/photos/49487266@N07/6950759784, from the San Diego Air and Space Museum Archive repository.

[6] Cropped image, original from NASA/JPL-Caltech, retrieved from https://mars.nasa.gov/mars2020/multimedia/raw-images/SI1_0046_0671022109_238ECM_N0031416SRLC07021_000085J

[7] Retrieved from http://www.aviastar.org/helicopters_eng/fw-61.php

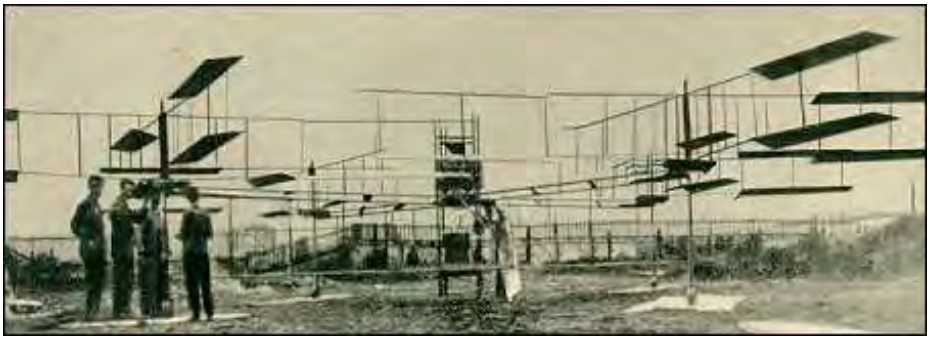
[8] Image by Dutch Ministry of Defense, https://www.defensie.nl/onderwerpen/materieel/vliegtuigen-en-helikopters/boeing-ch-47d-en-ch-47f-chinook-transporthelikopter

[9] Photo taken by Michael Starkey, retrieved from https://en.wikipedia.org/wiki/File:Iwo_Jima_Osprey.jpg

[10] Photo taken by Greg Goebel, retrieved from https://commons.wikimedia.org/wiki/File:Yakmx_2b_(29167519213).jpg

[11] Photo taken by Wikipedia User: JimCollaborator (https://en.wikipedia.org/wiki/User:JimCollaborator), retrieved from https://commons.wikimedia.org/wiki/File:Hiller_YH-32_Hornet.jpg

I



(a) *Gyroplane No. 1*, possibly the first quadcopter[3].

(b) *Cornu helicopter*, the first rotorcraft capable of free human flight[4].

(c) *Cierva C.8 Autogyro*, the first rotating wing aircraft to cross the English Channel[5].

Figure 1.2: Early 20$^{\text{th}}$ century rotorcraft designs.

(a) *Ingenuity*, currently on Mars[6].

(b) *Focke-Wulf Fw 61*[7].

(c) *Boeing CH-47 Chinook*[8].

(d) *Bell Boeing V-22 Osprey*[9].

(e) *Kaman K-Max*[10].

(f) *Hiller YH-32 Hornet.*[11].

Figure 1.3: Examples of different rotor designs: (a) coaxial , (b) transverse, (c) tandem, (d) transverse mounted tiltrotor, (e) intermeshing and (f) tip-jet multirotors.

I

accessible to the very rich. Moreover, they could only be operated by highly trained pilots. This started to change with the introduction of quadcopters around the year 2006, when the Federal Aviation Administration (FAA) issued its first commercial drone permit and companies started developing drones for commercial use. In 2010 Parrot introduced the Parrot AR drone [117, 32], which was a quadcopter type drone that was controllable via a smartphone. This drone opened the market to consumers, in fact, this introduction could be seen as the first time in history that controlled flight was accessible to the general public. When DJI introduced the quadcopter type Phantom 1 in 2013 [48], which allowed high quality aerial photos and videos to be taken by everyone, the commercial drone market saw a massive growth, reaching over $ 21 billion by 2021 [168]. Today, quadcopters are used in a wide variety of applications, including aerial photography, agricultural and industrial inspection, (parcel) delivery, research and artworks [127].

(a) *DJI Phantom*, one of the first quadcopters equipped with a camera and gimbal[12]

(b) *Avular Vertex*, a customizable quadcoper used in research and inspection[13]

(c) *Avular Vertex*, equipped with a microphone array to detect defects in industrial installations[12]

(d) DHL delivery drone used in China[14]

Figure 1.4: Examples of quadcopter applications

1.2 Quadcopter working principles

Modern quadcopters consist of four separate rotors mounted on a rigid frame, where each rotor is mounted at the end of an arm, see Figure 1.4. They are generally considered a type of robot, which becomes clear when the definition of robot from [43] is considered: *"a goal oriented machine that can sense, plan and act"*. Like most robotic systems, quadcopters consist of several components required to operate, which can be divided as follows:

1. *Sensing*: The gathering of information from the environment (data collection).

2. *Estimation*: The interpretation of the sensor information (turning the data into useful information).

3. *Planning*: The computation of the desired motion based on user specifications (planning how to arrive at a goal given the current information).

4. *Control*: The determination of the required actuation for a desired motion (turning the planned motion into actuator setpoints).

5. *Actuation*: The implementation of the actuation in the physical environment (achieving motion).

In this division, the first and last items, i.e., sensing and actuation, provide the interaction with the environment and are part of the hardware of the system. The other components handle the processing of information from the environment (estimation) and decide what to do (planning) and how to do it (control) to achieve a desired goal. These components are implemented in software. An overview of the components is given in Figure 1.5, together with the interaction between them.

In the remainder of this section the actuation and sensing components for quadcopters will be discussed in more detail, followed by separate sections on estimation, planning and control, since these are the main topics covered in this thesis.

1.2.1 Actuation

In quadcopters, motion is made possible by the lift generated by the four propellers, where two propellers opposite to each other rotate in the same direction

[12] Image by DJI, retrieved from https://www.dji.com/nl/phantom

[13] Image by Avular (https://avular.com/), used with permission

[14] Image by DHL, retrieved from https://www.dhl.com/tw-en/home/press/press-archive/2019/dhl-express-launches-its-first-regular-fully-automated-and-intelligent-urban-drone-delivery-service.html

I

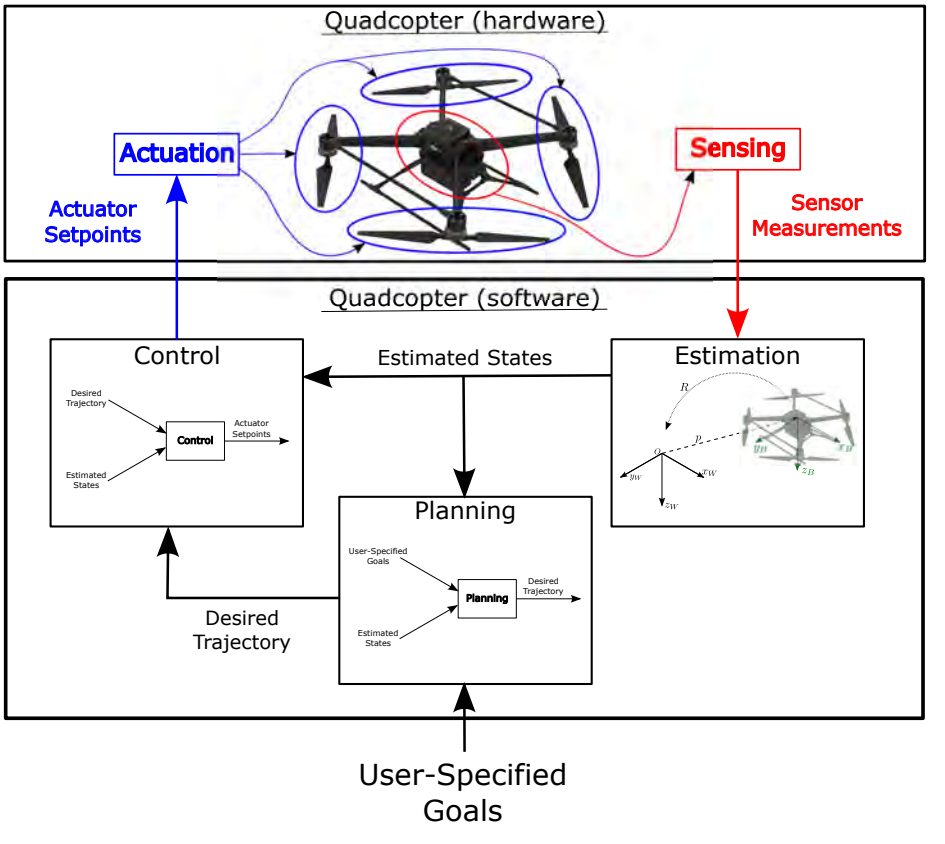


Figure 1.5: Schematic overview of the different components that make up a quadcopter. The sensors send measurements to the estimation component, which turns the measurements into useful information. This information is then used in the planning component, together with user-specified goals, to determine the trajectory the quadcopter should follow. The desired trajectory is combined with the estimated states in the control component that computes the required actuator setpoints. Finally, the actuators implement these setpoints in the physical environment.

(e.g., clockwise), whilst the other pair rotates in the reverse direction (e.g., counterclockwise), see Figure 1.6. Each propeller's rotational velocity, which is proportional to the generated thrust of the propeller, can be controlled individually. This allows for control of the quadcopter, which can be conceptually understood by considering the following scenarios:

1. Having all the propellers produce equal thrust results in the quadcopter experiencing a thrust upwards (Figure 1.6a). In hover, this thrust experienced by the quadcopter equals the gravitational force acting on the quadcopter. By increasing the thrust above this hover level, the quadcopter moves upwards, while decreasing the thrust below the hover level will make the quadcopter move downward.

2. By increasing the thrust of two adjacent propellers and decreasing the thrust of the other two by the same amount, a rotation around an axis can be achieved, as shown in Figures 1.6b and 1.6c.

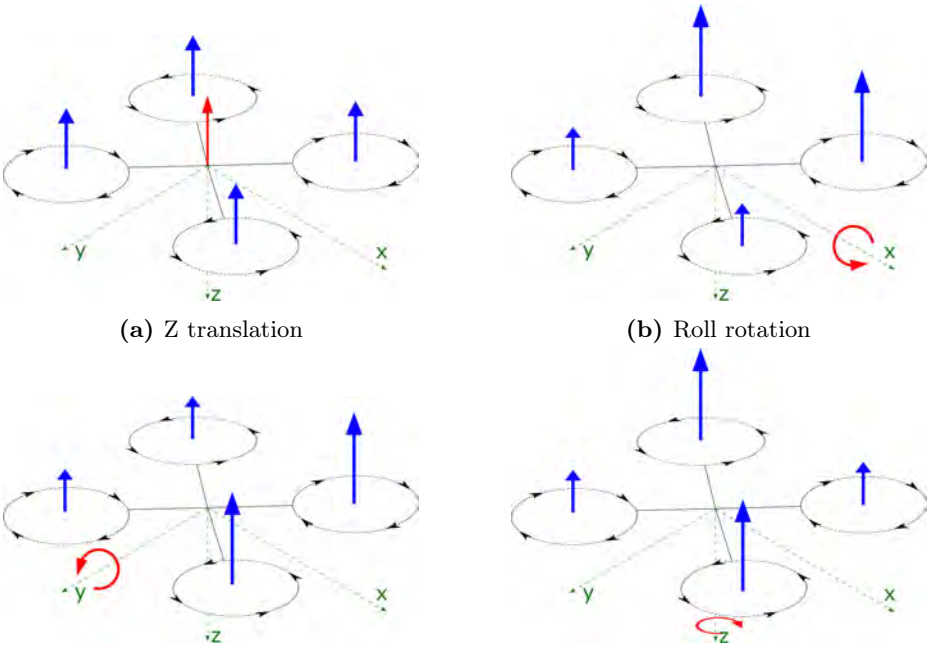
3. Increasing the rotation of the clockwise propellers with respect to the counterclockwise propellers results in a rotation about what is referred to as the yaw axis (Figure 1.6d).

From these scenarios it becomes clear that in order to move a quadcopter in the horizontal plane, the inputs cannot be used independently. Moreover, there are only four inputs, while there are six degrees of freedom (DOFs): three rotational and three translational DOFs. This makes the quadcopter an underactuated system, i.e., the inputs cannot be used to accelerate the quadcopter in arbitrary directions. In particular, horizontal movement can only be achieved from a rotation around the roll or pitch axes.

1.2.2 Sensing

In order to perform a task with a quadcopter (or robot in general), first its position and orientation (or attitude) with respect to its surroundings need to be determined. The position and orientation, as well as their derivatives such as velocity and acceleration, are commonly grouped into what is referred to as the *state* of the quadcopter. In order for the current state of a quadcopter to be determined, it is equipped with a variety of sensors. These sensors provide measurements of one or several quantities, which can then be used to determine the state of the quadcopter using estimation algorithms (see Section 1.3). The types of sensors that a quadcopter is equipped with vary per model, and each type of sensor is available in a range of configurations by different manufacturers. However, a rough division of common sensor types can be made as follows.

I



(a) Z translation **(b)** Roll rotation

(c) Pitch rotation **(d)** Yaw rotation

Figure 1.6: Movement scenario's for a quadcopter. In each sub-figure the disks represent the propellers with corresponding direction of rotation and the blue arrows represent the direction and magnitude of thrust provided by each propeller, where the magnitude is visualized by the size of the arrow. A body-fixed frame according to the North-East-Down convention is shown in dashed green and the red arrows represent the acceleration on the body-fixed frame as a result of the propeller thrusts.

1.2.2.1 Onboard Sensors

Onboard sensors, as the name suggests, are sensors that are place on the quadcopter itself. Most of the sensors used in quadcopters belong to this category and because they are placed on the quadcopter, they are often designed to be lightweight and energy-efficient.

- Accelerometer
  Accelerometers are used to measure the accelerations of the quadcopter in three directions. They are very common in quadcopters and are used in both attitude and position estimation. Accelerometers are often part of the Inertial Measurement Unit (IMU).

- Gyroscope
  Gyroscopes measure angular velocities in three directions. They are very common in quadcopters and are used in attitude estimation. Together with the accelerometer (and sometimes magnetometer) they are part of the Inertial Measurement Unit (IMU).

- Magnetometer
  Magnetometers provide measurements of the surrounding magnetic field. They can be used to determine the attitude of the quadcopter by using it as a compass of sorts. In particular, they are often key in determining the angle about the gravitational vector when the quadcopter is hovering.

- Camera
  Cameras are used as a sensor by using vision processing, detecting distance to objects among other things. By using several cameras pointing in the same direction, 3D images can be obtained and distance can also be determined.

- Optical flow sensor
  Optical flow sensors measure the visual motion and output a displacement measurement, which is useful for determining horizontal velocities and in obstacle avoidance.

- Barometer
  Barometers measure the pressure in the environment, which can be used to determine the height of the quadcopter by considering the barometric formula [124].

- LiDAR (Light detection and ranging) sensor
  LiDAR sensors are used to measure distances to objects. They are often used for height estimation and obstacle avoidance, but are also used in combination with localization algorithms such as SLAM (Simultaneous localization and mapping) [34].

1.2.2.2 External Sensors

External sensors are sensors that are not (fully) placed on the quadcopter itself. Some of them provide sensor information directly, while others are used in conjunction with equipment on the quadcopter itself.

- Global Navigation Satellite Systems (GNSSs)
  A Global Navigation Satellite System can be used to determine position of the quadcopter outdoors. It uses satellite position system, such as the well-known GPS, Galileo and GLONASS, to determine the position of the quadcopter relative to these satellites. By using the known position of

these satellites, the position of the quadcopter with respect to the world can be obtained.

I

- Ultra-wideband (UWB) system
  Contrarily to GNSSs, UWB systems are mostly suitable for indoor use. However, they use a method similar to that of GNSSs for determining the position of a quadcopter, by which beacons are positioned in a room as 'satellites' that transmit ultra-wide frequency signals. These can then be used by a receiver to determine the position of the quadcopter.

- Motion Capture (mocap) Systems
  Mocap systems consist of multiple cameras positioned in a room, which track markers (typically using infrared reflection). By using multiple markers on a quadcopter, the attitude can also be determined by considering the position of each marker relative to each other. They provide very high accuracy position and attitude estimates and are often used as 'ground truth' estimates to validate other sensors.

Armed with these sensors, quadcopters are capable of sensing the environment around them. However, the measurements provided by these sensors often do not provide the desired accuracy and/or they do not measure the state that is of interest directly.

1.3 Estimation

Although the quadcopter is equipped with several sensors, they often do not provide direct measurements of the desired states, or they do not provide the desired accuracy. In order to determine the state of the quadcopter, estimation algorithms are used, which, by combining different sensors' information and/or using model information, provide an (improved) estimate of the actual state of the quadcopter.

In order to discuss the orientation of a quadcopter (or any rigid-body for that matter) coordinate frames are used. The world frame, also referred to as the inertial frame, is fixed to the environment, providing a reference. The world frame is represented by three orthogonal unit vectors. Another frame is attached to the quadcopter, known as the body-fixed frame, again specified by three orthogonal unit vectors, see Figure 1.7. The orientation of this body-fixed frame, or equivalently its vectors, with respect to the world frame specifies the orientation of the quadcopter, which is also commonly referred to as the attitude of the quadcopter. This orientation can be represented in a variety of ways, with some examples discussed in the 'Representing Orientation' insert below. The position of the origin of this frame is used to describe the position of the quadcopter.

Estimation for quadcopters is typically divided into two parts: 1) attitude and angular velocity estimation; and 2) position and translational velocity estimation. Both categories will be treated separately. At the end of this section, methods that estimate the pose, i.e., the combination of position and orientation, directly will be discussed.

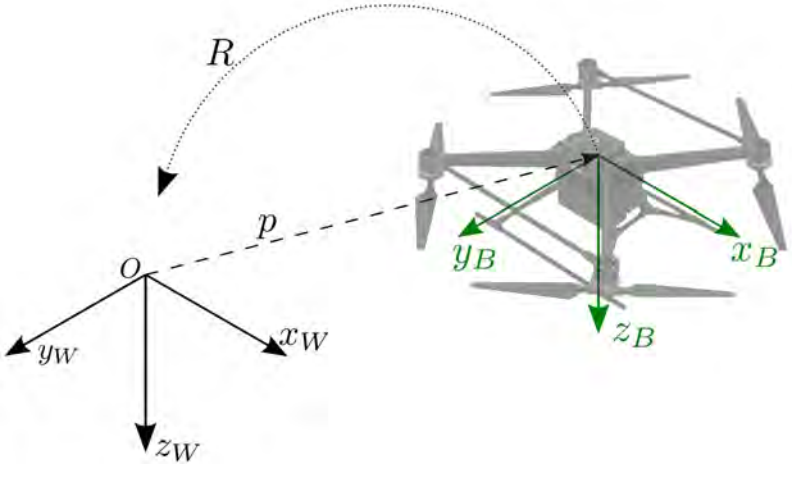


Figure 1.7: Coordinate frames. The world frame and its axes are depicted in black, whereas the body-fixed frame and its axes are shown in green. The position of the body-fixed frame, and thus that of the quadcopter, with respect to the world frame is shown with the dashed arrow and denoted as p. The orientation of the body-fixed frame with respect to the world frame is denoted by R and conceptually visualized by the dotted line.

Representing Orientation

Expressing the orientation of a quadcopter is equivalent to expressing the rotation in three dimensions of a frame attached to the quadcopter with respect to a world-fixed frame. A rotation in three dimensions can be represented in several different ways, here the ones most commonly used in quadcopters are briefly discussed.

Rotation Matrix: The mentioned rotation matrix contains the vectors of the body-fixed frame specified in the world frame as each of its columns. This results in a 3×3 matrix, having 9 parameters. Rotating a vector by using a rotation matrix is simply multiplying this vector by this rotation matrix and successive rotations can be achieved by multiplication of the corresponding rotation matrices. The advantage of representing the orientation in this manner is that it fully defines the rotation, while the disadvantage is that is contains 9 parameters, making it redundant since there are only three degrees of freedom.

Euler Angles: The orientation can also be represented by applying three elementary rotations around the axes of moving frames, starting at the body-fixed frame and ending at the world-fixed frame. In quadcopters (and aviation in general) it is common to use the Tait-Bryan $z - y - x$ rotation and the corresponding yaw (ψ), pitch (θ) and roll (ϕ) angles. That is, the rotation is represented by: (i) a fist rotation about the z-axis of the body-fixed frame by ψ; (ii) a second rotation around the y-axis of the resulting frame by θ; and (iii) a final rotation around the x-axis of the frame resulting from the first two rotations by ϕ. Euler angle representations are often used, because they use only three parameters and they allow for an easier interpretation of a rotation. The disadvantage is that they suffer from what is known as 'gimbal lock', which occurs when two of the mentioned rotation axes align, making the representation not unique.

Quaternions: According to Euler's rotation theorem, any rotation of a rigid body in a three-dimensional space around a fixed point can be represented by a single rotation around a fixed axis going through the fixed point. Unit quaternions can be viewed as a method to represent the axis and angle by a 4×1 vector. Using quaternions, subsequent rotations can be easily computed using the quaternion product and it has the advantage of being a more compact representation than a rotation matrix. Moreover, quaternions do not suffer from the gimbal lock problem as Euler angle representations do, making them a preferred choice in many applications that involve three-dimensional rotations. However, care must be taken, since the quaternion representation forms a double cover of the group of rotations in three-dimensions, i.e., a quaternion and its negative represent the same orientation.

1.3.1 Attitude and angular velocity estimation

Attitude estimation for quadcopters is typically carried out using the sensors of the IMU, i.e., the accelerometer, gyroscope and magnetometer. At low accelerations, the accelerometer provides a measure of the direction of gravity in the body-fixed frame. The magnetometer measures the ambient magnetic field, which, combined with knowledge of the Earth's magnetic field at the current position and some calibration procedures [112][1], provides a vector measurement of the direction of magnetic north. Together, the accelerometer and magnetometer provide two independent vector measurements[2], which are sufficient to characterize the attitude of a quadcopter. The angular velocities measured by the gyroscope can be integrated over time to provide another estimate of the orientation. The reason for using both vector-based and gyroscope-based measurements is that they both have their disadvantages that can (partially) be compensated for by the other. Indeed, the attitude measurement of the accelerometer is disturbed when the drone is experiencing exogenous forces other than gravity, while the magnetic field measured by the magnetometer is influenced by ferromagnetic elements in the environment[3] and the attitude estimate of the gyroscope typically has drift caused by noise and bias in the original measurement that is integrated over time.

IMU-based attitude estimation has been researched extensively over the past decades [3, 157]. The most common algorithms used are either complementary filter (CF)-based [137] or Kalman filter (KF)-based solutions [32, 135], although alternative solutions have also been proposed, such as sliding-mode [62], fuzzy-logic [161], gradient-descent [135] and neural network [207] based methods.

1.3.1.1 Complementary filter-based

Euler angle based linear complementary filters have been developed in [13, 141]. Complementary filters, which operate directly on SO(3), and can, therefore, be considered nonlinear, have been proposed by Robert Mahony, Tarek Hamel and Jean-Michel Pflimlin in a popular sequence of papers [136, 82, 137]. In these papers they developed several nonlinear complementary filters and provided extensive stability proofs for all of them. Others have continued their work, resulting in adaptive complementary filters (ACFs), where the noise characteristics of each sensor is updated in the estimation procedure, based on some adaption mechanism. This contrasts static filters, i.e., filters where the sensor characteristics are considered constant. ACFs have been proposed that use quaternion [198, 215] and rotation matrix [98] representations.

[1] A very good reference on inertial sensor based estimation is the PhD thesis of Manon Kok [111].

[2] Assuming that the quadcopter is not flown too close to the magnetic North or South poles.

[3] Such as the rebar (steel mesh) used in concrete, as experienced on the robotic's soccer field at the Eindhoven University of Technology (https://www.techunited.nl/en/).

1.3.1.2 Kalman filter-based

The standard KF [104] has been applied to IMU-based attitude estimation in [20], mainly using Euler angles. Nonlinear extensions of the KF have also been used, mainly the extended KF (EKF) [103] using both quaternion [197, 175] and SO(3) [90] representations and unscented KF (UKF), also using both quaternion [53] and SO(3) [185] representations. Also Bayesian filters have been used, such as the particle filter [156], Fisher information-based filters [121] and cubature KFs [211]. Clearly, there is a broad literature on Kalman filters and their variants in this context.

1.3.2 Position and translational velocity estimation

Estimating the position of a quadcopter is often considered more difficult than that of estimating its attitude. The problem is somewhat simpler when external sensors, such as GNSSs, UWB or mocap systems are available, since they provide high quality estimates of the position. However, they have their own challenges, such as latency in providing data to the quadcopter, but mainly their dependence on the environment, i.e., they are not available everywhere. For instance, GNSSs is only reliable outdoors and, to get high precision measurements, a base station with known position is needed [162]. Measurements from UWB systems [153, 113, 119] are only available in the neighborhood of the installed anchors, and similar issues hamper mocap systems. Mocap systems have the additional disadvantage of being quite expensive and requiring no obstructions between the cameras and the quadcopter (although some obstructions are allowed if a sufficient number of cameras is used).

If only onboard sensors are available, things get even harder [181], especially when an absolute position is desired. For estimating the horizontal position of the quadcopter, i.e., its x and y position in the world frame, (a combination of) cameras, optical flow sensors [60] and LiDARs is often used. The displacement measured by optical flow sensors can be integrated over time to give a position estimate relative to the starting point. LiDAR sensors, like cameras used in stereo, can be used to measure the distance to objects, from which a relative position estimate can be obtained. Combined with mapping techniques and some knowledge of the environment, this can lead to position estimates in the inertial frame [151]. Cameras can also be used in more advanced algorithms that use object classification and known relations between objects in the environment, sometimes referred to as world modeling, to determine the current position of the quadcopter [44, 21].

Determining the height of a quadcopter is typically done using the accelerometer and barometer [176], sometimes extended with the use of cameras and LiDAR. The acceleration measurement can be integrated twice over time, giving an estimate of the relative position, although it often shows considerable drift due to the noisy characteristics of the accelerometer and the double

integration performed [159]. The pressure measurement of the barometer yields an absolute height measurement through the barometric formula [71], however it is susceptible to changes in pressure not caused by a change in height, which yield disturbed height measurements. Examples of these disturbances include airflow caused by wind or the propulsion of the quadcopter itself as well as temperature drifts. Cameras and LiDAR are used in a similar manner as for horizontal position estimation [40, 64, 65], although one has to take into consideration the orientation of the quadcopter with respect to the ground, which is estimated using techniques discussed in the previous section. Since the estimated attitude is never perfect, errors in this estimate can propagate into the estimate of the height.

1.3.3 Direct pose estimation

There are also methods that combine the estimation of both the attitude and the position. Mocap systems, for instance, often provide both (if multiple trackers are used). Onboard cameras can also be used to directly estimate the pose by using markers in the environment, such as Aruco markers [70] or the Avular Starling system[1]. Moreover, there is information in each subsystem (attitude and position) that can be used to infer estimates of the other, e.g., a change in horizontal position is the result of a rotation around some of the axes. Methods using this information have been considered that use CFs [201, 80], EKFs [11, 59, 51] and UKFs [185], among others.

1.4 Planning

Once the state of the quadcopter is estimated with sufficient accuracy, the next step is to determine how the quadcopter should move in order to achieve a user-specified goal. This motion is often represented by a *trajectory*, which contains the required states over time that will result in achieving the goal(s), see Figure 1.8. Quadcopters have the property of being differentially flat [199], which means that a certain subset of the states and their derivatives can be used to completely describe the evolution of the other states and inputs. In the case of quadcopters, its position and yaw angle and their derivatives are sufficient to describe the other states and inputs. Therefore, determining a sufficiently smooth (i.e., so that the required derivatives exist) trajectory for the position and yaw angle, in order to achieve the user-specified goals, is the task of the planning component.

Often, the user specifies only the start and end points and it is up to the planning algorithm to determine a trajectory that avoids obstacles and respects the physical constraints of the quadcopter [223]. Other subgoals can also be considered, such as waypoints along a path that specify poses where the quadcopter

[1] https://avular.com/starling/

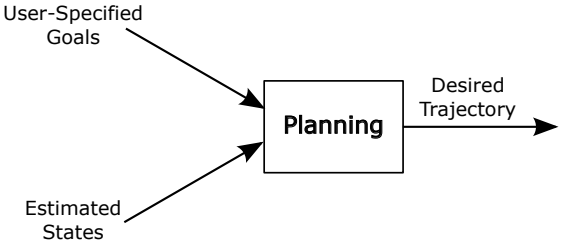


Figure 1.8: Schematic representation of the planning component of a quadcopter. Using the current states estimated in the estimation component a desired trajectory is computed that satisfies user-specified goals.

should stand still, for instance, for images to be shot [172, 178]. Trajectory generation often makes use of the differential flatness property of quadcopters [199] in combination with, for example, piecewise polynomial functions [147] or optimization methods [47]. The task of trajectory generation is sometimes divided into path planning, i.e., finding a parametric function that avoids obstacles, and motion planning, i.e., determining the speed of the quadcopter along the path such that dynamic feasibility constraints are satisfied [47, 31]. Methods used for planning include optimal control [87], receding horizon path planning [26], B-spline optimization [222] and motion primitives [154].

Planning strategies that use optimal control in particular have been studied extensively in quadcopters. For instance, in [95] time-optimal trajectories are computed for landing a quadcopter on a (translating and tilting) moving platform. They consider a 3 (2 translational and 1 rotational) DOF model and use the differential flatness property [199] of the system to convert the control problem to a nonlinear programming problem using time discretization. They allow for collision avoidance and time-varying final constraints and provide experimental results. In [220] a real-time time-optimal trajectory planning approach subject to velocity constraints and non-convex input (thrust) and attitude constraints is proposed. They assume a geometric path is given and attempt to follow it as fast as possible by using the differential flatness property. In [186] a minimum time-optimal control problem with constraints on the roll, pitch and yaw rates and angles, as well as position and thrust constraints, is tackled. They transcribe the problem into transverse coordinates, i.e. a geometric path, and transform the dynamics, cost and constraints into the new coordinates. They then use the PRONTO algorithm [86] with barrier functions for the constraints to obtain optimal trajectories. This allows them to avoid obstacles and they show experimental results using a Crazyflie [73] nano quadcopter. In [96] trajectories are formulated as piecewise polynomials using the differential flatness property, as was done in [147]. They formulate a quadratic program (QP) that generates trajectories through a series of waypoints, while satisfying physical constraints

of the quadcopter and corridor constraints on the position. In [72] the direct multiple shooting approach is used to compute trajectories for quadcopter based systems (such as with an end-effector). In [193] time-optimal descent trajectories that avoid the Vortex Ring State are provided by solving a minimum time-optimal control problem using GPOPS II [167], a general purpose optimal control solver. A performance benchmark of time-optimal control strategies for quadcopters is provided in [89].

In [194] a library of optimal trajectories is calculated offline and used to train a neural network, which then predicts new trajectories online from different initial conditions. The predicted trajectories are further optimized using a sparse QP solver and they demonstrate the results in experiments, where it calculates near-optimal trajectories in a few milliseconds. GPOPS II is also used in [195] to generate a training set of trajectories which are then encoded and generalized using Dynamic Movement Primitives (DMPs). They use a planar quadcopter model and generate new trajectories with the DMPs. In [170] time-optimal maneuvers for a two dimensional quadcopter subject to input constraints are calculated using Pontryagin's Maximum Principle (PMP). Flips are allowed and they use switching time optimization (STO) to solve the boundary value problem (BVP) that results from the PMP. Time-optimal trajectories in 2D are computed in [23] using a decomposition of the trajectory, resulting in analytical solutions. In [191] they use the differential flatness property and a polynomial parametrization for the flat outputs.

The computation of energy efficient trajectories has also received attention from the research community. For instance, in [132] energy-optimal trajectories are computed using GPOPS II by assuming that the quadcopter undergoes three phases: acceleration, constant speed, and deceleration. The energy consumption is then calculated afterwards and compared for different trajectories. In [214] models for the brushless direct current (BLDC) motors and the battery typically used in quadcopters are provided, as well as a calculation of the energy consumed by the quadcopter. They consider deterministic wind disturbances and calculate energy-optimal trajectories using GPOPS II. In [38] they calculate the power usage of the drone by explicitly considering the efficiency of the battery depending on its state-of-charge. [187] compares drone based delivery with truck based delivery and shows the potential of quadcopters to be more efficient in commercial delivery applications, showcasing the importance of energy efficient planning.

1.5 Control

Where *planning* is considered with determining a trajectory for the quadcopter from user-specified goals, *control* is concerned with determining the required actuator setpoints in order to follow the desired trajectory. As is the case with estimation, control of quadcopters is often divided into attitude and position

control [131, 67], which are then combined for full motion control. Such a scheme is typically referred to as *cascaded control*, because of the control loops forming a cascade, see Figure 1.9. In the rest of the section attitude control is discussed first, followed by position control strategies.

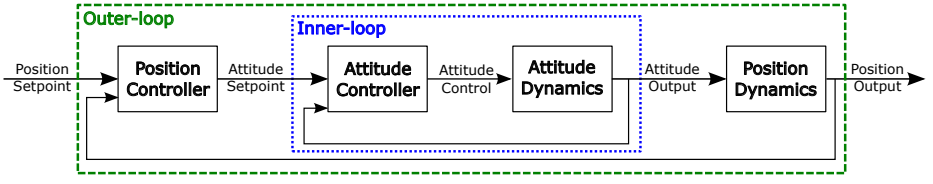


Figure 1.9: Cascaded control scheme often employed in quadcopter control. The *inner-loop* consists of the attitude dynamics and attitude controller in a feedback loop, which ensures attitude setpoints can be tracked. The *outer-loop* uses a position controller that generates setpoints for the inner-loop, these setpoints are such that the position dynamics follow a desired trajectory as determined in the planning component.

1.5.1 Attitude control

As in attitude *estimation*, attitude *control* is either linear, often by using Euler angles as representation, or nonlinear, commonly based on quaternion or rotation matrix representations.

1.5.1.1 Linear attitude control

Linear attitude control is typically achieved by linearization around the hover conditions of the quadcopter or around a trajectory [50]. This often results in the use of relatively simple controllers, such as proportional-integral-derivative (PID) [32, 160] and linear quadratic (LQ) [30] control strategies. Since the attitude kinematics and dynamics of a quadcopter are nonlinear, it is hard to give global guarantees using linear control. However, in practice, linear controllers are quite effective, providing that the trajectories flown do not require large angle deviations from hover. Because of their simplicity and effectiveness, linear controllers are widely used and offer a relatively easy starting point to control the angles of a quadcopter [109].

1.5.1.2 Nonlinear attitude control

Nonlinear attitude control has been extensively studied [208, 33, 179], often involving control on SO(3). One of the difficulties of control on SO(3) is that global asymptotic stability is not possible with continuous state-feedback laws [24], making almost global asymptotic stability the best one can hope to achieve.

Here, 'almost' refers to the requirement that the stability is guaranteed for all initial conditions not belonging to a set of measure zero [149, 123]. Hybrid [142] and discontinuous [122] control laws have been proposed to achieve global asymptotic stability of the attitude dynamics.

1.5.2 Position control

Similar to the attitude control, the position control of quadcopters has received much attention in the past decade [173]. A few examples of control strategies used include sliding-mode control [29], iterative learning control [88], nonlinear control [102, 123, 114], reinforcement learning [97] and model predictive control [22, 152, 79]. For more information and comparisons between control strategies, several survey papers provide adequate references [224, 173, 109, 105].

1.6 Contents and contributions of this thesis

Quadcopters are becoming increasingly important in today's society, and their use in both industry and consumer markets is expected to increase in the future [210, 183]. The increased use of, and new applications for, quadcopters is expected to result in stricter demands on their performance in terms of speed, accuracy, reliability and robustness [42]. In order for these demands to be met, improvement of existing estimation and control algorithms is of crucial importance. In this thesis, several existing methods for estimation and control of quadcopters are examined and improved upon by employing optimization-based techniques. These novel additions to the fields of estimation and control of quadcopters are discussed in more detail below.

1.6.1 Adaptive Complementary Filter

First, the problem of attitude estimation by using IMU measurements is tackled. As discussed in Section 1.3.1, the angular velocity measurements of the gyroscope are often fused with vector-based measurements provided by the accelerometer and/or magnetometer using a complementary filter, because of its stability properties (also on SO(3)) and intuitive tuning. The complementary filter is tuned by setting the cut-off frequency, which determines the relative trust in the gyroscope and vector measurement. This cut-off frequency is typically based on static measurements of the noise characteristics of the gyroscope and vector measurements and set as a constant. However, especially the noise characteristics of the vector measurements are not stationary, due to, for instance, accelerations of the quadcopter that disturb the gravity vector measurement of the accelerometer or magnetic disturbances acting on the magnetometer.

Based on the observation that the noise characteristics of the accelerometer and magnetometer are not stationary, a novel adaptation scheme for the complementary filter is proposed. This adaptation scheme modifies the complementary filter cut-off frequency and is based on the similarity between independent estimates obtained from the vector and gyroscope measurements. The adaptive complementary filter is also derived on the special orthogonal group and convergence of the filter is established. The effectiveness of the approach is demonstrated with simulation results.

Contribution 1.1. *The design of a new, adaptive, nonlinear complementary filter with a novel adaptation rule that provides a more effective way of handling disturbances acting on the vector measurements. The filter error converges asymptotically towards zero.*

1.6.2 Estimation and identification for Markov Jump Linear Systems

The second research direction originated from the problem of intermittent availability and reliability of sensors such as GNSS and UWB systems, as discussed in Section 1.3.2. This problem is reformulated as that of estimating the optimal joint maximum a posteriori probability (JMAP) state and mode of a Markov Jump Linear System (MJLS), for which several methods already existed in the literature. However, computing the JMAP estimates of the state and mode of a MJLS is known to be a computationally intractable problem and the existing methods provide no guarantees in the sense of being close to the optimal, with most methods often getting stuck in local minima. A novel approximate method for such problems is proposed that guarantees to be within a pre-specified bound of the optimal estimate. The proposed method builds upon relaxed dynamic programming [129]. Through numerical examples, it is shown that this method can lead to improved estimates with less computations than existing methods proposed in the literature.

This novel method has also been applied to the problem of identifying the parameters of a class of stochastic switched systems, where the active subsystem is determined by a Markov chain. This class includes autoregressive models with exogenous inputs (ARX) for which the parameters switch according to a Markov chain and general Markov Jump Linear Systems (MJLSs) with full-state information. The transition probabilities of the Markov chain are assumed to be known, but the active subsystem is unknown. The application of the method to identification results in a recursive identification method for the joint maximum a posteriori probability estimate of these parameters and of the unknown mode. The method is guaranteed to provide an estimate whose joint posteriori probability is within a constant factor of that of the optimal estimate while reducing the computational complexity. The advantages and disadvantages of this

method with respect to the standard Expectation-Maximization (EM) algorithm are discussed in detail and a numerical example is provided.

Contribution 1.2. *Novel estimation and identification methods for computing joint maximum a posteriori probability estimates of the state and mode, as well as models, of Markov Jump Linear Systems, which are guaranteed to lie within a pre-specified bound of the optimal estimates.*

1.6.3 Model Predictive Control for Quadcopters

The third research direction is related to the challenge of trajectory tracking control for quadcopters. That is, a control strategy is proposed that allows the quadcopter to follow any feasible reference for almost (see Section 1.5.1.2) any initial condition. A cascaded control approach is used, consisting of an inner- and an outer-loop, as is common in quadcopters as has already been discussed in Section 1.5. The outer-loop controller handles the position control and provides references for the inner-loop controller, which, in turn, handles the attitude control. This cascaded strategy allows the outer-loop control problem to be tackled in the realm of linear systems, whereas the inner-loop control problem is still nonlinear, but simpler than the original control problem. The desired features of this controller include: the ability to anticipate future reference information, i.e., 'to look ahead'; explicit handling of constraints, such as maximum thrust generated by the propellers; fast computation times, allowing the controller to be implemented on a quadcopter platform; and stability guarantees, which ensure safety of the quadcopter if the model assumptions are satisfied.

To achieve this, Model Predictive Control (MPC) [143] is used for the outer-loop controller. As the name suggests, MPC uses a model of the quadcopter to predict which control inputs are needed to track the reference. It also allows the handling of constraints by solving an optimal control problem and fast computation times because of its receding horizon implementation. This outer-loop controller uses recent results for global stabilization of linear systems subject to input saturations in a Model Predictive Control strategy. The inner-loop controller is an almost globally asymptotically stable attitude controller that guarantees that references generated from the outer-loop controller are tracked asymptotically.

By combining the globally stable MPC controller for the outer-loop with an almost globally asymptotically stable attitude controller for the inner-loop, the whole cascaded closed-loop system is proven to be almost globally asymptotically stable.

Contribution 1.3. *A cascaded, model predictive control strategy that (i) allows for tracking of references while explicitly considering thrust constraints; (ii) is almost globally asymptotically stable; (iii) and is computationally inexpensive due to the formulation of the MPC as a quadratic program.*

1.6.4 Fast Landing of Quadcopters while Avoiding Vortex Ring State

The fourth and last challenge considered in this thesis is that of computing fast and efficient landing trajectories for a quadcopter, while avoiding the Vortex Ring State (VRS). The VRS is a well-known effect in helicopters that occurs when a fast vertical descent is performed with low horizontal speed. This causes the vortices at the tips of the blade to grow, which results in high fluctuations in the lift that the quadcopter can provide, making it (almost) uncontrollable. Although recovery is possible in some cases, entering the VRS often results in a crash, with potential damage to the quadcopter and the environment as a result[1]. The aerodynamics in the VRS are turbulent and difficult to model, therefore the aim is to avoid this region altogether and it is modeled as a hard constraint on the system.

Both time-optimal and energy-optimal vertical descent trajectories are investigated, considering the VRS constraint. The constraint is a region in the velocity space of the blade disk frames of the quadcopter that cannot be entered. This constraint is non-convex, which, combined with the nonlinear dynamics and high-dimensional state of a quadcopter, makes computing these trajectories a complex problem. Moreover, the common strategy of separating the problem into path and motion planning (see Section 1.4) is not possible, since the optimal path itself depends on the motion of the quadcopter.

Because of the complexity of the problem a numerical, approximate method is used to compute the trajectories. The modeling of the VRS is improved compared to previous papers [193, 192] and the first energy-optimal descent trajectories while avoiding the VRS are provided. Finally, the results are carefully discussed and several interesting directions for future work are proposed.

Contribution 1.4. *The Vortex Ring State for quadcopters is carefully modeled and time-optimal and energy-optimal vertical descent trajectories are computed that avoid the VRS altogether, providing fast and efficient vertical maneuvers while ensuring the safety of the quadcopter.*

1.7 Further interest of the thesis

Although this thesis is written from the viewpoint of, and inspired by, quadcopter applications, some of the provided solutions are more general and can be of interest for other applications. For instance, the method described in Section 1.6.1 can be used in any application that uses an IMU to estimate the attitude. Moreover, the introduced similarity measure and corresponding adaptation rule could be useful in other applications where adaptation is needed based on the distortion of one signal compared to another. Moreover, as already mentioned, the estimation algorithms described in Section 1.6.2 apply to general MJLS that

[1] https://youtu.be/LCret4rv0HE

need not be associated with quadcopters. The methods used in the cascaded MPC strategy described in Section 1.6.3 could also be applied to other cascaded systems. Moreover, the established stability results for the discrete-time MPC strategy used on a continuous time system is of interest in a broader application context. Lastly, the planning methods described in Section 1.6.4 can also be used in other settings where time and energy efficient trajectories are desired.

1.8 Structure of this thesis

The remainder of this thesis is organized into three parts: Part II contains the contributions related to estimation; the chapters related to control and planning are provided in Part III; and the thesis is concluded in Part IV, together with recommendations for future work. Each of the chapters in this thesis are based on either conference or journal articles and can thus be read independently.

The first part contains three chapters, of which the first is related to Contribution 1.1 and in the other two chapters Contribution 1.2 is presented.

In the second part two chapters are presented, related to Contribution 1.3 and 1.4, respectively.

The last part contains the concluding statements of this thesis, together with recommendations for future work. The bibliography is provided at the end of the thesis, sorted alphabetically.

1.9 List of publications

Peer-reviewed journal articles

- A.R.P. Andriën, E.Lefeber, D.J. Antunes, and W.P.M.H. Heemels, "Model Predictive Controller for Quadcopters with Almost Global Trajectory Tracking Guarantees", *In preparation for journal submission.*
- A.R.P. Andriën and D.J. Antunes, "Near-Optimal MAP Estimation for Markov Jump Linear Systems Using Relaxed Dynamic Programming", *IEEE Control Systems Letters*, vol.4, no. 4, pp. 815-820, 2020.
- L. Zhou, J.Y. Yoon, A.R.P. Andriën, M.I. Nejad, B.T. Allison, and D.L. Trumper, "FlexLab and LevLab: A Portable Control and Mechatronics Educational System", *IEEE/ASME Transactions on Mechatronics*, vol.25, no. 1, pp. 305-315, 2020.

Peer-reviewed conference articles

- A.R.P. Andriën, A. Talaeizadeh, D.J. Antunes, and W.P.M.H. Heemels, "Avoiding the Vortex Ring State in Quadcopters", *Submitted to the IEEE Conference on Robotics and Automation (ICRA)*, 2022.
- A.R.P. Andriën and D.J. Antunes, "Near-optimal Recursive Identification for Markov Switched Systems", *Accepted to the 2021 IEEE 60th Conference on Decision and Control (CDC)*, 2021.
- A.R.P. Andriën, D. Kremers, D. Kooijman, and D.J. Antunes, "Model Predictive Tracking Controller for Quadcopters with Setpoint Convergence Guarantees", *2020 American Control Conference (ACC)*, pp. 3205-3210, 2020.
- A.R.P. Andriën and D.J. Antunes, "Filtering and smoothing in the presence of outliers using duality and relaxed dynamic programming", *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 6038-6043, 2019.
- A.R.P. Andriën, D.J. Antunes, M.J.G. van de Molengraft, and W.P.M.H. Heemels, "Similarity-Based Adaptive Complementary Filter for IMU Fusion", *2018 European Control Conference (ECC)*, pp. 3044-3049, 2018.

Part II

Estimation

I

Chapter 2

Similarity-Based Adaptive Complementary Filter for IMU Fusion

This chapter addresses the attitude estimation problem using vector and gyroscope measurements. It proposes a novel adaptation scheme for the complementary filter cut-off frequency which is based on the similarity between independent estimates obtained from the vector and gyroscope measurements. The adaptive complementary filter is also derived on the special orthogonal group and convergence of the filter is established. The effectiveness of the approach is demonstrated with simulation results.

This chapter is based on [8]

2.1 Introduction

Estimating the attitude of a rigid body based on sensor data from an Inertial Measurement Unit (IMU), consisting of (three-axes) accelerometers, gyroscopes and magnetometers, is crucial in many applications and in particular in the context of Unmanned Aerial Vehicles (UAVs). As a result much work has been developed on attitude estimation over the years [135, 133, 218, 19, 69, 140, 174, 162, 53, 13, 136, 82, 137]. However, the attitude estimation problem has gained a renewed interest recently. In fact, the trend in the last few years in the UAV industry is that of size and cost reduction. Therefore, it is often the case that the available data from low-cost IMUs does not provide sufficiently accurate information for the current estimation techniques to deliver the desired attitude accuracy [137].

The most popular IMU-based attitude estimation approaches are the Kalman filter (KF) and the complementary filter (CF) [135]. There are many variants of the Kalman filter, such as linear [133], [218], extended Kalman filters (EKFs) [19, 69, 140, 174] and unscented Kalman filters (UKFs) [162], [53]. Some of the advantages of linear KFs are their effectiveness and low implementation cost, as well as the guaranteed optimality and stability under the assumption of linear process and measurement models as well as Gaussian noise and disturbance processes. However, for nonlinear processes the EKF and UKF typically show better results, at the cost of more computational complexity and the absence of stability guarantees. In the context of complementary filters, one of the first applications of the linear CF to attitude estimation was presented in [13]. Nonlinear CFs that operate directly on the Special Orthogonal Group (SO(3)) have also been developed and successfully implemented in [136], [82] and [137]. One of the main advantages of the CFs over KFs is the fact that they do not require an explicit model of the dynamics and/or sensors, which is hard to obtain in many applications and in particular in the context of small-scale UAVs, where some aerodynamic effects such as turbulence, drag and ground effects have a significant effect but are hard to model. Another advantage is its simplicity, since it relies on simple low and high pass filtering of the sensor data.

The complementary filter is a time-invariant filter, both in the linear [13] and nonlinear versions [136, 137], which performs well under the assumption that the accelerometer and magnetometer only measure the gravitational vector and earth magnetic field, respectively. However, under the effect of body accelerations and magnetic disturbances caused by the environment, this assumption is no longer valid, resulting in poor attitude estimates (see Figure 2.2). Therefore it is reasonable to expect that adaptive filtering [184], meant to cope with signals with time-varying frequency content, could potentially lead to better performance. While the work in this direction is limited, there are some recent contributions for UAV attitude estimation, which have focused on the adaptation of the filters in order to account for the shortcomings in the sensors. For instance, in [125] an

adaptive linear KF is proposed, where the measurement noise covariance matrix is adapted based on the difference between the accelerometer measurements and the gravity vector. In [182] and [51] adaptive EKFs are proposed, where the adaptation is based on the deviation of the actual from the expected measurement and the same deviation used in a fuzzy logic setting, respectively. Adaptive complementary filters (ACFs) have also been used as in [158], [216] and [198], where adaptation based on comparing the accelerometer with the gravity vector is used. Furthermore, [139] provides a complementary filter in the least squares sense (also known as Wahba's problem [204]), using the difference between the output of the filter and the measurement by the accelerometer for adaptation and [115] proposes a multiple model adaptive complementary filter, with adaptation based on the same measure.

In this chapter a new adaptation rule for changing the parameter of the complementary filter online is proposed, determining which measurement (either from the gyroscope or from the vector) is more relevant for the attitude estimate. The rationale behind the approach is that the attitude estimate should rely on vector measurements if the independent estimates using only vector measurements and only gyroscope measurements are *similar over a time window*, possibly apart from a constant factor due to low-frequency noise of the gyroscope estimate. Otherwise the attitude estimate should rely instead on the gyroscope, since not meeting this similarity condition is an indication that significant disturbances are affecting the vector measurements. Moreover, it is shown that this adaptation concept is not only applicable for estimating single angles with linear models but also for estimating rotational matrices considering the nonlinear kinematic model in SO(3). In fact, the work in [137] is extended to this adaptation setting and it is shown that the convergence proofs given there can be easily extended to the proposed setting in this chapter as well. The effectiveness of the approach is illustrated through simulation results using a full non-linear model of a quadcopter and IMU sensors.

The contribution of this chapter is therefore twofold. First a novel adaptation rule for complementary filters based on the similarity between gyroscope and vector estimates is proposed; and second convergence of this filter is established.

The remainder of the chapter is organized as follows. The standard attitude estimation framework is described in Section 2.2, where the need for an adaptive approach is also motivated. In Section 2.3 the proposed method considering linear models is introduced and in Section 2.4 the method is extended to the special orthogonal group and convergence is shown. Section 2.5 shows the main advantages over the non-adaptive filter using simulations and Section 2.6 contains concluding remarks and directions for future work.

2.2 Background & Motivation

Let $\mathcal{I}$ denote an inertial world fixed frame with z-axis aligned with the gravity vector, but pointing upwards, and the IMU fixed frame be denoted by $\mathcal{B}$, which will be referred to as the body-fixed frame. Moreover, let $R \in SO(3)$ denote the rotation matrix from $\mathcal{B}$ to $\mathcal{I}$ and $\omega = [\omega_x, \omega_y, \omega_z]^T \in \mathbb{R}^3$ denote the angular velocity of $\mathcal{B}$ with respect to $\mathcal{I}$ expressed in $\mathcal{B}$. Then the attitude kinematics are given by

$$\dot{R} = R\omega_\times, \tag{2.1}$$

where $a_\times$ denotes the skew symmetric matrix of a such that $a_\times v = a \times v$ for all vectors a and v and $\text{vex}(\cdot)$ denotes the inverse operation, so that $\text{vex}(a_\times) = a$. The rotation matrix R can be parametrized by the three x-y-z Tait-Bryan angles rotating around the axes of the body frame of the UAV, namely roll (ϕ), pitch (θ) and yaw (ψ). The attitude kinematics in these coordinates are given by

$$\dot{\lambda} = Q(\lambda)\omega, \tag{2.2}$$

where $\lambda = [\phi, \theta, \psi]^T \in \mathbb{R}^3$ is the angle vector and $Q(\lambda)$ is given by

$$Q(\lambda) = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & \frac{s_\phi}{c_\theta} & \frac{c_\phi}{c_\theta} \end{bmatrix}, \tag{2.3}$$

where c_ϵ, s_ϵ and t_ϵ denote the cosine, sine and tangent of an angle ϵ, respectively. Note that for small angles $Q(\lambda) \approx I$, and a linear model is obtained. This linear model is often considered in practice and will be used in the sequel to illustrate the ideas in this chapter.

The accelerometer measures body accelerations in the three axes in $\mathcal{B}$ (neglecting Coriolis effects), as given by

$$a_\mathcal{B} = R^T \ddot{\rho} - R^T \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + n_a, \tag{2.4}$$

where $\ddot{\rho} \in \mathbb{R}^3$ are the body accelerations in $\mathcal{I}$, g is the Earth's gravitational acceleration and n_a is the accelerometer noise.

The magnetometer measures the magnetic field in the body-frame, which is characterized by

$$m_\mathcal{B} = R^T m_I + d_m + n_m, \tag{2.5}$$

where $m_I \in \mathbb{R}^3$ is the Earth's magnetic field in the inertial frame, $n_m \in \mathbb{R}^3$ is the magnetometer noise and $d_m \in \mathbb{R}^3$ contains the magnetic disturbances caused by the environment, such as electric motors and nearby ferromagnetic materials.

The accelerometer and magnetometer measurements can be combined to give an algebraic estimate of the angle vector. In fact, by assuming negligible accelerations and magnetic disturbances, the measurements are given by

$$a_B \approx -R^T \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}, \qquad m_B \approx R^T m_I. \tag{2.6}$$

This results in a direct estimate of the angle vector as

$$\lambda_v = \lambda + n_v, \tag{2.7}$$

where $n_v \in \mathbb{R}^3$ is considered non-stationary noise. The non-stationary behavior results from the assumption that the accelerometer only measures the gravity vector, which is valid only at slow accelerations in near-hover, and magnetic disturbances caused by the environment and the motors of the quadcopter are negligible. This results in correlated, high frequency noise for the angle vector measurements.

An estimate of the angular velocities is given by the gyroscope as

$$\omega_g = \omega + n_\omega + n_{\text{bias}}, \tag{2.8}$$

where $n_\omega \in \mathbb{R}^3$ is considered white noise and n_{bias} is a sensor bias. By assuming small angles and integrating the angular rate measurements according to (2.2) another estimate for the angle vector, denoted as λ_g, can be obtained. However, even when assuming $n_{\text{bias}} = 0$, due to the white noise component in (2.8) this estimate will be corrupted by a random walk signal $b(t)$ leading to

$$\lambda_g = \lambda + b(t), \tag{2.9}$$

which in practice leads to a poor low-frequency estimate of λ using this method, often interpreted as a time-varying offset/bias. Note that is assumed here that there are no other disturbances, e.g. device failure, acting on the gyroscope, so that the angle estimate resulting from it can be assumed to be accurate up to the bias.

2.2.1 Complementary Filter

As mentioned previously, the complementary filter is a common method to fuse the two measurements, λ_v and λ_g, combining the strengths of both. In the Laplace domain it takes the form

$$\hat{\Lambda}(s) = \underbrace{\frac{C(s)}{s + C(s)}}_{F_1(s)} \Lambda_v(s) + \underbrace{\frac{s}{s + C(s)}}_{F_2(s)} \frac{\Omega_g(s)}{s}, \tag{2.10}$$

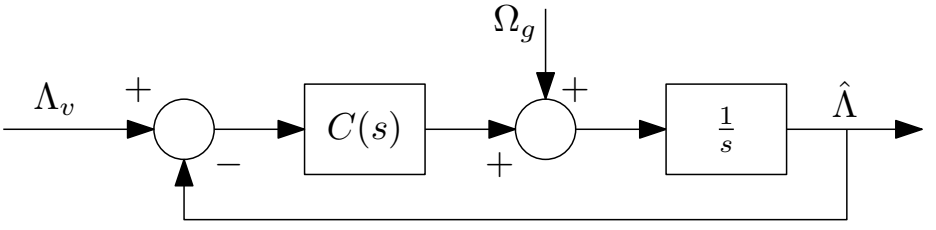


Figure 2.1: Complementary filter.

where F_1 and F_2 are a low-pass and a high-pass filter, respectively, which satisfy

$$F_1(s) + F_2(s) = 1, \tag{2.11}$$

for any choice of the compensator $C(s)$ and $\Lambda_v(s)$ and $\Omega_g(s)$ are the Laplace transforms of λ_v and ω_g, respectively. Schematically this can be represented as in Figure 2.1. Although there have been considerable efforts made in designing $C(s)$ in the literature, the most common choice is that of a simple gain $C(s) = \alpha > 0$, for which the filter in (2.10) combined with (2.7) and (2.8) results in

$$\begin{aligned}\hat{\Lambda}(s) = \frac{\alpha}{s+\alpha}\Lambda(s) \; &+ \frac{s}{s+\alpha}\frac{\Omega(s)}{s} + \frac{\alpha}{s+\alpha}N_v(s) \\ &+ \frac{s}{s+\alpha}\frac{N_\omega(s)}{s},\end{aligned} \tag{2.12}$$

where $\Lambda(s)$, $\Omega(s)$, $N_v(s)$ and $N_\omega(s)$ are the Laplace transforms of λ, ω, n_v and n_ω, respectively. Note that the first two terms on the right hand side in (2.12) assure that the method converges in the absence of noise, whereas the two noise terms are filtered. The random walk term is high-pass filtered, which leads to low-pass filtered white noise. The term N_v, which as discussed before is assumed as mostly having high-frequency content, is low-pass filtered.

2.2.2 Motivation

If the noise N_v would be stationary, a single α could be selected and the complementary filter would provide an adequate solution to the problem of estimating the angles. However, this noise is not stationary since:

- Under accelerations the accelerometer does not measure purely the gravity vector anymore, thereby distorting the angles calculated from this measurement
- Disturbances of the earth magnetic field caused by (intensive) motor usage and/or the environment will distort the angle vector estimate

This is illustrated in the top of Figure 2.2, where some typical angle estimates from the accelerometer and gyroscope measurements are shown for the pitch angle ϕ, together with the actual angle and Figure 2.2 shows the difference between the accelerometer measurements and the gravity vector, normalized. This shows that the integrated gyroscope measurement is accurate up to the bias, whereas the accelerometer angle does not show a bias in steady state but has more noise as well as large distortions when accelerations are present. As shown in the figure, for the complementary filter, a small α will filter significantly the non-stationary noise of the vector estimates but will not be robust to the offset of the estimate obtained with the gyroscope. In turn, a large α will be too sensitive to the non-stationary noise of the accelerometer. This motivates making α time-dependent and adapt it with respect to perceived properties of N_v (for which a similarity based approach will be used in the sequel).

2.2.3 Problem Formulation

The problem considered in this chapter is that of estimating the attitude of a body-fixed frame with respect to that of the inertial frame, parametrized by the rotation matrix R, subject to the kinematics in (2.1) and given the vector and gyroscope measurements given by (2.7) and (2.8), respectively. In particular, the attitude estimate should be robust to disturbances acting on the vector measurements.

2.3 Proposed Adaptive Method

In order to address the shortcomings described in the previous section an adaptive gain complementary filter is proposed, where the gain is adapted according to the reliability of the angle vector measurement, λ_v.

The adaption is based on the following observation:
If the angular estimate achieved from integrating the gyroscope measurement is similar to the angular estimate determined from the accelerometer and magnetometer measurements over a time window, then the accelerometer and magnetometer measurements are not distorted.

In order to quantify this similarity between the angle vector and gyroscope measurement, a similarity measure is introduced as

$$S(t) = \min\left(\bar{S}, \min_c J(c,t)\right), \tag{2.13}$$

where $0 < \bar{S} < \infty$ is the upper bound for the similarity measure and

$$J(c,t) = \sqrt{\int_{t-h}^{t} \left(\lambda_v(\tau) - \lambda_g(\tau) - c\right)^2 \, \mathrm{d}\tau}, \tag{2.14}$$

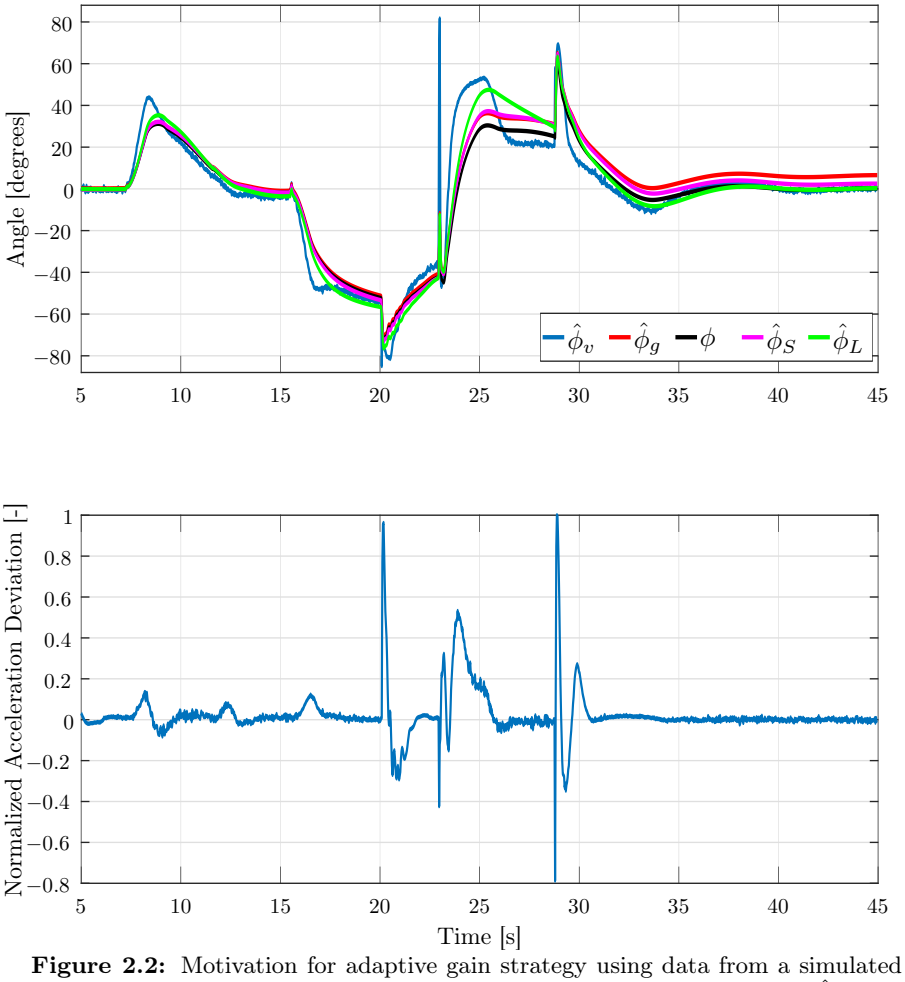


Figure 2.2: Motivation for adaptive gain strategy using data from a simulated quadcopter. (top) Typical angle estimates obtained using the vector ($\hat{\phi}_v$) and gyroscope ($\hat{\phi}_g$) measurements, together with the actual angle (ϕ) and the estimate given by the complementary filter with a small α ($\hat{\phi}_S$) and a large α ($\hat{\phi}_L$). (bottom) Associated accelerometer signals, showing the connection between accelerations and deviations for the vector estimate. Note that for the complementary filter, a small α will filter significantly the non-stationary noise of the vector estimates but will be not robust to the offset of the estimated obtained with the gyroscope and a large α will be too sensitive to the non-stationary noise of the accelerometer.

which depends on the signal values λ_g and λ_v in a moving time window of length h, i.e., in the interval $\tau \in [t-h, t]$. The cost function in (2.14) essentially shifts the two signals on top of each other by using c, after which the signals are subtracted from each other and integrated, which gives a measure of similarity of shape. It can be shown that the minimizer of (2.13) is simply the mean value of the two vectors over the window length, i.e.

$$c^*(t) := \arg\min_c J(c,t) = \frac{1}{h}\int_{t-h}^{t} (\lambda_v(\tau) - \lambda_g(\tau))\,\mathrm{d}\tau. \tag{2.15}$$

This results in very fast calculation times for $S(t)$. The adaptive gain is then defined as

$$\alpha(t) = \bar{\alpha} e^{-KS(t)}, \tag{2.16}$$

where $\bar{\alpha} > 0$ and $K > 0$. This results in

$$0 < \underline{\alpha} \leq \alpha(t) \leq \bar{\alpha} \qquad \forall t \in \mathbb{R}_{\geq 0}, \tag{2.17}$$

where $\underline{\alpha} = \bar{\alpha} e^{-K\bar{S}}$. By varying the gain by the exponential of the similarity measure, a quick reaction to disturbances on the vector measurements is ensured. Moreover, $\bar{S}$ ensures that the adaptive gain remains strictly positive and in practice it can be chosen arbitrarily high as to not interfere with the cost function in (2.14).

In order to show convergence of the observer with adaptive gain in the absence of noise, consider the time domain representation of the linear complementary filter as presented in (2.12) with the adaptive gain (2.16), leading to

$$\dot{\hat{\lambda}} = \omega_g + \alpha(t)(\lambda_v - \hat{\lambda}). \tag{2.18}$$

The following proposition establishes convergence of the proposed observer.

Proposition 2.1. *Consider the rotation kinematics* (2.2) *and measurements given by* (2.7) *and* (2.8). *Let* $\hat{\lambda}$ *denote the solution to* (2.18) *and let* $\tilde{\lambda} = \lambda - \hat{\lambda}$ *denote the angle error. Then, under the assumption that* $n_{bias} = 0$, *the error* $\tilde{\lambda}$ *converges to zero in the absence of noise and is input-to-state (ISS) stable in the presence of noise considering the noise properties as discussed in Section 2.2.*

Proof. We introduce the Lyapunov function

$$V(\tilde{\lambda}) = \frac{1}{2}\tilde{\lambda}^2. \tag{2.19}$$

Taking the derivative of (2.19) with respect to time gives

$$\dot{V} = -\alpha(t)\tilde{\lambda}^2 - n_\omega\tilde{\lambda} - \alpha(t)n_v\tilde{\lambda}, \tag{2.20}$$

which is negative definite in the absence of noise, so that it can be concluded that the estimation error converges to zero in the absence of noise and is input-to-state (ISS) stable in the presence of noise considering the noise properties as discussed in Section 2.2. ■

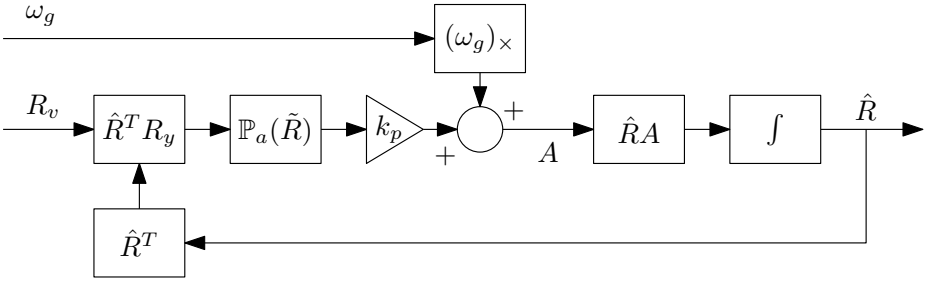


Figure 2.3: Passive complementary filter on $SO(3)$

2.4 Adaptive Complementary Filter on SO(3)

Now that the rationale behind the approach was shown for the linear case, the ACF is presented directly on $SO(3)$. The problem of estimating the attitude directly on the special orthogonal group can be formulated as determining an estimate $\hat{R} \in SO(3)$ of the rotation matrix R that rotates the body-fixed frame $\mathcal{B}$ to the inertial frame $\mathcal{I}$, from measurements provided by the accelerometer, gyroscope and magnetometer of the IMU. If the estimator frame is defined as $\mathcal{E}$, then $\hat{R}$ rotates $\mathcal{E}$ to $\mathcal{I}$. As the error of the filter on $SO(3)$, consider

$$\tilde{R} = \hat{R}^T R \in SO(3), \tag{2.21}$$

which is the rotation from the estimator frame to the inertial frame. Note that the goal of the observer is to drive the estimation error to $\tilde{R} \to I_3$, since this means that $\hat{R}$ and R coincide.

In [136] a passive complementary filter on $SO(3)$ was introduced, which is expanded here by introducing a similarity measure for the special orthogonal group. The proposed estimator in [136] has the following kinematics

$$\dot{\hat{R}} = \hat{R}\left((\omega_g)_\times + k_p \mathbb{P}_a(\tilde{R})\right), \tag{2.22}$$

where $k_p > 0$ is the observer gain and $\mathbb{P}_a(H) = \frac{1}{2}(H - H^T)$ is the anti-symmetric matrix projection operator in matrix space. Naturally R in (2.21) is not available, but by using $R = R_y$ to generate the error term $\tilde{R}$, a filter resembling the complementary filter is achieved. This becomes clear by comparing the block-diagram representation in Figure 2.3 with the classical complementary filter in Figure 2.1.

As in the linear case, the estimator suffers from disturbances applied to the vector measurements, for which a similar adaptation scheme is presented. In order to compare the gyroscope and vector measurements on $SO(3)$, first the rotation of the body frame with respect to the inertial frame as determined

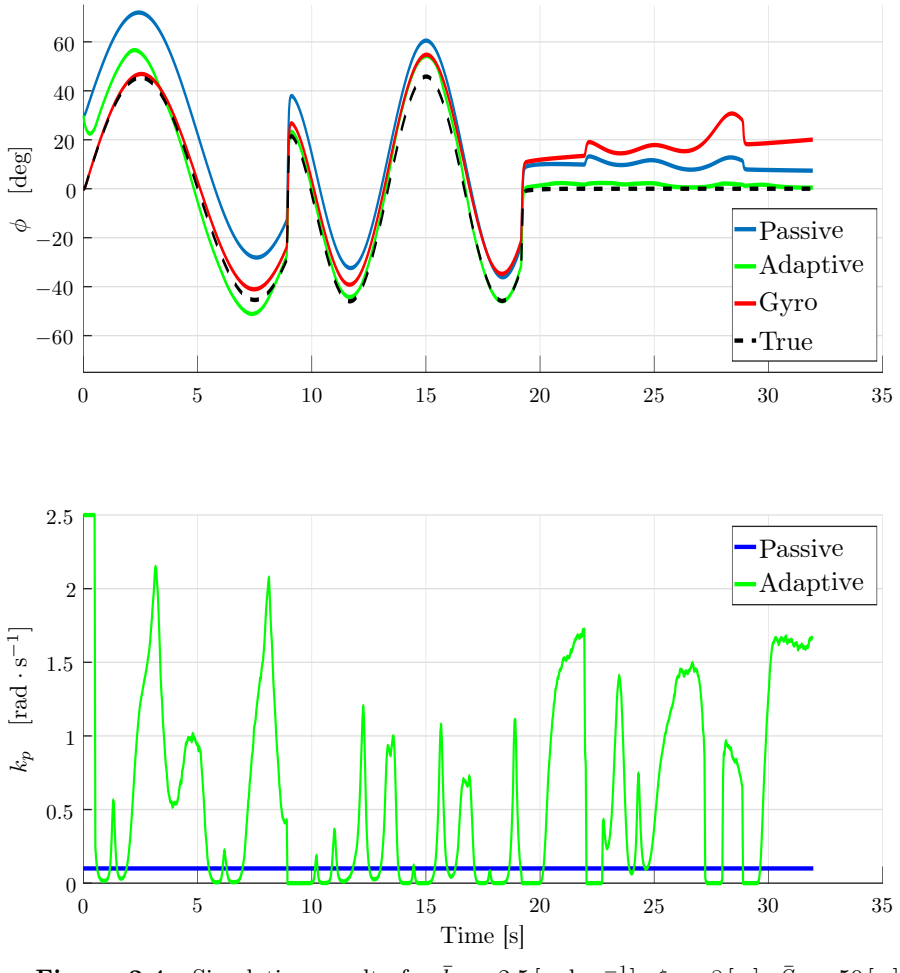


Figure 2.4: Simulation results for $\bar{k} = 2.5\,[\text{rad} \cdot \text{s}^{-1}]$, $\xi = 8\,[-]$, $\bar{S} = 50\,[-]$, $h = 0.5\,[s]$, $k_p = 0.1\,[\text{rad} \cdot \text{s}^{-1}]$. The top shows the angular estimate resulting from the non-adaptive (blue) and adaptive filter (green), gyroscope integration (red) and the actual angle (dashed black) for the roll angle. The bottom plot shows the adaptive (green) and fixed gain (blue) over time.

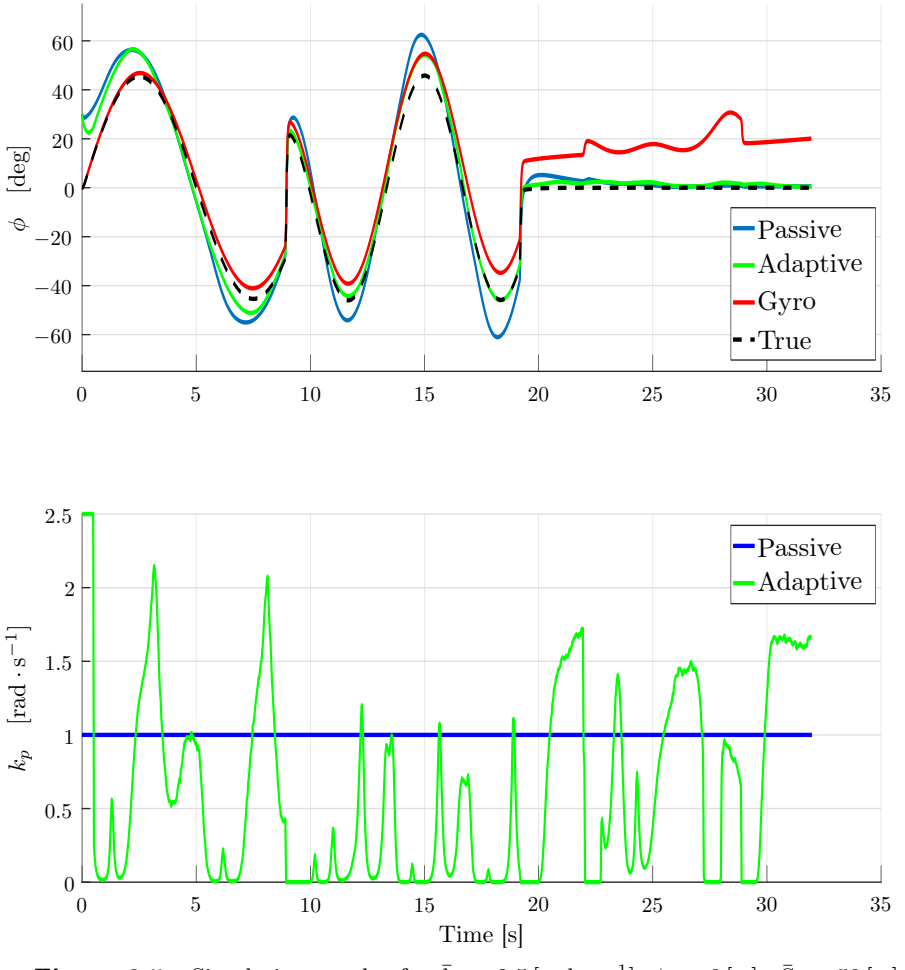


Figure 2.5: Simulation results for $\bar{k} = 2.5\,[\text{rad} \cdot \text{s}^{-1}]$, $\xi = 8\,[-]$, $\bar{S} = 50\,[-]$, $h = 0.5\,[s]$, $k_p = 1\,[\text{rad} \cdot \text{s}^{-1}]$. The top shows the angular estimate resulting from the non-adaptive (blue) and adaptive filter (green), gyroscope integration (red) and the actual angle (dashed black) for the roll angle. The bottom plot shows the adaptive (green) and fixed gain (blue) over time.

from the vector and gyroscope measurements are defined as $R_v \in SO(3)$ and $R_g \in SO(3)$, respectively. The error between these two measurements in $SO(3)$ can be seen as

$$\tilde{R}_{vg} = R_v^T R_g \in SO(3), \tag{2.23}$$

so that by the same insight as for the linear case, if this is close to I_3, it can be concluded that the vector measurements can be trusted. To quantify this the cost function is defined as

$$E_{vg} = \frac{1}{4}||I_3 - \tilde{R}_{vg}||_F^2 = \frac{1}{2}\mathrm{tr}(I_3 - \tilde{R}_{vg}), \tag{2.24}$$

which should be small over a time window. Next, the similarity measure on $SO(3)$ is defined, with a slight abuse of notation, as

$$S(t) = \min\left(\bar{S}, \min_c J(c,t)\right), \tag{2.25}$$

where $0 < \bar{S} < \infty$,

$$J(c,t) = \sqrt{\int_{t-h}^{t} (E_{vg}(\tau) - c)^2 \,\mathrm{d}\tau}, \tag{2.26}$$

and $c^*(t)$ can again be shown to be the mean of $E_{vg}(t)$ over the time window. The adaptive gain is again defined as

$$k_p(t) = \bar{k}e^{-\xi S(t)}, \tag{2.27}$$

with $\bar{k} > 0$ and $\xi > 0$, so that

$$0 < \underline{k} \leq k_p(t) \leq \bar{k} \qquad \forall t \in \mathbb{R}_{\geq 0}, \tag{2.28}$$

with $\underline{k} = \bar{k}e^{-\xi\bar{S}}$. The following theorem establishes convergence of the proposed adaptive estimator.

Theorem 2.2. *Consider the rotation kinematics* (2.1) *and measurements given by* R_v *and* R_g*. Let* $\hat{R}$ *denote the solution to* (2.22) *with the adaptive gain* (2.27) *and let the error variable* $\tilde{R}$ *be defined as in* (2.21)*. Then the estimation error* $\tilde{R}$ *converges to zero.*

Proof. In order to analyze the convergence of the estimator with adaptation the same steps as in [137] are followed. First, consider the Lyapunov function

$$E_t = \frac{1}{4}||I_3 - \tilde{R}||_F^2 = \frac{1}{2}\mathrm{tr}(I_3 - \tilde{R}) \tag{2.29}$$

and its derivative with respect to time is

$$\dot{E}_t = -k_p(t)|\mathrm{vex}(\mathbb{P}_a(\tilde{R}))|^2, \tag{2.30}$$

which is negative definite so that it can be concluded, using similar arguments as in [137], that the estimation error converges to zero. □

The above theorem shows that the filter in [137] can be extended to an adaptive filter, and in particular to the one proposed here, without affecting the convergence guarantees.

2.5 Simulations

In order to demonstrate the effectiveness of the approach simulations were performed using a model of a quadcopter and compared to the non-adaptive passive complementary filter as proposed in [137].

Two different settings for the passive complementary filter are compared for the same trajectory in Figures 2.4 and 2.5. For the adaptive filter the maximum gain was set to $\bar{k} = 2.5\,[\mathrm{rad}\cdot\mathrm{s}^{-1}]$, the gain factor was set to $\xi = 8\,[-]$, the upper bound for the similarity measure was set to $\bar{S} = 50\,[-]$ and the window length was set to $h = 0.5\,[s]$ for both figures. For the non-adaptive filter the gain was fixed to $k_p = 0.1\,[\mathrm{rad}\cdot\mathrm{s}^{-1}]$ and $k_p = 1\,[\mathrm{rad}\cdot\mathrm{s}^{-1}]$ in Figures 2.4 and 2.5, respectively. The trajectories in both figures are the same.

In both figures the top plot shows the angular estimate resulting from the non-adaptive (blue) filter, adaptive filter (green), gyroscope integration (red) and the actual angle (dashed black) for the roll angle. The bottom plot shows the adaptive gain (green) together with the fixed gain (blue) over time.

Together the figures display the advantage of the adaptive over the non-adaptive filter. In Figure 2.4 the fixed gain k_p is set low, thus giving more priority to the gyroscope measurements, resulting in good estimates during accelerations but a poor convergence in steady-state, resulting in a large steady-state deviation. In contrast, the larger gain setting of Figure 2.5 results in a better performance in steady-state conditions, but yields poor performance during accelerations.

The proposed adaptive filter performs better than its non-adaptive counterpart in both simulations. It allows for a larger initial gain, resulting in better steady-state behavior, whilst still allowing the gain to be lowered using the adaptation rule from (2.27) in order to avoid deviations during accelerations. This essentially gives the user more knobs to turn in order to adjust the filter to the behavior of the sensors of the IMU. That the adaptive filter outperforms the passive filter becomes more clear by comparing the error as defined in (2.29) for the passive filter with low and high gain settings and the adaptive filter as in Figure 2.6.

2.6 Conclusions & Future Work

The problem of attitude estimation considering disturbances acting on the vector measurements was solved using an adaptation scheme for the passive complementary filter directly on the special orthogonal group. The adaptation scheme presented is this chapter was proven to converge and simulations results showed

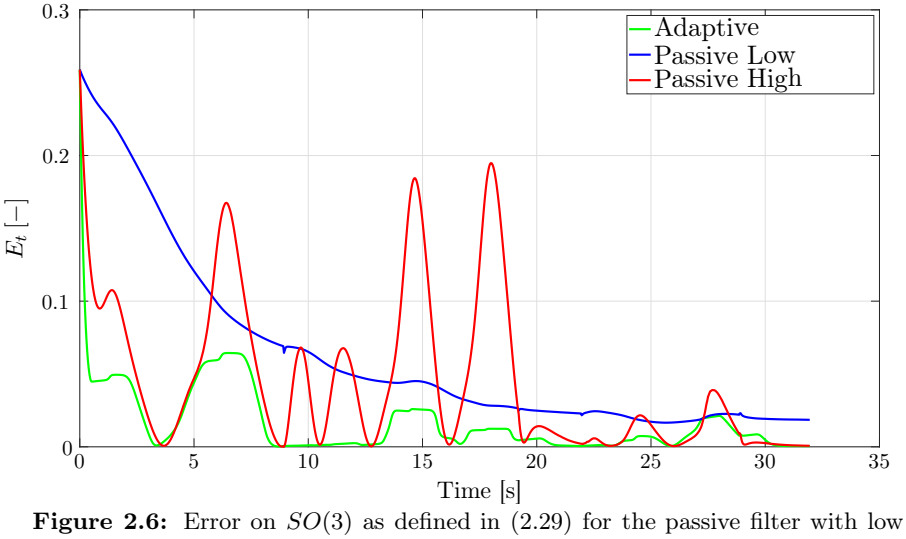


Figure 2.6: Error on $SO(3)$ as defined in (2.29) for the passive filter with low (blue) and high gain (red) settings and the adaptive filter (green)

that the adaptation scheme performs as expected, showing the advantages compared to the non-adaptive filter.

Future work will include experimental testing of the method, as well as the use of the adaptation scheme on the passive complementary filter with bias correction and other, more advanced complementary filters that have been proposed in the literature. Moreover, different adaptation schemes based on machine learning can be considered.

Chapter 3

Estimation for Markov Jump Linear Systems

Computing the optimal joint maximum a posteriori probability (JMAP) estimate of the state and mode of a Markov jump linear system (MJLS) is known to be a computationally intractable problem. This chapter provides a novel approximate method for such a problem that guarantees to be within a pre-specified bound of the optimal estimate. The proposed method builds upon relaxed dynamic programming. Through numerical examples, it is shown that this method can lead to better estimates with less computations than previous suboptimal methods proposed in the literature.

This chapter is based on [6, 4]

3.1 Introduction

Markov Jump Linear Systems (MJLS) are switched linear systems, where the switching between modes occurs according to a finite state Markov chain. One can find applications of MJLS in many fields, such as signal processing [116], control [45], economics [28] and maneuvering target tracking [171].

It is well-known that the optimal state and mode sequence estimator, in the sense of joint maximum a posteriori probability (JMAP), for discrete time MJLS subject to uncorrelated Gaussian disturbances and noise, relies on a bank of Kalman filters (KFs), see [17]. However, the required number of filters grows exponentially with time, resulting in a computationally intractable solution [17]. This directly leads to the use of suboptimal algorithms, such as sampling [58], iterative [57] and moving horizon filters [188], among others [55]. The most commonly used MJLS filters are the optimal linear minimum mean square filter [46] and the interacting multiple model (IMM) algorithm [17]. Suboptimal strategies for JMAP estimation for MJLS have been studied in [130], where iterative algorithms based on the expectation maximization algorithm have been implemented, iteratively combining a discrete optimization via the Viterbi algorithm with a fixed-interval Kalman smoother. More recently, suboptimal JMAP estimation for MJLS was investigated in [120], where a moving horizon strategy is combined with a constrained Viterbi algorithm in order to find the JMAP estimate subject to constraints. These and other algorithm for maneuvering target tracking, which are often modeled as MJLS, are reviewed and compared in [171].

In this work, a novel method for finding the JMAP estimate of both the state and mode sequence of a MJLS is proposed. The method differs from existing ones in the use of relaxed dynamic programming (RDP) to prune the number of mode histories, referred to as hypotheses. RDP was introduced in [129] as a method to reduce the complexity of solving dynamic programming (DP) problems by allowing for a suboptimal solution, which is still within a certain bound of the optimum. The method was applied to the *control* of switched (linear) systems in [129, 77, 219], where the mode can be controlled rather than it following a Markov chain as in MJLS.

To the best of the authors' knowledge, this chapter is the first to provide a method for the JMAP estimation of MJLS that guarantees near global optimality to within a pre-specified bound, whereas other suboptimal algorithms [130, 120], which often end up in local optima, have no such guarantees. Moreover, numerical examples show that this method can lead to better estimates with less computations than previous methods.

This chapters generalizes the results from [6], which focuses on linear systems with independent and identically distributed outliers, to MJLS.

The remainder of this chapter is organized as follows; Section 3.2 introduces the problem and Section 3.3 discusses the optimal solution. Section 3.4 contains the main methods and results of the chapter, showing that the JMAP estimator

for MJLS can be determined within an ϵ-factor of the optimal. Simulations show the efficacy of the approach in Section 3.5 and the chapter is concluded in Section 3.6.

Notation: A Gaussian random variable $a \in \mathbb{R}^n$ with mean μ and covariance $\Theta \succ 0$ is denoted by $a \sim \mathcal{N}(\mu, \Theta)$, i.e., the probability density function of a is given by $\Pr\{a\} = ((2\pi)^n|\Theta|)^{-\frac{1}{2}} e^{-\frac{1}{2}\|a-\mu\|^2_{\Theta^{-1}}}$, where $|\Theta|$ denotes the determinant of a matrix Θ and $\|a\|_\Theta := \sqrt{a^\intercal \Theta a}$.

3.2 Problem Description

Consider the discrete-time Markov jump linear system

$$x_{k+1} = A_{\sigma_{k+1}} x_k + B_{\sigma_{k+1}} u_k + G_{\sigma_{k+1}} w_k, \tag{3.1a}$$

for $k \in \mathbb{N}_0 := \mathbb{N} \cup \{0\}$, where $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^m$ and $w_k \in \mathbb{R}^{n_w}$ denote the system state, known exogenous input and process noise, respectively. Furthermore, consider the following output equation

$$y_k = C_{\sigma_k} x_k + D_{\sigma_k} u_k + H_{\sigma_k} v_k \tag{3.1b}$$

for $k \in \mathbb{N}$, where $y_k \in \mathbb{R}^r$ and $v_k \in \mathbb{R}^{n_v}$ denote the observation and measurement noise, respectively. Moreover, let σ_k denote the mode the system is in at time k, which is described by a Markov chain with s states and known transition probabilities

$$\rho_{ij} := \Pr\{\sigma_{k+1} = i | \sigma_k = j\}, \quad i \in S,\, j \in S, \tag{3.2}$$

where $S = \{1, 2, \ldots, s\}$, and initial distribution $\rho_i = \Pr\{\sigma_0 = i\}$. The initial state x_0, w_k and v_k are considered to be independent and identically distributed (i.i.d.) Gaussian random variables with the following distributions

$$w_k \sim \mathcal{N}(0, Q), \quad v_k \sim \mathcal{N}(0, R), \quad x_0 \sim \mathcal{N}(\bar{x}_0, P_0), \tag{3.3}$$

with known mean $\bar{x}_0$ and such that, for all $i \in S$,

$$\mathcal{Q}_i := G_i Q G_i^\intercal \succ 0, \quad \mathcal{R}_i := H_i R H_i^\intercal \succ 0, \quad P_0 \succ 0.$$

The matrices A_i, $i \in \mathcal{S}$, are assumed to be invertible, which is typically the case when (3.1) results from discretization of a continuous-time problem. All the parameters of the model are assumed to be known, whereas, in practice, these can be estimated by identification methods for MJLS such as the ones given in [92, 41].

The objective is to find both the state estimates, $\hat{X}_0^H := [\hat{x}_0^\intercal, \ldots, \hat{x}_H^\intercal]^\intercal$, as well as the mode sequence, $\hat{\Sigma}_0^H := [\hat{\sigma}_0, \ldots, \hat{\sigma}_H]^\intercal$, that have the maximum likelihood, given measurements $Y_1^H := [y_1^\intercal, \ldots, y_H^\intercal]^\intercal$ and an arbitrary time $H \in \mathbb{N}$. This is equivalent to finding the state estimates and mode sequence that maximize the

joint conditional probability of the state estimates $\hat{X}_0^H$ and outlier sequence $\hat{\Sigma}_0^H$ given the measurements Y_1^H, i.e.

$$\left(\hat{X}_0^H, \hat{\Sigma}_0^H\right) = \arg \max_{\hat{X}_0^H, \hat{\Sigma}_0^H} \quad \Pr\left\{\hat{X}_0^H, \hat{\Sigma}_0^H | Y_1^H\right\}. \tag{3.4}$$

The conditional and joint probabilities are related as follows

$$\Pr\left\{\hat{X}_0^H, \hat{\Sigma}_0^H | Y_1^H\right\} = \frac{\Pr\left\{Y_1^H, \hat{X}_0^H | \hat{\Sigma}_0^H\right\} \Pr\left\{\hat{\Sigma}_0^H\right\}}{\Pr\left\{Y_1^H\right\}} \tag{3.5}$$

and since the denominator is independent of $\hat{X}_0^H$ and $\hat{\Sigma}_0^H$, (3.5) can be combined with (3.4) to give

$$\left(\hat{X}_0^H, \hat{\Sigma}_0^H\right) = \arg \max_{\hat{X}_0^H, \hat{\Sigma}_0^H} \quad \Pr\left\{Y_1^H, \hat{X}_0^H | \hat{\Sigma}_0^H\right\} \Pr\left\{\hat{\Sigma}_0^H\right\}. \tag{3.6}$$

Since it will be easier to work with logarithms of probabilities, the following problem is considered, equivalent to (3.4)

$$\begin{aligned}\left(\hat{X}_0^H, \hat{\Sigma}_0^H\right) = \arg \min_{\hat{X}_0^H, \hat{\Sigma}_0^H} \quad & -\log(\Pr\left\{Y_1^H, \hat{X}_0^H | \hat{\Sigma}_0^H\right\}) \\ & -\log(\Pr\left\{\hat{\Sigma}_0^H\right\}).\end{aligned} \tag{3.7}$$

3.3 Optimal Solution

The solution to the optimization problem given in the previous section is known to rely on a bank of Kalman filters, and for a non-switching model it reduces to a single Kalman filter [17, 46]. Here, the optimization will be rewritten as an optimal control problem, starting with the conditional probability of Y and X given Σ. By using the chain rule ($\Pr\{a, b|c\} = \Pr\{a|b, c\} \Pr\{b|c\}$, for random variables a, b, c) and the Markov property of the model (3.1), the natural logarithm of the conditional probability can be expressed as

$$\begin{aligned}\log(\Pr\left\{Y_1^H, \hat{X}_0^H | \hat{\Sigma}_0^H\right\}) = & \log(\Pr\{x_0\}) \\ & + \sum_{k=1}^{H} \log(\Pr\{y_k|x_k, \sigma_k\}) + \log(\Pr\{x_k|x_{k-1}, \sigma_k\}).\end{aligned} \tag{3.8}$$

Moreover, for the mode sequence it follows that

$$-\log\left(\Pr\left\{\hat{\Sigma}_0^H\right\}\right) = \ell_{\sigma_0} + \sum_{k=1}^{H} \ell_{\sigma_k, \sigma_{k-1}},$$

where $\ell_{\sigma_k,\sigma_{k-1}} := -\log(\rho_{\sigma_k,\sigma_{k-1}})$ and $\ell_{\sigma_0} := -\log(\rho_{\sigma_0})$. Expressions for the probabilities in (3.8) are obtained from the model (3.1) and distributions (3.3)

$$-\log(\Pr\{x_k|x_{k-1},\sigma_k\}) = \frac{1}{2}\left(\|w_{k-1}\|^2_{\mathcal{Q}^{-1}_{\sigma_k}} + \beta^w_{\sigma_k} + c_0\right), \tag{3.9a}$$

$$-\log(\Pr\{y_k|x_k,\sigma_k\}) = \frac{1}{2}\Big(\|y_k - C_{\sigma_k}x_k - D_{\sigma_k}u_k\|^2_{\mathcal{R}^{-1}_{\sigma_k}} + \beta^v_{\sigma_k} + c_1\Big), \tag{3.9b}$$

$$-\log\left(\Pr\{\hat{x}_0\}\right) = \frac{1}{2}\left(\|\hat{x}_0 - \bar{x}_0\|^2_{P_0^{-1}} + c_2\right), \tag{3.9c}$$

where $\beta^w_i := \log(|\mathcal{Q}_i|)$, $\beta^v_i := \log(|\mathcal{R}_i|)$ for $i \in S$ and $c_0 = c_2 = n\log(2\pi)$, $c_1 = r\log(2\pi)$, which are subsequently omitted since they do not depend on x_k and σ_k. By substituting these expressions into (3.7) and by defining the variables $\bar{w}_k = w_{k-1}$ and $\alpha_k = \hat{\sigma}_{k-1}$, the optimization problem becomes

$$\begin{aligned} J(\hat{x}_H,\hat{\sigma}_H) = \min_{\bar{W}_1^H,\alpha_1^H} \quad & \sum_{k=1}^{H} g(\hat{x}_k,\hat{\sigma}_k,\bar{w}_k,\alpha_k,y_k) \\ & + h(\hat{x}_0,\hat{\sigma}_0,\bar{x}_0), \\ \text{s.t.} \quad & \hat{x}_k = A_{\hat{\sigma}_k}\hat{x}_{k-1} + B_{\hat{\sigma}_k}u_{k-1} + G_{\hat{\sigma}_k}\bar{w}_k, \end{aligned} \tag{3.10}$$

where $\bar{W}_1^H := (\bar{w}_1,\ldots,\bar{w}_H)$, $\alpha_1^H := (\alpha_1,\ldots,\alpha_H)$ and

$$g(\hat{x}_k,\hat{\sigma}_k,\bar{w}_k,\alpha_k,y_k) = \frac{1}{2}\Big[\|\bar{w}_k\|^2_{\mathcal{Q}^{-1}_{\hat{\sigma}_k}} + \beta^w_{\hat{\sigma}_k} + \beta^v_{\hat{\sigma}_k} + \|y_k - C_{\hat{\sigma}_k}\hat{x}_k - D_{\hat{\sigma}_k}u_k\|^2_{\mathcal{R}^{-1}_{\hat{\sigma}_k}} + 2\ell_{\hat{\sigma}_k,\alpha_k}\Big], \tag{3.11}$$

$$h(\hat{x}_0,\hat{\sigma}_0,\bar{x}_0) = \ell_{\hat{\sigma}_0} + \frac{1}{2}\|\hat{x}_0 - \bar{x}_0\|^2_{P_0^{-1}}. \tag{3.12}$$

After the minimization one can find

$$(\hat{x}_H,\hat{\sigma}_H) = \arg\min_{\hat{x}_H,\hat{\sigma}_H} J(\hat{x}_H,\hat{\sigma}_H) \tag{3.13}$$

and iterate backwards for $k \in \{H-1,\ldots,1,0\}$

$$\hat{x}_k = A^{-1}_{\hat{\sigma}_{k+1}}\left(\hat{x}_{k+1} - B_{\hat{\sigma}_{k+1}}u_k - G_{\hat{\sigma}_{k+1}}\bar{w}_{k+1}\right), \tag{3.14}$$

$$\hat{\sigma}_k = \alpha_{k+1}, \tag{3.15}$$

to find the state and mode histories.

The optimal control problem in (3.10) can be solved using dynamic programming, which consists of the following steps:

1. Start with the *arrival cost* for $k = 0$, defined as

$$J_0(\hat{x}_0, \hat{\sigma}_0) = h(\hat{x}_0, \hat{\sigma}_0, \bar{x}_0) \tag{3.16}$$

2. For $k \in \{1, \dots, H\}$, compute the arrival cost

$$\begin{aligned} J_k(\hat{x}_k, \hat{\sigma}_k) = \min_{\bar{w}_k, \alpha_k} \{ & g(\hat{x}_k, \hat{\sigma}_k, \bar{w}_k, \alpha_k, y_k) \\ & + J_{k-1}(A^{-1}_{\hat{\sigma}_k}(\hat{x}_k - B_{\hat{\sigma}_k} u_{k-1} - G_{\hat{\sigma}_k} \bar{w}_k), \alpha_k) \}. \end{aligned} \tag{3.17}$$

Then J in (3.10) equals the arrival cost J_H. An expression for the cost function J_k can be found using the following proposition. Let $\mathcal{E}_k$ denote the set of all possible mode histories up to time k, i.e.

$$\mathcal{E}_k := \{\mathcal{T}_k^1, \mathcal{T}_k^2, \dots, \mathcal{T}_k^{n_{\mathcal{E}_k}}\}, \tag{3.18}$$

where $n_{\mathcal{E}_k} = s^k$ is the cardinality of the set $\mathcal{E}_k$, and

$$\mathcal{T}_k^i = \{\hat{\sigma}_0^i, \hat{\sigma}_1^i, \dots, \hat{\sigma}_{k-1}^i\}, \quad i \in \{1, \dots, s^k\}, \tag{3.19}$$

denotes the i^{th} possible mode history from time 0 to time $k-1$, where $\hat{\sigma}_k^j \in S$ for every $j \in \{1, \dots, s^k\}$.

Proposition 3.1. *The arrival cost $J_k(\hat{x}_k, \hat{\sigma}_k)$ is given by*

$$J_k(\hat{x}_k, \hat{\sigma}_k) = \min_{\mathcal{T} \in \mathcal{E}_k} \frac{1}{2} \left(\hat{x}_k - \hat{x}_{k|k}^{\mathcal{T}}\right)^{\intercal} \Pi_{\mathcal{T}} \left(\hat{x}_k - \hat{x}_{k|k}^{\mathcal{T}}\right) + \gamma_{\mathcal{T}}, \tag{3.20}$$

where for each $\mathcal{T} \in \mathcal{E}_k$, $\Pi_{\mathcal{T}} = \left(P_{k|k}\right)^{-1}$, $\gamma_{\mathcal{T}} = c_k$, $\hat{x}_{k|k}^{\mathcal{T}} = \hat{x}_{k|k}$ and $\hat{x}_{k|k}$, c_k, $P_{k|k}$ are obtained by iterating

$$\hat{x}_{0|0} = \bar{x}_0, \quad P_{0|0} = P_0, \quad c_0 = \ell_{\hat{\sigma}_0}, \tag{3.21a}$$

and, for $j \in \{0, \dots, k-1\}$,

$$\hat{x}_{j+1|j} = A_{\hat{\sigma}_{j+1}} \hat{x}_{j|j} + B_{\hat{\sigma}_{j+1}} u_j, \tag{3.21b}$$

$$P_{j+1|j} = A_{\hat{\sigma}_{j+1}} P_{j|j} A_{\hat{\sigma}_{j+1}}^{\intercal} + \mathcal{Q}_{\hat{\sigma}_{j+1}}, \tag{3.21c}$$

$$L_{j+1} = (\mathcal{R}_{\hat{\sigma}_{j+1}} + C_{\hat{\sigma}_{j+1}} P_{j+1|j} C_{\hat{\sigma}_{j+1}}^{\intercal})^{-1}, \tag{3.21d}$$

$$K_{j+1} = P_{j+1|j} C_{\hat{\sigma}_{j+1}}^{\intercal} L_{j+1}, \tag{3.21e}$$

$$e_{j+1} = y_{j+1} - C_{\hat{\sigma}_{j+1}} \hat{x}_{j+1|j} - D_{\hat{\sigma}_{j+1}} u_{j+1} \tag{3.21f}$$

$$\hat{x}_{j+1|j+1} = \hat{x}_{j+1|j} + K_{j+1} e_{j+1}, \tag{3.21g}$$

$$P_{j+1|j+1} = P_{j+1|j} - K_{j+1} C_{\hat{\sigma}_{j+1}} P_{j+1|j}, \tag{3.21h}$$

$$\begin{aligned} c_{j+1} = & \frac{1}{2} \|e_{j+1}\|^2_{L_{j+1}} + \ell_{\hat{\sigma}_{j+1}, \hat{\sigma}_j} \\ & + \frac{1}{2} \left(\beta^w_{\hat{\sigma}_{j+1}} + \beta^v_{\hat{\sigma}_{j+1}}\right) + c_j \end{aligned} \tag{3.21i}$$

The proof follows by establishing by induction that (3.16), (3.17) takes the form (3.20). It is omitted for the sake of brevity, however a similar proof can be found in [76].

Note that it follows from Proposition 3.1 that one can run a Kalman filter for each possible mode history in $\mathcal{E}_k$ and choose the one with the lowest cost to provide the state estimate. However, the cardinality of the set $\mathcal{E}_k$ grows exponentially with increasing k, which makes computing the optimal solution intractable for any reasonably large k.

Remark 3.2. *Note that the obtained expression for the cost-to-go requires the minimization over all possible mode histories and that by taking the minimum of the cost over all possible mode histories it only depends on $\hat{x}_k$ and $\hat{\sigma}_k$ and not on the actual or any other specific mode history.*

Remark 3.3. *Note that the cost* (3.20) *is not necessarily positive since β_i^w and β_i^v might be negative which may lead to negative $\gamma_{\mathcal{T}}$ obtained from* (3.21i). *However, the problem can always be scaled to enforce that β_i^w and β_i^v are positive for every $i \in S$, which leads to positive $\gamma_{\mathcal{T}}$ and positive cost* (3.20). *In fact, applying the transformation $z \to \zeta z$, $\zeta \in \mathbb{R}$, to all the variables x_k, w_k, u_k, $k \in N_0$ and y_k, v_k, $k \in N$ an equivalent JMAP problem is obtained, where* (3.1) *holds for the transformed variables but with $Q \to \mathbb{E}[\zeta w_k \zeta w_k^\intercal] = \zeta^2 Q$ and $R \to \mathbb{E}[\zeta v_k \zeta v_k^\intercal] = \zeta^2 R$ leading to $\beta_i^w \to \beta_i^w + 2n\log(\zeta)$ and $\beta_i^v \to \beta_i^v + 2r\log(\zeta)$. By choosing $\zeta > 1$ sufficiently large, it can be ensured that β_i^w and β_i^v are positive for every $i \in S$ and that the cost* (3.20) *is positive. Note that after using the proposed method, the obtained estimates can always be scaled back to the original variable dimensions.*

The previous remark justifies the following assumption, which can be made without loss of generality

Assumption 3.4. *For every $i \in S$, $\beta_i^w > 0$, $\beta_i^v > 0$ and thus $J_k(\hat{x}_k, \hat{\sigma}_k) > 0$, $\forall k$.*

3.4 Proposed approximate method

A technique from approximate dynamic control, known as relaxed dynamic programming [129], is employed in order to find an approximate solution to (3.20) . The idea of relaxed dynamic programming is to find simple functions to approximate J_k. In this chapter the following functions $V_0(\hat{x}_0, \hat{\sigma}_0) = J_0(\hat{x}_0, \hat{\sigma}_0)$ are considered, and, for $k \in \{1, \ldots, H\}$,

$$V_k(\hat{x}_k, \hat{\sigma}_k) = \min_{\mathcal{T} \in \mathcal{P}_k} \frac{1}{2} \left(\hat{x}_k - \hat{x}_{k|k}^{\mathcal{T}}\right)^\intercal \Pi_{\mathcal{T}} \left(\hat{x}_k - \hat{x}_{k|k}^{\mathcal{T}}\right) + \gamma_{\mathcal{T}}, \tag{3.22}$$

where $\mathcal{P}_k \subseteq \mathcal{E}_k$ can be seen as a pruned version of $\mathcal{E}_k$, where certain mode histories are discarded, thus reducing the complexity of the solution.

In order to achieve this, let

$$\mathcal{C}_k := \{(\mathcal{T}_{k-1}, \alpha_k) | \mathcal{T}_{k-1} \in \mathcal{P}_{k-1}, \alpha_k \in S\}, \tag{3.23}$$

denote the set containing all possible mode histories at time k considering the mode histories at $k-1$ that were not removed during pruning.

One first approach to prune, which would still ensure optimality, could be the following procedure:

Pruned Dynamic Programming procedure

1. Initialize $\mathcal{P}_k$ as empty.

2. Take the element (mode history) $\bar{\mathcal{T}}$ in $\mathcal{C}_k \setminus \mathcal{P}_k$ with the smallest $\gamma_{\bar{\mathcal{T}}}$ and check if it satisfies

$$\frac{1}{2}(x_k - \hat{x}_{k|k}^{\bar{\mathcal{T}}})^\intercal \Pi_{\bar{\mathcal{T}}}(x_k - \hat{x}_{k|k}^{\bar{\mathcal{T}}}) + \gamma_{\bar{\mathcal{T}}} \geq \min_{\mathcal{T} \in \mathcal{P}_k} \frac{1}{2}(x_k - \hat{x}_{k|k}^{\mathcal{T}})^\intercal \Pi_{\mathcal{T}}(x_k - \hat{x}_{k|k}^{\mathcal{T}}) + \gamma_{\mathcal{T}}, \, \forall x_k \in \mathbb{R}^n. \tag{3.24}$$

3. If (3.24) is not satisfied, then add the mode history $\bar{\mathcal{T}}$ of $\mathcal{C}_k$ to $\mathcal{P}_k$. If there are no more elements in $\mathcal{C}_k$, then stop, otherwise go to step 2.

Note that to facilitate step 2 the elements in $\mathcal{C}_k$ can be ordered initially with increasing values of $\gamma_{\mathcal{T}}$. This entails first putting the mode history corresponding to the lowest cost into the pruned set, after which for each remaining mode history it is checked if it is worse *for all* x_k. If this is the case, that specific mode history can be discarded while still ensuring that V_k will be equal to J_k. This follows by the principle of optimality (the tail of an optimal path is also optimal), which in this case translates to: if a mode sequence is redundant at step k of the dynamic programming algorithm, then it is a redundant tail of any mode sequence at steps $k+1, \ldots, H$.

Note that this procedure may not result in much pruning, since a mode history that is worse for all x_k may not exist. In order to *relax* this, the optimal arrival cost is no longer desired, but rather the set $\mathcal{P}_k$ is iteratively chosen in such a way that the approximated arrival cost function $V_k(\hat{x}_k, \hat{\sigma}_k)$ is always within a factor $\epsilon \geq 0$ of the optimal function $J_k(\hat{x}_k, \hat{\sigma}_k)$, for every $\hat{x}_k$ and $\hat{\sigma}_k$, i.e.,

$$J_k(\hat{x}_k, \hat{\sigma}_k) \leq V_k(\hat{x}_k, \hat{\sigma}_k) \leq (1+\epsilon) J_k(\hat{x}_k, \hat{\sigma}_k), \, \forall \hat{x}_k, \hat{\sigma}_k. \tag{3.25}$$

For (3.25) to make sense it needs to be ensured that the cost J_k is positive, which can be assumed without loss of generality as mentioned in Remark 3.3.

In order to choose the set $\mathcal{P}_k$ such that (3.25) holds, let

$$\begin{aligned} U_k(\hat{x}_k, \hat{\sigma}_k) &:= \min_{\alpha_k \in S} \min_{\bar{w}_k} (1+\epsilon) g(\hat{x}_k, \hat{\sigma}_k, \bar{w}_k, \alpha_k, y_k) \\ &\quad + V_{k-1}(A_{\hat{\sigma}_k}^{-1}(\hat{x}_k - B_{\hat{\sigma}_k} u_{k-1} - G_{\hat{\sigma}_k} \bar{w}_k), \alpha_k), \\ &= \min_{\mathcal{T} \in \mathcal{C}_k} \frac{1}{2} (\hat{x}_k - \bar{x}_{k|k}^{\mathcal{T}})^\intercal \bar{\Pi}_{\mathcal{T}} (\hat{x}_k - \bar{x}_{k|k}^{\mathcal{T}}) + \bar{\gamma}_{\mathcal{T}}, \end{aligned}$$

where for each $\mathcal{T} = (\hat{\sigma}_0, \ldots, \hat{\sigma}_{k-2}, \alpha_k) \in \mathcal{C}_k$, $\bar{\Pi}_{\mathcal{T}} = \left(\bar{P}_{k|k}\right)^{-1}$, $\bar{\gamma}_{\mathcal{T}} = \bar{c}_k$, $\bar{x}_{k|k}^{\mathcal{T}} = \bar{x}_{k|k}$ and $\bar{x}_{k|k}$, $\bar{c}_k$, $\bar{P}_{k|k}$ are obtained by computing for $j = k-1$

$$\bar{x}_{j+1|j} = A_{\hat{\sigma}_{j+1}} \hat{x}_{j|j} + B_{\hat{\sigma}_{j+1}} u_j, \tag{3.26a}$$

$$\bar{P}_{j+1|j} = A_{\hat{\sigma}_{j+1}} P_{j|j} A_{\hat{\sigma}_{j+1}}^\intercal + \psi^{-1} \mathcal{Q}_{\hat{\sigma}_{j+1}}, \tag{3.26b}$$

$$\bar{L}_{j+1} = (\psi^{-1} \mathcal{R}_{\hat{\sigma}_{j+1}} + C_{\hat{\sigma}_{j+1}} \bar{P}_{j+1|j} C_{\hat{\sigma}_{j+1}}^\intercal)^{-1}, \tag{3.26c}$$

$$\bar{K}_{j+1} = \bar{P}_{j+1|j} C_{\hat{\sigma}_{j+1}}^\intercal \bar{L}_{j+1}, \tag{3.26d}$$

$$\bar{e}_{j+1} = y_{j+1} - C_{\hat{\sigma}_{j+1}} \bar{x}_{j+1|j} - D_{\hat{\sigma}_{j+1}} u_{j+1} \tag{3.26e}$$

$$\bar{x}_{j+1|j+1} = \bar{x}_{j+1|j} + \bar{K}_{j+1} \bar{e}_{j+1}, \tag{3.26f}$$

$$\bar{P}_{j+1|j+1} = \bar{P}_{j+1|j} - \bar{K}_{j+1} C_{\hat{\sigma}_{j+1}} \bar{P}_{j+1|j}, \tag{3.26g}$$

$$\begin{aligned} \bar{c}_{j+1} = \frac{1}{2} \left\| \bar{e}_{j+1} \right\|_{\bar{L}_{j+1}}^2 + \psi \ell_{\hat{\sigma}_{j+1}, \hat{\sigma}_j} \\ + \frac{1}{2} \psi \left(\beta_{\hat{\sigma}_{j+1}^i}^w + \beta_{\hat{\sigma}_{j+1}^i}^v \right) + c_j \end{aligned} \tag{3.26h}$$

with $\psi = (1+\epsilon)$ and $\hat{x}_{j|j}$, $P_{j|j}$, c_j are obtained by iterating the recursion (3.21a)-(3.21i) for $j \in \{0, \ldots, k-2\}$. The function U_k coincides with V_k when $\epsilon = 0$. However, this function is defined with an extra cost term for the running cost g when $\epsilon > 0$.

At each timestep k, the set $\mathcal{P}_k$ is a pruned version of the set $\mathcal{C}_k$ obtained as follows:

Relaxed Dynamic Programming procedure

1. Initialize $\mathcal{P}_k$ as empty.
2. Take the element (mode history) $\bar{\mathcal{T}}$ in $\mathcal{C}_k \setminus \mathcal{P}_k$ with the smallest $\gamma_{\bar{\mathcal{T}}}$ and check if it satisfies

$$\begin{aligned} &\frac{1}{2} (x_k - \bar{x}_{k|k}^{\bar{\mathcal{T}}})^\intercal \bar{\Pi}_{\bar{\mathcal{T}}} (x_k - \bar{x}_{k|k}^{\bar{\mathcal{T}}}) + \bar{\gamma}_{\bar{\mathcal{T}}} \geq \\ &\qquad \min_{\mathcal{T} \in \mathcal{P}_k} \frac{1}{2} (x_k - \hat{x}_{k|k}^{\mathcal{T}})^\intercal \Pi_{\mathcal{T}} (x_k - \hat{x}_{k|k}^{\mathcal{T}}) + \gamma_{\mathcal{T}}, \, \forall x_k \in \mathbb{R}^n. \end{aligned} \tag{3.27}$$

3. If (3.27) is not satisfied, then add the mode history $\bar{\mathcal{T}}$ of $\mathcal{C}_k$ to $\mathcal{P}_k$. If there are no more elements in $\mathcal{C}_k$, then stop, otherwise go to step 2.

This can be viewed as performing the same procedure as in (3.24), but using the cost including the $(1+\epsilon)$ factor calculated using (3.26a)-(3.26h). The next result shows that this procedure guarantees that (3.25) is met.

Theorem 3.5. *Suppose that Assumption 3.4 holds and let V_k be defined by* (3.22) *with the set $\mathcal{P}_k$ obtained from the procedure* (3.27)*. Then* (3.25) *holds.*

Proof. Using induction, $V_0(\hat{x}_0, \hat{\sigma}_0) = J_0(\hat{x}_0, \hat{\sigma}_0)$ and assuming

$$V_{k-1}(\hat{x}_{k-1}, \hat{\sigma}_{k-1}) \leq (1+\epsilon) J_{k-1}(\hat{x}_{k-1}, \hat{\sigma}_{k-1})$$

then

$$\begin{aligned} U_k(\hat{x}_k, \hat{\sigma}_k) \leq & \min_{\alpha_k \in S} \min_{\bar{w}_k} (1+\epsilon) g(\hat{x}_k, \hat{\sigma}_k, \bar{w}_k, \alpha_k, y_k) \\ & + (1+\epsilon) J_{k-1}(A_{\hat{\sigma}_k}^{-1}(\hat{x}_k - B_{\hat{\sigma}_k} u_{k-1} - G_{\hat{\sigma}_k} \bar{w}_k), \alpha_k) \\ = & (1+\epsilon) J_k(\hat{x}_k, \hat{\sigma}_k) \end{aligned}$$

It is clear, from the definition of $U_k(\hat{x}_k, \hat{\sigma}_k)$, that

$$\min_{\mathcal{T} \in \mathcal{C}_k} \quad \frac{1}{2} (\hat{x}_k - \hat{x}_{k|k}^{\mathcal{T}})^{\intercal} \Pi_k^{\mathcal{T}} (\hat{x}_k - \hat{x}_{k|k}^{\mathcal{T}}) + \gamma_{\mathcal{T}} \leq U_k(\hat{x}_k, \hat{\sigma}_k)$$

and due to the inclusion of the element with the lowest cost in $\mathcal{C}_k$ into $\mathcal{P}_k$ according to (3.27), the set $\mathcal{C}_k$ can be replaced by the set $\mathcal{P}_k$ in the last inequality, obtaining

$$V_k(\hat{x}_k, \hat{\sigma}_k) \leq U_k(\hat{x}_k, \hat{\sigma}_k) \leq (1+\epsilon) J_k(\hat{x}_k, \hat{\sigma}_k)$$

which concludes the proof. ■

This implies that at each time step the joint log-likelihood of the state and mode sequence estimate is within a factor ϵ of the optimal joint log-likelihood. If this procedure is conducted for $\epsilon = 0$ the procedure in (3.24) is obtained and the optimal sequence is guaranteed, but the complexity (number of hypotheses or cardinality of $\mathcal{P}_k$) grows exponentially with time k. This will also be the case for sufficiently small ϵ, making it impossible to bound the complexity of the method for a general ϵ. However, simulations (see Section 3.5) show that when ϵ is sufficiently large the number of hypotheses remains bounded and small even for large time spans, as also reported in [129]. It is, however, hard to provide results on if this is the case in general and, if so, how large ϵ should be.

Testing (3.27) might in general be hard. Alternatively, a tighter bound is proposed, resulting in less pruning, but simpler to test. This test is given in terms of the following linear matrix inequalities (LMIs)

$$\exists \tau_j \quad \geq \quad 0, \text{ such that } \sum_{j=1}^{n_{\mathcal{P}_k}} \tau_j \quad = \quad 1 \text{ and } \bar{Z} \quad \geq \quad \sum_{j=1}^{n_{\mathcal{P}_k}} \tau_j Z_j \quad (3.28)$$

where $n_{\mathcal{P}_k}$ denotes the cardinality of the set $\mathcal{P}_k$ and

$$\bar{Z} = \begin{bmatrix} \bar{\Pi}_{\bar{\mathcal{T}}} & -\bar{\Pi}_{\bar{\mathcal{T}}} \bar{x}^{\bar{\mathcal{T}}}_{k|k} \\ -(\bar{x}^{\bar{\mathcal{T}}}_{k|k})^\intercal \bar{\Pi}_{\bar{\mathcal{T}}} & (\bar{x}^{\bar{\mathcal{T}}}_{k|k})^\intercal \bar{\Pi}_{\bar{\mathcal{T}}} \bar{x}^{\bar{\mathcal{T}}}_{k|k} + 2\bar{\gamma}_{\bar{\mathcal{T}}} \end{bmatrix},$$
$$Z_j = \begin{bmatrix} \Pi_{\mathcal{T}_j} & -\Pi_{\mathcal{T}_j} \hat{x}^{\mathcal{T}_j}_{k|k} \\ -(\hat{x}^{\mathcal{T}_j}_{k|k})^\intercal \Pi_{\mathcal{T}_j} & (\hat{x}^{\mathcal{T}_j}_{k|k})^\intercal \Pi_{\mathcal{T}_j} \hat{x}^{\mathcal{T}_j}_{k|k} + 2\gamma_{\mathcal{T}_j} \end{bmatrix},$$

with $\mathcal{T}_j$ the sequences in $\mathcal{P}_k$, i.e., $\mathcal{P}_k = \{\mathcal{T}_1, \ldots, \mathcal{T}_{n_{\mathcal{P}_k}}\}$. If this new check holds then (3.27) holds (the converse is not in general true). This can be quickly seen by noting that if (3.28) holds, then, for every $z \in \mathbb{R}^{n+1}$

$$z^\intercal \bar{Z} z \geq z^\intercal \sum_{j=1}^{n_{\mathcal{P}_k}} \tau_j Z_j z \geq \min_j z^\intercal Z_j z$$

must hold and therefore, considering $z = \begin{bmatrix} x_k^\intercal, & 1 \end{bmatrix}^\intercal$, it is concluded that (3.27) must hold as well.

This procedure amounts to checking $n_{\mathcal{P}_k}$ LMIs for each element in $\mathcal{C}_k$, which in the worst case results in a total of $\frac{1}{2}(n^2_{\mathcal{C}_k} - n_{\mathcal{C}_k})$ LMIs to be checked at each timestep, which might be computationally hard to do online. A simpler alternative is to search for a single mode sequence that has less cost for all x_k. This entails checking if

$$\exists i \in \{1, \ldots, n_{\mathcal{P}_k}\}, \text{ s.t. } \bar{Z} \geq Z_i, \tag{3.29}$$

which is easily verified, but could result in an (unnecessarily) large amount of hypotheses being maintained.

Remark 3.6. *Note that the guarantee* (3.25) *in Theorem 1 concerns the negative log-likelihood as in* (3.7) *and not the likelihood itself as in* (3.6)*. In order to translate* (3.25) *to a guarantee in terms of the likelihood, let*

$$L^* = \max \Pr\left\{Y_1^H, \hat{X}_0^H | \hat{\Sigma}_0^H\right\} \Pr\left\{\hat{\Sigma}_0^H\right\}$$

be the right hand side of (3.6) *corresponding to the optimal cost J and let L be the value obtained by the proposed relaxed dynamic programming method corresponding to V. The costs are then $J = -\log(L^*) - c$ and $V = -\log(L) - c$, where $c = c_0 + c_1 + c_2$ is a positive constant due to the terms neglected in* (3.9)*. Then, the method guarantees*

$$-\log(L^*) - c \leq -\log(L) - c \leq (1+\epsilon)(-\log(L^*) - c),$$

which implies $L^{*^{1+\epsilon}} e^{\epsilon c} \leq L \leq L^*$ (Ass. 3.4 implies $L^* < 1$).

3.5 Simulations

In order to demonstrate the effectiveness of the proposed approach, two examples are considered: a random walk model subject to measurement failures and the tracking of multiple targets[1].

3.5.1 Example 1: System subject to measurement failures

Consider a scalar random walk model subject to measurement failures given by

$$\begin{aligned} x_{k+1} &= x_k + w_k \\ y_k &= x_k + H_{\sigma_k} v_k \end{aligned}$$

where $\sigma_k \in S = \{1, 2\}$, $H_1 = 20$, $H_2 = 1$, and the mode sequence probabilities follow probability transition matrix $\Gamma = [\rho_{ij}]$ with $\rho_{11} = 0.15$, $\rho_{12} = 0.4$, $\rho_{21} = 0.85$, $\rho_{22} = 0.6$.

For this model $N_{MC} = 50$ Monte-Carlo simulations are performed, each with a horizon of $H = 12$. The proposed methods using the LMI check in (3.28) (LMI) and using the loosened check of (3.29) (PosDef) are compared to the expectation maximization (EM) algorithm adapted from [130] and the time varying KF algorithm using the true mode sequence (KF) to the optimal cost for varying values of ϵ. The results are shown in Figure 3.1, where from the top figure it can be observed that the proposed methods manage to find the optimal cost for $\epsilon = 0$ as expected from (3.25) and that they perform better than both the EM and the KF algorithms. Moreover, they achieve the same cost, which stays well below the $(1+\epsilon)$ bound. In the bottom figure it becomes clear that the loosened check of (3.29) only retains more hypotheses for low ϵ, and by looking at the computation times displayed in Figure 3.2 it is concluded that for this problem the larger amount of hypotheses retained does not outweigh the reduced computation times.

3.5.2 Example 2: Multiple model tracking (MMT)

Consider a discretized version of a target moving in a 2 dimensional (x, y) space according to

$$\begin{aligned} x_{k+1} &= A_{\sigma_{k+1}} x_k + w_k, \\ y_k &= C x_k + v_k, \end{aligned}$$

where the state is given by $x = [p_\mathrm{x}, v_\mathrm{x}, p_\mathrm{y}, v_\mathrm{y}]^\intercal$, with p_x, p_y and v_x, v_y denoting the position and velocity in (x, y) direction, respectively. The measurements are the position in x and y, so that $C = [e_1^\intercal \, e_3^\intercal]^\intercal$, $e_1 = [1\,0\,0\,0]$, $e_3 = [0\,0\,1\,0]$ and $w_k \in \mathcal{N}(0, 0.001I)$, $v_k \in \mathcal{N}(0, 0.01I)$. For the motion the model is considered

[1] All files used for the simulation results are available at https://github.com/aandrien/relaxed-JMAP.

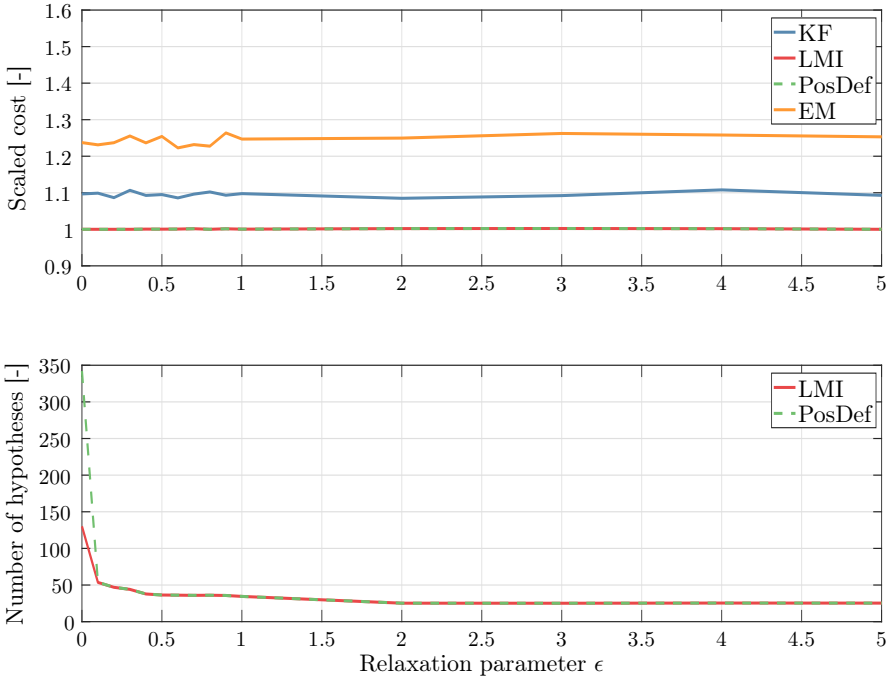


Figure 3.1: (top) The cost of several algorithms scaled to the optimal cost. (bottom) The sum of the number of hypotheses that are retained at each time step by the proposed two methods.

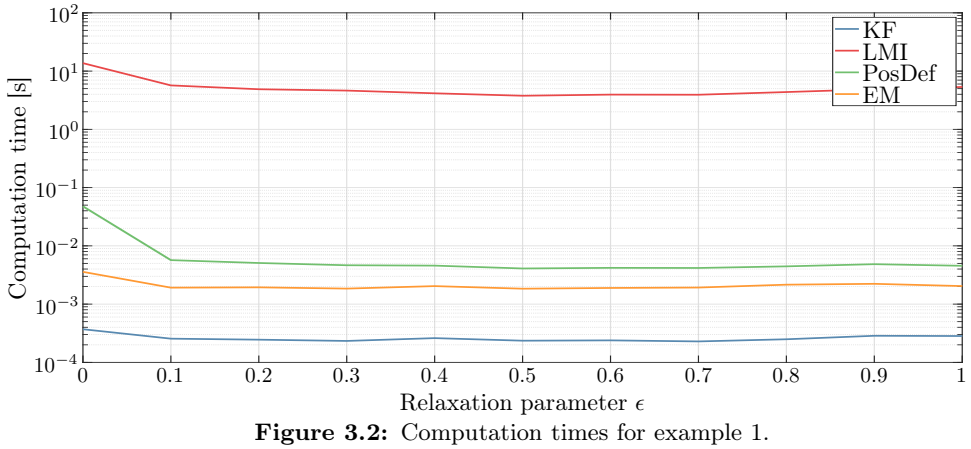


Figure 3.2: Computation times for example 1.

to have three operating modes $\sigma_k \in S = \{1, 2, 3\}$ and the modes consist of a constant velocity, positive constant turn rate ω and negative constant turn rate

Table 3.1: RMS and mode identification results for example 2

Algorithm	p_x [m]	v_x [m/s]	p_y [m]	v_y [m/s]	Mode [%]
KF	0.0413	0.0771	0.0411	0.0764	100
PosDef	0.0467	0.1812	0.0462	0.1813	67.9
EM	0.0613	0.3437	0.0614	0.3440	38.2
IMM	0.0804	0.3969	0.0792	0.3916	48.0

$-\omega$ model, respectively. For the constant velocity model,

$$A_1 = \begin{bmatrix} A_{CV} & 0_{2\times 2} \\ 0_{2\times 2} & A_{CV} \end{bmatrix}, \quad A_{CV} = \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix},$$

where $\tau = 0.1$ is the sample time and for the constant turn rate model, $A_2 = A_{CT}(-\omega)$, $A_3 = A_{CT}(\omega)$, where

$$A_{CT}(\omega) = \begin{bmatrix} 1 & \frac{\sin(\omega\tau)}{\omega} & 0 & -\frac{1-\cos(\omega\tau)}{\omega} \\ 0 & \cos(\omega\tau) & 0 & -\sin(\omega\tau) \\ 0 & \frac{1-\cos(\omega\tau)}{\omega} & 1 & \frac{\sin(\omega\tau)}{\omega} \\ 0 & \sin(\omega\tau) & 0 & \cos(\omega\tau) \end{bmatrix},$$

The transition probability matrix and initial probabilities are

$$\Gamma = \begin{bmatrix} 0.85 & 0.1 & 0.05 \\ 0.1 & 0.8 & 0.1 \\ 0.05 & 0.1 & 0.85 \end{bmatrix}, \qquad \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \end{bmatrix} = \begin{bmatrix} 0.6 \\ 0.2 \\ 0.2 \end{bmatrix}.$$

The other parameters are $\epsilon = 0.5$, $\omega = 2$ and a scaling of $\zeta = 100$ was used to have a positive cost.

This model is simulated for $N_{MC} = 500$ Monte-Carlo simulations, each with a length of $H = 100$, comparing the estimation errors for the proposed algorithm using the check in (3.29), a time-varying Kalman filter using the true mode sequence, the expectation maximization algorithm and the interactive multiple model (IMM) filter [17]. Although the last algorithm only provides estimates of the current state, as opposed to both the state and mode sequence over the whole horizon, it is used in this comparison because it is commonly used for MMT. An instance of one of the simulations is shown in Figure 3.3. The results are displayed in Table 3.1, showing the RMS errors of all 4 states for each algorithm, as well as the mode identification results, displayed as a percentage of correct estimates of the true mode. From these it becomes clear that the proposed method outperforms both the EM and IMM algorithms, having a smaller RMS error for each state as well as a larger percentage of correctly identified modes. The performance of the proposed methods is similar to the KF with respect to position, but is significantly worse for velocity. This is because the

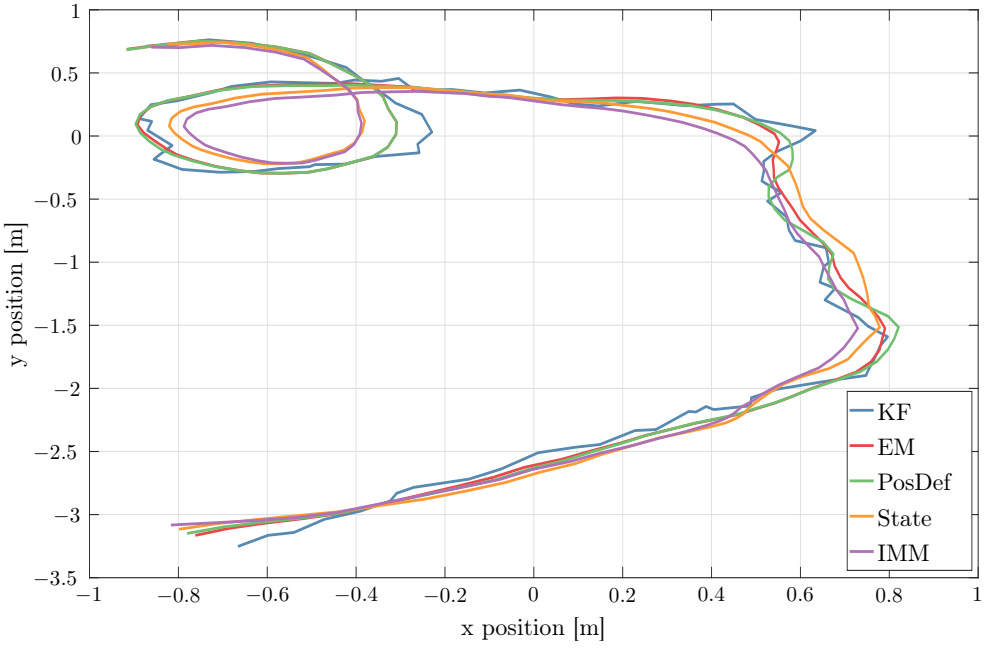


Figure 3.3: Position estimate results for example 2.

measurements cover only position, so that incorrect identification of the mode results in incorrect prediction, which influences mainly the velocity estimates as they are not directly corrected by the measurements.

3.6 Conclusions & Future Work

In this chapter novel methods for obtaining the JMAP estimate of the state and mode sequence of MJLS have been presented that use relaxed dynamic programming. By employing RDP the estimates are guaranteed to be within an ϵ bound of the optimal joint log-likelihood. Furthermore, provided two methods to perform the pruning have been provided: one based on LMIs and a simpler, and often faster, check that prunes based on the positive definiteness of a matrix. Simulation results show the effectiveness of both methods on two examples as well as the advantage of the method compared to the well-known expectation maximization technique.

Future work includes the investigation of the robustness of the method against uncertainties in the transition probabilities and exploring other methods of pruning the hypotheses.

Chapter 4

System Identification for Markov Jump Linear Systems

This chapter tackles the problem of identifying the parameters of a class of stochastic switched systems, where the active subsystem is determined by a Markov chain. This class includes autoregressive models with exogenous inputs (ARX) for which the parameters switch according to a Markov chain and general Markov Jump Linear Systems (MJLSs) with full-state information. The transition probabilities of the Markov chain are assumed to be known, but the active subsystem is unknown. A recursive identification method for the joint maximum a posteriori probability estimate of these parameters and of the unknown mode is proposed relying on relaxed dynamic programming. The method is guaranteed to provide an estimate whose joint posteriori probability is within a constant factor of that of the optimal estimate while reducing the computational complexity. The advantages and disadvantages of this method with respect to the standard Expectation-Maximization (EM) algorithm are discussed and a numerical example is provided.

This chapter is based on [7]

4.1 Introduction

Switched (and hybrid) systems [126] extend traditional linear and non-linear systems and are characterized by a set of subsystems among which the system switches abruptly during operation. Extensive research has been conducted on these systems in recent decades. One of the considered challenges has been system identification. Several identification techniques are now available that consider broad classes of systems and more specific ones. See, for example, [36, 101, 163, 27, 93] and [118], for a recent book on this subject. A particularly interesting class is that of switched autoregressive systems with exogenous inputs (sARX), where the system parameters switch arbitrarily among a finite set of values [163, 202]; the active subsystem (mode) is assumed not to be known (otherwise traditional techniques can be used to identify the subsystems). Recursive identification methods, interesting in many contexts such as adaptive control and well-established for traditional ARX, are also available for sARX [202].

When the switching is governed by a Markov chain, these systems are referred to as Markovian Jump systems [221], also referred to here as Markov switched systems. A prominent subclass, considering linear dynamics, are Markov Jump Linear systems (MJLS) [46]. They appear often in applications such as networked control systems [12], econometrics [83], chemistry [37], among many others. The identification of Markov switched systems with unknown mode can rely on the mentioned methods for the identification of switched systems with arbitrary switching [36, 163, 202, 118], which ignore the probabilistic knowledge of the switching mechanism. However, as shown for instance in [148], by taking this knowledge into account, more efficient identification methods can naturally be obtained.

The literature on the identification of Markov switched systems is tied in with the Expectation Maximization (EM) algorithm, a suboptimal approach for maximum likelihood (or maximum a posteriori probability) problems with unobserved variables. For some classes of systems analytic solutions for parameter identification can be found by using the EM algorithm. These include:

(A) switched ARX with switching governed by a Markov chain, see [99, 37] for the EM analytic expressions.

(B) non-linear Markov jump linear systems with subsystem dynamics that affinely depend on a set of known state and control dependent features and the coefficients of the affine combination are unknown; the expressions for the EM algorithm without considering external control inputs can be found in [18].

(C) Markov jump linear systems with full state information, which are special cases of (B).

Only a few exceptions in the literature do not rely on the EM algorithm when considering these models, see, e.g., [92]. For more general classes including MJLS with partial information [16, 15, 189, 41], auto-regressive moving-average with exogenous inputs (ARMAX) systems [25], non-linear Jump Markov systems [164], one must rely on numerical alternatives to the EM algorithm such as Gibbs sampling or more generally Monte-Carlo Markov chain (MCMC) methods. The transition matrix of the Markov chain is sometimes considered to be unknown and in other cases known.

While it is remarkable that the EM algorithm can provide an analytic solution for the mentioned classes (A, B, C), it has some shortcomings:

(i) It can get stuck in local maxima of the objective function (e.g., log-likelihood); this is a known and shared problem with the usual application of the EM algorithm in the context of finite mixture models [145] and results from the fact that the objective functions are typically multi-modal. A related issue is that the local maximum the algorithm converges to depends on parameter initialization.

(ii) It is not recursive as opposed to for instance the methods for sARX systems with arbitrary switching (see [202] as mentioned before), hampering their use in applications such as adaptive control.

The literature on handling these issues is scarce. Shortcoming (i) can be mitigated by considering the maximum a posteriori probability estimator and a good prior function on the parameters rather than the maximum likelihood estimator [18]. In turn, [74] addresses this shortcoming by devising an algorithm that iteratively learns the location of local maxima and focuses the search away from these local maxima. To the best of my knowledge, there are currently no recursive algorithms to the problem at hand. These shortcomings motivate the present paper which proposes a different method for the identification of models (A), (B), and (C).

First, a problem formulation is proposed, which encompasses the identification problems for models (A), (B) and (C), with known Markov chain transition probabilities. The goal is to find the joint maximum a posteriori probability estimator of the unknown parameters and unknown modes. However, as it will turn out, the complexity of computing this joint MAP estimator grows exponentially with the data size.

Motivated by this, a recursive method is proposed that relies on relaxed dynamic programming (see [129]), which is guaranteed to provide an estimate whose joint posteriori probability is within a constant factor of that of the optimal estimate while reducing the computational complexity. This method is borrowed from Chapter 3 and the connection will be clarified next.

The proposed method, relying on dynamic programming, is inspired by the method proposed in Chapter 3, although in the present chapter and in Chapter 3, the problems tackled are quite different. In fact, here we tackled the problem of identifying the parameters of a Markov jump linear system whereas in Chapter 3

we tackled the problem of estimating the state of a Markov jump linear system based on the output and with unknown Markov chain state. It is well-known, for standard linear time-invariant (LTI) systems represented as standard ARMAX models, that parameter identification can be carried out with Kalman filtering, which is a standard method for state estimation for LTI systems [106]. In fact, one can see the parameters of LTI systems as unknown dynamical state variables with trivial dynamics and formulate a state estimation problem to estimate these. One of the strong advantages of this method with respect to others (such as the Maximum Likelihood method) is that this leads to a recursive method for parameter estimation which can run online. We resort to a similar line of thought in this chapter and by leveraging on the results proposed in the previous chapter, we propose a new method for the system identification of a Markov jump linear system. A variant of the method proposed in Chapter 3, which can still provide guarantees on the joint posteriori probability, is also provided. Although these guarantees are much less tight, this variant shows superior performance in the provided numerical example. By providing global optimality guarantees, these methods depart from the EM algorithm, which constructs a sequence of cost improving approximations, which are prone to get stuck in local optima.

On the other hand, for the proposed method the assumption that the Markov chain transition probabilities are known is needed, which is not the case for the EM algorithm. The shortcoming of relaxed dynamic programming regarding the difficulty in predicting upfront the complexity of the method is also inherited, although in many applications it stays within acceptable levels.

Through a numerical example the effectiveness of the proposed method is shown.

The remainder of the paper is organized as follows. Section 4.2 proposes a problem formulation which encompasses the identification problems for models (A), (B) and (C) and formulates the joint MAP parameter estimation problem. Section 4.3 provides the optimal solution to the joint MAP estimation problem, showing that it is in general computationally infeasible to obtain. Section 4.4 introduces the proposed methods relying on relaxed dynamic programming. Section 4.5 discusses a numerical example and Section 4.6 provides some final remarks and conclusions.

Notation: I_n denotes the $n \times n$ identity matrix. For vectors $u_1, \ldots, u_n$, $(u_1, \ldots, u_n) := \begin{bmatrix} u_1^\intercal & \cdots & u_n^\intercal \end{bmatrix}^\intercal$. Vector $\mathsf{e}_i \in \mathbb{R}^m$ denotes the ith element of the canonical basis, i.e., it is zero except at entry i where it is one. The notation $w \sim \mathcal{N}(v, R)$ indicates that w is Gaussian distributed with mean v and covariance R. For a symmetric matrix M and a vector v, $|M|$ is the determinant and $\|x\|_M := \sqrt{x^\intercal M x}$. The Kronecker product between A and B is denoted by $A \otimes B$.

4.2 Problem formulation

In this section it is shown that the identification problem for the three models (A), (B), (C) mentioned in the introduction can be written in terms of the following problem. Consider the following model

$$\zeta_t = \Gamma_{\sigma_t,t}\theta + w_t, \quad t \geq 0, \tag{4.1}$$

where $\theta \in \mathbb{R}^{n_\theta}$ is a vector of parameters to be estimated and $\sigma_t \in \mathcal{S} := \{1, 2, \ldots, m\}$ is a Markov chain with transition probabilities

$$p_{ij} = \text{Prob}[\sigma_{t+1} = j | \sigma_t = i].$$

The vectors $\zeta_t \in \mathbb{R}^{n_\zeta}$ are available for estimating the parameters. Moreover, the gains $\Gamma_{i,t}$, switching according to the Markov chain $i = \sigma_t$, can be decomposed as

$$\Gamma_{i,t} = \mathsf{e}_i^\intercal \otimes x_t^\intercal$$

for some vectors $x_t \in \mathbb{R}^{n_x \times n_\zeta}$ available for estimating the parameters, where $\mathsf{e}_i \in \mathbb{R}^m$, $n_\theta = m \times n_x$. The noise variables $w_t \in \mathbb{R}^{n_\zeta}$ are independent Gaussian random variables with zero mean and covariance that depends on the active mode at time t, $w_t \sim \mathcal{N}(0, R_{\sigma_t})$, where R_i are assumed to be positive definite. The p_{ij} are assumed to be known together with the initial probabilities $p_i^0 := \Pr\{\sigma_0 = i\}$, $i \in \{1, \ldots, m\}$. The problem of interest is to estimate θ from the data set

$$\mathcal{D} = \{(\zeta_t, x_t) | t \in \{0, 1, \ldots, H\}\},$$

where H is the size of the data set. Note that the Markov chain mode is not available in this data set.

The models (A), (B), and (C) can be written in the general form (4.1), as shown in Sections 4.2.1, 4.2.2, 4.2.3, respectively. In Section 4.2.4, the joint MAP estimation problem is formulated.

4.2.1 sARX

Consider the following autoregressive model with exogenous inputs (ARX models):

$$y_t = \sum_{j=1}^{q} a_j^{\sigma_t} y_{t-j} + \sum_{\ell=1}^{r} b_\ell^{\sigma_t} u_{t-\ell} + w_t \tag{4.2}$$

for which the unknown parameters $(a_1^i, \ldots, a_q^i, b_1^i, \ldots, b_r^i)$, $i \in \{1, 2, \ldots, m\}$ switch according to a Markov chain σ_t identical to the one described before. Let us first assume that $y_t \in \mathbb{R}$ and $u_t \in \mathbb{R}$. Suppose that the following data

$$\begin{aligned}\mathcal{D}_A = &\{(y_t) | t \in \{-q+1, \ldots, 0, 1, \ldots, H\}\} \\ &\cup \{(u_t) | t \in \{-r+1, \ldots, 0, 1, \ldots, H\}\}\end{aligned}$$

is available from experiments. The goal is to estimate the parameters

$$\theta = (a^1, b^1, a^2, b^2, \ldots, a^m, b^m)$$

with $a^i = (a^i_1, \ldots, a^i_q)$ and $b^i = (b^i_1, \ldots, b^i_r)$ from the data in $\mathcal{D}_A$.

Let $\zeta_t = y_t$ and

$$x_t^\intercal = \begin{bmatrix} y_{t-1} & \cdots & y_{t-q} & u_{t-1} & \cdots & u_{t-r} \end{bmatrix}.$$

Then (4.2) can be rewritten as (4.1) with

$$\Gamma_{i,t} = \mathsf{e}_i^\intercal \otimes x_t^\intercal$$

Suppose now that $y_k \in \mathbb{R}^{n_y}$, $u_k \in \mathbb{R}^{n_u}$, possibly for $n_y \geq 1$ and $n_u \geq 1$ and the coefficients a^i_j, b^i_j in (4.2) are matrices rather than scalars; by redefining

$$\theta = (\nu(\bar{a}^1), \nu(\bar{b}^1), \nu(\bar{a}^2), \nu(\bar{b}^2), \ldots, \nu(\bar{a}^m), \nu(\bar{b}^m))$$

where for $A = [a_1, a_2, \cdots, a_n]$, $\nu(A) = [a_1^\intercal, a_2^\intercal, \cdots, a_n^\intercal]^\intercal$, $\bar{a}^i = [a^i_1, \cdots, a^i_q]$, $\bar{b}^i = [b^i_1, \cdots, b^i_q]$ and

$$\Gamma_{i,t} = e_i^\intercal \otimes x_t^\intercal$$

with

$$x_t^\intercal = [[y_{t-1}^\intercal \cdots y_{t-q}^\intercal] \otimes I_{n_y q} \quad [u_{t-1}^\intercal \cdots u_{t-r}^\intercal] \otimes I_{n_u r}]$$

(4.1) can still be obtained, where the fact that $\nu(ABC) = (C^\intercal \otimes A)\nu(B)$ was used.

4

4.2.2 Jump Markov Non-linear systems

Consider the following class of Jump Markov Non-linear systems

$$\xi_{t+1} = \phi_0(\xi_t, u_t) + \sum_{\ell=1}^{p} \alpha_\ell^{\sigma_t} \phi_\ell(\xi_t, u_t) + w_t \tag{4.3}$$

where $\xi_t \in \mathbb{R}^{n_\xi}$ is the state, $u_t \in \mathbb{R}^{n_u}$ is the control input,

$$\theta := (\alpha^1, \alpha^2, \ldots, \alpha^m),$$

with $\alpha^i = (\alpha^i_1, \ldots, \alpha^i_p)$, $\alpha^i_j \in \mathbb{R}$ is the set of parameters to be estimated, and $\phi_\ell : \mathbb{R}^{n_\xi} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_\xi}$ are known functions. Suppose now that the data available from experiments contains the full state, besides the control input

$$\mathcal{D}_B = \{(\xi_t, u_t) | t \in \{0, 1, \ldots, H\}\}.$$

Then, let $\zeta_t = \xi_{t+1} - \phi_0(\xi_t, u_t)$, $x_{t,\ell} = \phi_\ell(\xi_t, u_t)$, $x_t^\intercal = \begin{bmatrix} x_{t,1} & \ldots & x_{t,p} \end{bmatrix}$ to obtain (4.1) with

$$\Gamma_{i,t} = e_i^\intercal \otimes x_t^\intercal.$$

4.2.3 Markov Jump Linear Systems

Consider a general MJLS

$$\xi_{t+1} = A_{\sigma_t}\xi_t + B_{\sigma_t}u_t + w_t$$

where $\xi_t \in \mathbb{R}^{n_\xi}$ and $u_t \in \mathbb{R}^{n_u}$. Suppose that the matrices A_i and B_i are unknown or a subset of entries are unknown. In general, one can write

$$A_i = \Phi^0 + \sum_{j=1}^{a} \gamma^{i,j}\Phi^j, \quad B_i = \Theta^0 + \sum_{j=1}^{b} \beta^{i,j}\Theta^j$$

for constant matrices Φ^j and Θ^j and parameters $\gamma^{i,j}$, $\beta^{i,j}$. Then the MJLS is written as (4.3) with

$$\phi_0(\xi_t, u_t) = \Phi^0\xi_t + \Theta^0 u_t$$

$$\phi_\ell(\xi_t, u_t) = \Phi^\ell \xi_t, \quad 1 \le \ell \le a$$
$$\phi_\ell(\xi_t, u_t) = \Theta^{\ell-a} u_t, \quad a+1 \le \ell \le a+b$$

and $\alpha_j^i = \gamma^{i,j}$ for $1 \le j \le a$ and $\alpha_j^i = \beta^{i,j-a}, a+1 \le j \le a+b$.

Likewise if the MJLS also contains output equations

$$\bar{\xi}_{t+1} = \bar{A}_{\sigma_t}\bar{\xi}_t + \bar{B}_{\sigma_t}u_t + \bar{w}_t$$
$$y_t = \bar{C}_{\sigma_t}\bar{\xi}_t + \bar{D}_{\sigma_t}u_t + \bar{v}_t$$

one can write $\xi_t = \begin{bmatrix}\bar{\xi}_t^\intercal & y_t^\intercal\end{bmatrix}^\intercal$, and $w_t = \begin{bmatrix}\bar{w}_t^\intercal & \bar{v}_t^\intercal\end{bmatrix}^\intercal$

$$\xi_{t+1} = \underbrace{\begin{bmatrix}\bar{A}_{\sigma_t} & 0\\ \bar{C}_{\sigma_t} & 0\end{bmatrix}}_{A_{\sigma_t}}\xi_t + \underbrace{\begin{bmatrix}\bar{B}_{\sigma_t}\\ \bar{D}_{\sigma_t}\end{bmatrix}}_{B_{\sigma_t}}u_t + w_t$$

and obtain a similar formulation.

4.2.4 Maximum a posteriori probability (MAP) estimation problem

In the maximum a posteriori probability (MAP) framework, the goal is to provide both an estimate $\hat{\theta}$ of the parameters θ as well as an estimate $\hat{\sigma}_{0:H} := [\hat{\sigma}_0, \ldots, \hat{\sigma}_H]^\intercal$ of the mode sequence $\sigma_{0:H} := [\sigma_0, \ldots, \sigma_H]^\intercal$, that have the maximum likelihood, given data $\mathcal{D} = \mathcal{D}_\zeta \cup \mathcal{D}_x$, with $\mathcal{D}_\zeta := \{\zeta_t | t \in \{1, \ldots, H\}\}$ and $\mathcal{D}_x := \{x_t | t \in \{1, \ldots, H\}\}$. It is assumed that the prior information is Gaussian $\theta \sim \mathcal{N}(\bar{\theta}_0, P_0)$. The MAP problem that is tackled is then equivalent to finding the state estimates and mode sequence that maximize the joint conditional probability of the parameter estimates and sequence $\hat{\sigma}$ given the data $\mathcal{D}$, i.e.

$$\left(\hat{\theta}, \hat{\sigma}_{0:H}\right) = \arg\max_{\theta, \sigma_{0:H}} \quad \Pr\left\{\theta, \sigma_{0:H} | \mathcal{D}\right\}. \tag{4.4}$$

where

$$\Pr\{\theta, \sigma_{0:H}|\mathcal{D}\} \propto \Pr\{\mathcal{D}_\zeta|\theta, \sigma_{0:H}, \mathcal{D}_x\} \Pr\{\sigma_{0:H}\} \Pr\{\theta\} \tag{4.5}$$

4.3 Joint MAP estimator

Using logarithms (4.4) is rewritten as

$$\begin{aligned}\left(\hat{\theta}, \hat{\sigma}_{0:H}\right) = \arg\min_{\theta, \sigma_{0:H}} & -\log(\Pr\{\sigma_{0:H}\}) - \log(\Pr\{\theta\}) \\ & -\log(\Pr\{\mathcal{D}_\zeta|\theta, \sigma_{0:H}, \mathcal{D}_x\}),\end{aligned} \tag{4.6}$$

where by taking into account the prior probabilities

$$-\log\left(\Pr\{\theta\}\right) = \frac{1}{2}\left(\left\|\theta - \bar{\theta}_0\right\|^2_{P_0^{-1}} + c_1\right), \tag{4.7}$$

with $c_1 = n_\theta \log(2\pi) + \log(|P_0|)$, and

$$-\log\left(\Pr\{\sigma_{0:H}\}\right) = \ell_{\sigma_0} + \sum_{t=1}^{H} \ell_{\sigma_t, \sigma_{t-1}},$$

with $\ell_{\sigma_t, \sigma_{t-1}} := -\log(p_{\sigma_t, \sigma_{t-1}})$ and $\ell_{\sigma_0} := -\log(p^0_{\sigma_0})$. Due to the independence of the noise random variables,

$$-\log(\Pr\{\mathcal{D}_\zeta|\theta, \sigma_{0:H}, \mathcal{D}_x\}) = -\sum_{t=1}^{H} \log(\Pr\{\zeta_t|\theta, \sigma_t, x_t\}) \tag{4.8}$$

and due to the Gaussian assumption

$$-\log(\Pr\{\zeta_t|\theta, \sigma_t, x_t\}) = \frac{1}{2}\left(\|\zeta_t - \Gamma_{\sigma_t, t}\theta\|^2_{R^{-1}_{\sigma_t}} + \beta_{\sigma_t} + c_2\right), \tag{4.9}$$

$\beta_i := \log(|R_i|)$ and $c_2 = n_\zeta \log(2\pi)$. It is assumed that $\beta_i \geq 0$, which can always be met by properly scaling the problem (see Chapter 3). By substituting these expressions into (4.6) and by defining the variables $\alpha_t = \sigma_{t-1}$, for $t \in \{1, 2, \ldots, H\}$, with $\alpha_{1:H} = (\alpha_1, \ldots, \alpha_H)$, the optimization problem becomes

$$J(\theta, \sigma_H) = \min_{\alpha_{1:H}} \quad \sum_{t=1}^{H} g(\theta, \sigma_t, \alpha_t, \zeta_t) + h(\theta, \sigma_0), \tag{4.10}$$

where $\alpha := (\alpha_1, \ldots, \alpha_H)$ and

$$g(\theta, \sigma_t, \alpha_t, \zeta_t) = \frac{1}{2}\left[\beta_{\sigma_t} + \|\zeta_t - \Gamma_{\sigma_t, t}\theta\|^2_{R^{-1}_{\sigma_t}} + 2\ell_{\sigma_t, \alpha_t}\right], \tag{4.11}$$

$$h(\theta, \sigma_0) = \ell_{\sigma_0} + \frac{1}{2} \left\| \theta - \bar{\theta}_0 \right\|^2_{P_0^{-1}}. \tag{4.12}$$

After the minimization one can find

$$(\hat{\theta}, \hat{\sigma}_H) = \underset{\theta, \sigma_H}{\arg\min}\, J(\theta, \sigma_H) \tag{4.13}$$

and obtain the desired estimate $\hat{\theta}$.

The optimal control problem in (4.10) can be solved using dynamic programming, which consists of the following steps:

1. Start with the *arrival cost* for $t = 0$, defined as

$$J_0(\theta, \sigma_0) = h(\theta, \sigma_0) \tag{4.14}$$

2. For $t \in \{1, \dots, H\}$, compute the arrival cost

$$J_t(\theta, \sigma_t) = \min_{\alpha_t} \{ g(\theta, \sigma_t, \alpha_t, \zeta_t) + J_{t-1}(\theta, \alpha_t) \}. \tag{4.15}$$

Then J in (4.10) equals the arrival cost J_H. An expression for the cost function J_t can be found using the following proposition. Let $\mathcal{E}_t$ denote the set of all possible mode histories up to time t, i.e.

$$\mathcal{E}_t := \{ \mathcal{T}_t^1, \mathcal{T}_t^2, \dots, \mathcal{T}_t^{n_{\mathcal{E}_t}} \}, \tag{4.16}$$

where $n_{\mathcal{E}_t} = m^t$ is the cardinality of the set $\mathcal{E}_t$, and

$$\mathcal{T}_t^i = \{ \sigma_0^i, \sigma_1^i, \dots, \sigma_{t-1}^i \}, \quad i \in \{1, \dots, m^t\}, \tag{4.17}$$

denotes the i^{th} possible mode history from time 0 to time $t-1$, where $\sigma_t^j \in S$ for every $j \in \{1, \dots, m^t\}$.

Proposition 4.1. *The arrival cost $J_t(\theta, \sigma_t)$ is given by*

$$J_t(\theta, \sigma_t) = \min_{\mathcal{T} \in \mathcal{E}_t} \frac{1}{2} \left(\theta - \hat{\theta}_t^{\mathcal{T}} \right)^\intercal \Pi_{\mathcal{T}} \left(\theta - \hat{\theta}_t^{\mathcal{T}} \right) + \gamma_{\mathcal{T}}, \tag{4.18}$$

where for each $\mathcal{T} \in \mathcal{E}_t$, $\Pi_{\mathcal{T}} = (P_t)^{-1}$, $\gamma_{\mathcal{T}} = c_t$, $\hat{\theta}_t^{\mathcal{T}} = \hat{\theta}_t$ and $\hat{\theta}_t$, c_t, P_t are obtained by iterating

$$\hat{\theta}_0 = \bar{\theta}_0, \quad c_0 = \ell_{\sigma_0}, \tag{4.19a}$$

and, for $j \in \{0, \dots, t-1\}$,

$$L_{j+1} = (R_{\sigma_{j+1}} + \Gamma_{\sigma_{j+1}, j+1} P_j \Gamma^\intercal_{\sigma_{j+1}, j+1})^{-1}, \tag{4.19b}$$

$$K_{j+1} = P_j \Gamma^\intercal_{\sigma_{j+1}, j+1} L_{j+1}, \tag{4.19c}$$

$$e_{j+1} = \zeta_{j+1} - \Gamma_{\sigma_{j+1}, j+1} \hat{\theta}_j \tag{4.19d}$$

$$\hat{\theta}_{j+1} = \hat{\theta}_j + K_{j+1} e_{j+1}, \tag{4.19e}$$

$$P_{j+1} = P_j - K_{j+1} \Gamma_{\sigma_{j+1}, j+1} P_j, \tag{4.19f}$$

$$c_{j+1} = \frac{1}{2} \left(\| e_{j+1} \|^2_{L_{j+1}} + \beta_{\sigma_{j+1}} \right) + \ell_{\sigma_{j+1}, \sigma_j} + c_j \tag{4.19g}$$

Note that the cardinality of the set $\mathcal{E}_t$ grows exponentially with increasing t, which makes computing the optimal solution intractable with increasing t.

4.4 Proposed recursive estimator using relaxed dynamic programming

In this section, relaxed dynamic programming [129] is used to approximate the solution to (4.18), described in Section 4.4.1. In Section 4.4.2 a variant that will lead to better results in the numerical example is also proposed, although it provides less guarantees.

4.4.1 Relaxed dynamic programming

Using relaxed dynamic programming [129] to approximate the solution to (4.18) amounts to finding approximating functions for J_t. To this effect, consider the following approximating functions $V_0(\theta, \sigma_0) = J_0(\theta_0, \sigma_0)$, and, for $t \in \{1, \dots, H\}$,

$$V_t(\theta, \sigma_t) = \min_{\mathcal{T} \in \mathcal{P}_t} \frac{1}{2} \left(\theta - \hat{\theta}_t^{\mathcal{T}}\right)^{\intercal} \Pi_{\mathcal{T}} \left(\theta - \hat{\theta}_t^{\mathcal{T}}\right) + \gamma_{\mathcal{T}}, \tag{4.20}$$

where $\mathcal{P}_t \subseteq \mathcal{E}_t$ can be seen as a pruned version of $\mathcal{E}_t$, where certain mode histories are discarded, thus reducing the complexity of the solution.

In order to achieve this, let

$$\mathcal{C}_t := \{(\mathcal{T}_{t-1}, \alpha_t) | \mathcal{T}_{t-1} \in \mathcal{P}_{t-1}, \alpha_t \in S\}, \tag{4.21}$$

denote the set containing all possible mode histories at time t considering the mode histories at $t-1$ that were not removed during pruning.

The set $\mathcal{P}_t$ is iteratively chosen in such a way that the approximated arrival cost function $V_t(\theta, \sigma_t)$ is always within a factor $\epsilon \geq 0$ of the optimal function $J_t(\theta, \sigma_t)$, for every θ and σ_t, i.e.,

$$J_t(\theta, \sigma_t) \leq V_t(\theta, \sigma_t) \leq (1+\epsilon) J_t(\theta, \sigma_t), \, \forall \theta, \sigma_t. \tag{4.22}$$

For (4.22) to make sense the cost J_t needs to be positive, which is the case due to the assumption that $\beta_i \geq 0$.

In order to choose the set $\mathcal{P}_t$ such that (4.22) holds, let

$$\begin{aligned} U_t(\theta, \sigma_t) &:= \min_{\alpha_t \in S} (1+\epsilon) g(\theta, \sigma_t, \alpha_t, \zeta_t) + V_{t-1}(\theta, \alpha_t), \\ &= \min_{\mathcal{T} \in \mathcal{C}_t} \frac{1}{2} (\theta - \bar{\theta}_t^{\mathcal{T}})^{\intercal} \bar{\Pi}_{\mathcal{T}} (\theta - \bar{\theta}_t^{\mathcal{T}}) + \bar{\gamma}_{\mathcal{T}}, \end{aligned}$$

where for each $\mathcal{T} = (\sigma_0, \ldots, \sigma_{t-2}, \alpha_t) \in \mathcal{C}_t$, $\bar{\Pi}_{\mathcal{T}} = \left(\bar{P}_t\right)^{-1}$, $\bar{\gamma}_{\mathcal{T}} = \bar{c}_t$, $\bar{\theta}_t^{\mathcal{T}} = \bar{\theta}_t$ and $\bar{\theta}_t$, $\bar{c}_t$, $\bar{P}_t$ are obtained by computing for $j = t-1$

$$\bar{L}_{j+1} = (\psi^{-1} R_{\sigma_{j+1}} + \Gamma_{\sigma_{j+1},j+1} P_j \Gamma_{\sigma_{j+1},j+1}^{\intercal})^{-1}, \tag{4.23a}$$

$$\bar{K}_{j+1} = P_j \Gamma_{\sigma_{j+1},j+1}^{\intercal} \bar{L}_{j+1}, \tag{4.23b}$$

$$\bar{e}_{j+1} = \zeta_{j+1} - \Gamma_{\sigma_{j+1},j+1} \hat{\theta}_j \tag{4.23c}$$

$$\bar{\theta}_{j+1} = \hat{\theta}_j + \bar{K}_{j+1} \bar{e}_{j+1}, \tag{4.23d}$$

$$\bar{P}_{j+1} = P_j - \bar{K}_{j+1} \Gamma_{\sigma_{j+1},j+1} P_j, \tag{4.23e}$$

$$\bar{c}_{j+1} = \frac{1}{2}\left(\|\bar{e}_{j+1}\|^2_{\bar{L}_{j+1}} + \psi \beta_{\sigma_{j+1}}\right) + \psi \ell_{\sigma_{j+1},\sigma_j} + c_j \tag{4.23f}$$

with $\psi = (1+\epsilon)$ and $\hat{\theta}_j$, P_j, c_j are obtained by iterating the recursion (4.19a)-(4.19g) for $j \in \{0, \ldots, t-2\}$. Note that U_t coincides with V_t when $\epsilon = 0$.

At each timestep t, the set $\mathcal{P}_t$ is a pruned version of the set $\mathcal{C}_t$ obtained as follows:

Relaxed Dynamic Programming procedure

1. Initialize $\mathcal{P}_t$ as empty.

2. Take the element (mode history) $\bar{\mathcal{T}}$ in $\mathcal{C}_t \setminus \mathcal{P}_t$ with the smallest $\gamma_{\bar{\mathcal{T}}}$ and check if it satisfies

$$\frac{1}{2}(\theta - \bar{\theta}_t^{\bar{\mathcal{T}}})^{\intercal} \bar{\Pi}_{\bar{\mathcal{T}}} (\theta - \bar{\theta}_t^{\bar{\mathcal{T}}}) + \bar{\gamma}_{\bar{\mathcal{T}}} \geq \min_{\mathcal{T} \in \mathcal{P}_t} \frac{1}{2}(\theta - \hat{\theta}_t^{\mathcal{T}})^{\intercal} \Pi_{\mathcal{T}} (\theta - \hat{\theta}_t^{\mathcal{T}}) + \gamma_{\mathcal{T}}, \, \forall \theta \in \mathbb{R}^{n_\theta}. \tag{4.24}$$

3. If (4.24) is not satisfied, then add the mode history $\bar{\mathcal{T}}$ of $\mathcal{C}_t$ to $\mathcal{P}_t$. If there are no more elements in $\mathcal{C}_t$, then stop, otherwise go to step 2.

The next result shows that this procedure guarantees that (4.22) is met. The proof is omitted since it follows similar steps as in the proof of a similar result in Chapter 3.

Theorem 4.2. *Let V_t be defined by* (4.20) *with the set $\mathcal{P}_t$ obtained from the procedure* (4.24). *Then* (4.22) *holds.*

□

This implies that at each time step the joint log-likelihood of the state and mode sequence estimate is within a factor ϵ of the optimal joint log-likelihood.

4.4.2 Variant of relaxed dynamic programming

As the first method is not always able to prune enough mode histories to keep the computation time reasonable (as is the case in the example), a second method is proposed that is able to prune more. In this method, consider

$$\begin{aligned} W_t(\theta, \sigma_t) &:= \min_{\alpha_t \in S} (1+\epsilon) \left[g(\theta, \sigma_t, \alpha_t, \zeta_t) + V_{t-1}(\theta, \alpha_t) \right], \\ &= \min_{\mathcal{T} \in \mathcal{C}_t} \frac{1}{2}(1+\epsilon)(\theta - \hat{\theta}_t^{\mathcal{T}})^\intercal \Pi_{\mathcal{T}} (\theta - \hat{\theta}_t^{\mathcal{T}}) + \gamma_{\mathcal{T}}, \end{aligned}$$

where for each $\mathcal{T} \in \mathcal{C}_t$, $\Pi_{\mathcal{T}} = (P_t)^{-1}$, $\gamma_{\mathcal{T}} = c_t$, $\hat{\theta}_t^{\mathcal{T}} = \hat{\theta}_t$ and $\hat{\theta}_t$, c_t, P_t are obtained by iterating (4.19a)-(4.19g).

The same procedure as in Section 4.4.1 is performed to find the pruned set, except for the check (4.24), which is replaced by

$$\begin{aligned} \frac{1}{2}(1+\epsilon) \left[(\theta - \hat{\theta}_t^{\bar{\mathcal{T}}})^\intercal \Pi_{\bar{\mathcal{T}}} (\theta - \hat{\theta}_t^{\bar{\mathcal{T}}}) + \gamma_{\bar{\mathcal{T}}} \right] \geq \\ \min_{\mathcal{T} \in \mathcal{P}_t} \frac{1}{2}(\theta - \hat{\theta}_t^{\mathcal{T}})^\intercal \Pi_{\mathcal{T}} (\theta - \hat{\theta}_t^{\mathcal{T}}) + \gamma_{\mathcal{T}}, \, \forall \theta \in \mathbb{R}^{n_\theta}. \end{aligned} \tag{4.25}$$

In the next theorem it will be shown that by performing this method for pruning it can be guaranteed that the following bound on the pruned cost

$$J_t(\theta, \sigma_t) \leq V_t(\theta, \sigma_t) \leq (1+\epsilon)^t J_t(\theta, \sigma_t), \, \forall \theta, \sigma_t, \tag{4.26}$$

is met.

Theorem 4.3. *Let V_t be defined by* (4.20) *with the set $\mathcal{P}_t$ obtained from the procedure* (4.25)*. Then* (4.26) *holds.*

□

Proof. The proof is provided by induction, by definition $V_0(\theta, \sigma_0) = J_0(\theta, \sigma_0)$ and assuming

$$V_{t-1}(\theta, \sigma_{t-1}) \leq (1+\epsilon)^{t-1} J_{t-1}(\theta, \sigma_{t-1}) \tag{4.27}$$

then it follows that

$$\begin{aligned} W_t(\theta, \sigma_t) &\leq \min_{\alpha_t \in S} (1+\epsilon) \left[g(\theta, \sigma_t, \alpha_t, \zeta_t) \right. \\ &\quad \left. + (1+\epsilon)^{t-1} J_{t-1}(\theta, \sigma_{t-1}) \right], \\ &\leq \min_{\alpha_t \in S} (1+\epsilon)^t \left[g(\theta, \sigma_t, \alpha_t, \zeta_t) + J_{t-1}(\theta, \sigma_{t-1}) \right], \\ &= (1+\epsilon)^t J_t(\theta, \sigma_t). \end{aligned}$$

Since $\epsilon \geq 0$ it follows from the definition of $W_t(\theta, \sigma_t)$ that

$$\min_{\mathcal{T} \in \mathcal{C}_t} \frac{1}{2}(\theta - \hat{\theta}_t^{\mathcal{T}})^\intercal \Pi_{\mathcal{T}} (\theta - \hat{\theta}_t^{\mathcal{T}}) + \gamma_{\mathcal{T}} \leq W_t(\theta, \sigma_t)$$

and by performing the pruning according to (4.25), where the inequality is checked for all $\theta \in \mathbb{R}^{n_\theta}$, the pruned set $\mathcal{P}_t$ can be substituted for $\mathcal{C}_t$ in the previous inequality. Then

$$V_t(\theta, \sigma_t) \leq W_t(\theta, \sigma_t) \leq (1+\epsilon)^t J_t(\theta, \sigma_t),$$

is obtained, achieving (4.26) and concluding the proof. □

4.4.3 Checking the pruning condition

Since checking if (4.24) (or similarly (4.25)) holds is in general hard, the following alternative test in terms of Linear Matrix Inequalities (LMIs) is proposed. First note that if the following condition holds then (4.24) holds

$$\begin{aligned}\exists \tau_j \geq 0, \text{ such that } & \sum_{j=1}^{n_{\mathcal{P}_t}} \tau_j = 1 \text{ and} \\ & \frac{1}{2}(\theta - \bar{\theta}_t^{\bar{\mathcal{T}}})^\intercal \bar{\Pi}_{\bar{\mathcal{T}}} (\theta - \bar{\theta}_t^{\bar{\mathcal{T}}}) + \bar{\gamma}_{\bar{\mathcal{T}}} \geq \\ & \sum_{j=1}^{n_{\mathcal{P}_t}} \tau_j \Big(\frac{1}{2}(\theta - \hat{\theta}_t^{\mathcal{T}_j})^\intercal \Pi_{\mathcal{T}_j} (\theta - \hat{\theta}_t^{\mathcal{T}_j}) + \gamma_{\mathcal{T}_j} \Big), \forall \theta \in \mathbb{R}^{n_\theta}. \quad (4.28)\end{aligned}$$

where $n_{\mathcal{P}_t}$ denotes the cardinality of the set $\mathcal{P}_t$ and $\mathcal{T}_j$ are the sequences in $\mathcal{P}_t$, i.e., $\mathcal{P}_t = \{\mathcal{T}_1, \ldots, \mathcal{T}_{n_{\mathcal{P}_t}}\}$. This follows from the fact that the right hand side of this latter equation is always larger or equal than the right hand side of (4.24). Letting $z = \begin{bmatrix}\theta^\intercal & 1\end{bmatrix}$, the inequality of this latter condition as can be written as

$$z^\intercal \bar{Z} z \geq z^\intercal (\sum_{j=1}^{n_{\mathcal{P}_t}} \tau_j Z_j) z, \forall \theta \in \mathbb{R}^{n_\theta},$$

where

$$\bar{Z} = \begin{bmatrix} \bar{\Pi}_{\bar{\mathcal{T}}} & -\bar{\Pi}_{\bar{\mathcal{T}}} \bar{\theta}_t^{\bar{\mathcal{T}}} \\ -(\bar{\theta}_t^{\bar{\mathcal{T}}})^\intercal \bar{\Pi}_{\bar{\mathcal{T}}} & (\bar{\theta}_t^{\bar{\mathcal{T}}})^\intercal \bar{\Pi}_{\bar{\mathcal{T}}} \bar{\theta}_t^{\bar{\mathcal{T}}} + 2\bar{\gamma}_{\bar{\mathcal{T}}} \end{bmatrix},$$

$$Z_j = \begin{bmatrix} \Pi_{\mathcal{T}_j} & -\Pi_{\mathcal{T}_j} \hat{\theta}_t^{\mathcal{T}_j} \\ -(\hat{\theta}_t^{\mathcal{T}_j})^\intercal \Pi_{\mathcal{T}_j} & (\hat{\theta}_t^{\mathcal{T}_j})^\intercal \Pi_{\mathcal{T}_j} \hat{\theta}_t^{\mathcal{T}_j} + 2\gamma_{\mathcal{T}_j} \end{bmatrix}.$$

Equivalently, one can consider this condition for every z instead of for $z = \begin{bmatrix}\theta^\intercal, & 1\end{bmatrix}$. In fact, letting

$$\begin{bmatrix} W_1 & w_2 \\ w_2^\intercal & w_3 \end{bmatrix} = \bar{Z} - (\sum_{j=1}^{n_{\mathcal{P}_t}} \tau_j Z_j)$$

this condition can be written as

$$\begin{bmatrix}\theta^\intercal & 1\end{bmatrix}\begin{bmatrix}W_1 & w_2\\ w_2^\intercal & w_3\end{bmatrix}\begin{bmatrix}\theta\\ 1\end{bmatrix}\geq 0,\quad \forall\theta\in\mathbb{R}^{n_\theta},$$

and by the Schur complement this condition is equivalent to $w_3\geq 0$ and $\theta^\intercal(W_1 - w_2(w_3^{-1})w_2^\intercal)\theta$, for every θ, so that the last entry of z plays no role. One can then check (4.28) (which implies (4.24)) with the following LMI condition: there exist $\tau_j\geq 0$ adding up to one $\sum_{j=1}^{n_{\mathcal{P}_t}}\tau_j = 1$ such that the following linear matrix inequalities (LMIs) hold

$$\exists\tau_j\geq 0,\text{ such that }\sum_{j=1}^{n_{\mathcal{P}_t}}\tau_j = 1\text{ and }\bar{Z}\geq\sum_{j=1}^{n_{\mathcal{P}_t}}\tau_j Z_j. \tag{4.29}$$

Similar LMIs can be obtained to check (4.25), but are omitted here for brevity.

4.5 Example

The effectiveness of the proposed recursive estimator is demonstrated on a second order, switched ARX model as introduced in Section 4.2.1:

$$y_t = \begin{cases}-0.2y_{t-1} + 0.24y_{t-2} + u_t + w_t & \text{for } \sigma_t = 1\\ -1.4y_{t-1} - 0.53y_{t-2} + u_t + w_t & \text{for } \sigma_t = 2\end{cases}$$

with $w_t\in\sim\mathcal{N}(0,\mathcal{R}_{\sigma_t})$ where $R_1 = R_2 = 1.0$. The goal is to estimate the parameter vector

$$\begin{aligned}\theta &= (a_1^1, a_2^1, b^1, a_1^2, a_2^2, b^2)\\ &= (-0.2, 0.24, 1, -1.4, -0.53, 1)\end{aligned}$$

as well as the mode sequence $\sigma_{0:H}$, where the length of the dataset is taken to be $H = 50$. As the input signal a sine wave with an amplitude of 10 and a frequency of 20Hz is chosen, i.e.

$$u(t) = 10\sin(2\pi\cdot 0.05t)$$

and the transition probabilities are given by

$$\begin{bmatrix}p_{11} & p_{12}\\ p_{21} & p_{22}\end{bmatrix} = \begin{bmatrix}0.7 & 0.3\\ 0.1 & 0.9\end{bmatrix}.$$

Figure 4.1 shows the input and output data in the top together with the mode sequence in the bottom.

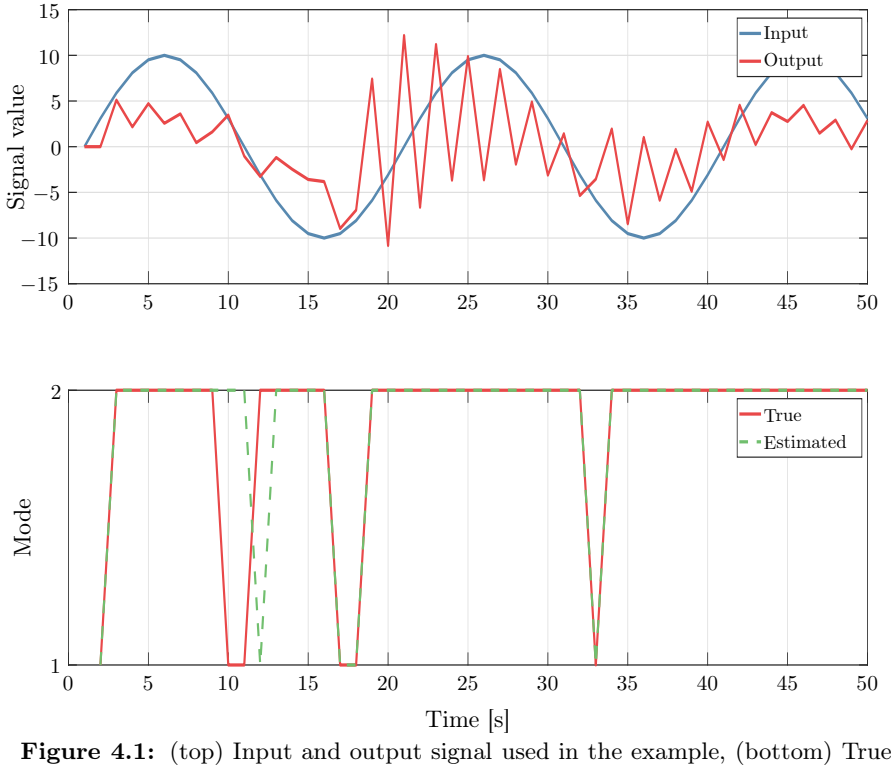


Figure 4.1: (top) Input and output signal used in the example, (bottom) True and estimated mode sequence

The method proposed in Section 4.4.2 is used, since the first method did not result in sufficient pruning of mode histories. The bound is set to $\epsilon = 1$ and the resulting estimates are compared with those obtained from running (4.19b)-(4.19g) using the actual modes (Known). The resulting parameter estimates (Estimated) together with the actual values (True) are shown in Figure 4.2, were it can be seen that the proposed method is able to estimate the parameters accurately. The same holds for the estimation of the mode sequence as shown in the bottom of Figure 4.1. From these results it is concluded that the bound given in (4.26) might be overly conservative.

In Figure 4.3 the number of hypotheses that the proposed method stores at each timestep for increasing values of ϵ are shown, showing that the proposed method greatly reduces the complexity of the original problem. Even for low values of ϵ the number of hypotheses converges to a constant level, making it feasible for online identification.

4.6 Conclusions

In this paper a method for the identification of Markov switched systems is provided that is recursive, has theoretical guarantees and is computationally feasible for online identification.

Suggestions for future research include the tightening of the guarantees and establishing a relation between the (maximum) number of hypotheses that need to be stored and a corresponding bound on the cost.

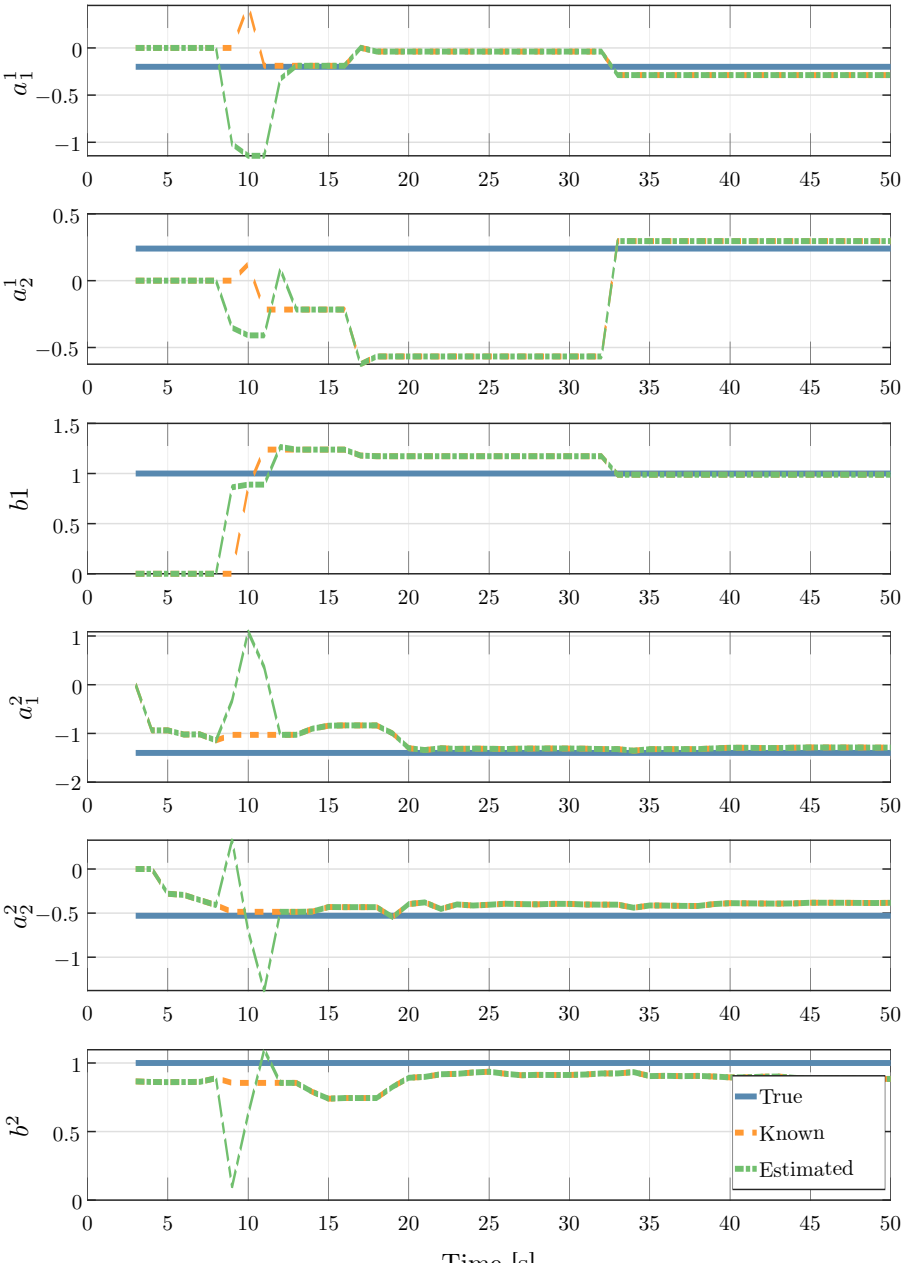


Figure 4.2: System parameters

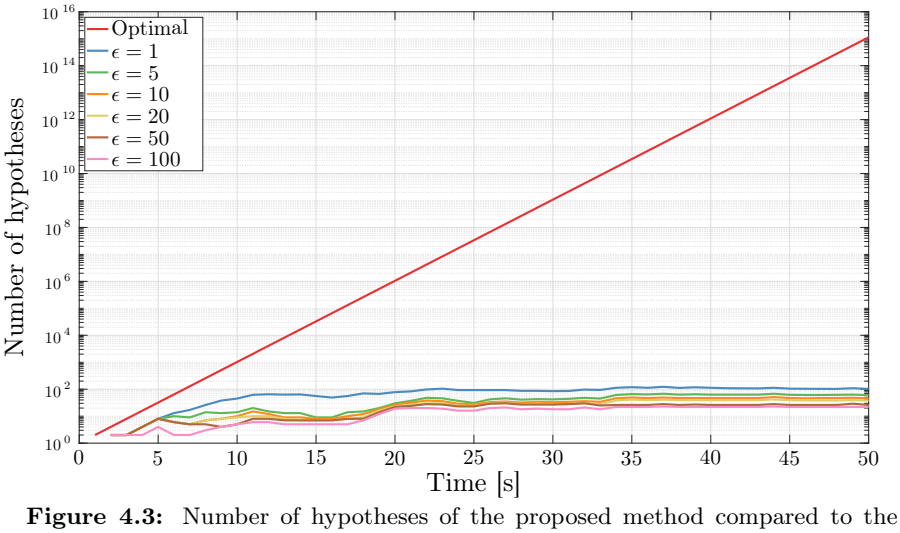


Figure 4.3: Number of hypotheses of the proposed method compared to the optimal solution

Part III

Control and Planning

Chapter 5

Model Predictive Controller for Quadcopters with Almost Global Trajectory Tracking Guarantees

This chapter provides a new method for trajectory tracking for quadcopters following a cascaded control approach. An outer-loop model predictive controller generates twice differentiable acceleration references, which provide attitude and angular velocity and acceleration references for a non-linear inner-loop controller. The model predictive controller allows for tracking of references while explicitly considering that the thrust of the quadcopter is upper and lower limited. It is proven that the overall strategy renders the trajectory tracking errors uniformly almost globally asymptotically stable. Via a numerical case study the advantages of the novel method are highlighted.

This chapter is based on [9, 5]

5.1 Introduction

Quadcopters are now widespread in the consumer market [54] and are used in many applications, such as agriculture [196], surveillance [81], wildlife monitoring [128], construction [61], (medicine) delivery [14, 217] and even extraterrestrial exploration [203, 85]. From a research point of view, quadcopters have received much attention as well, due to the challenging nonlinear dynamics needed to properly describe them [138], their under-actuated configuration [63] and high maneuverability [35]. In particular, for the control of quadcopters many avenues have been explored, such as sliding mode control [29], iterative learning control [88], nonlinear control [102, 123], reinforcement learning [97], to name a few. However, despite the broad range of approaches in quadcopter control, it is still hard to find in the literature a control approach with the following highly desired features:

1. Able to anticipate on future reference information;
2. Explicit handling of constraints on the states and inputs;
3. Implementable in real-time on embedded hardware;
4. Having stability or tracking error convergence guarantees.

Model Predictive Control (MPC) can potentially provide these features, due to its ability to anticipate using future reference information, handle constraints explicitly and the availability of well-established theoretical results that can be used to provide guarantees on closed-loop behavior [144]. Moreover, the typically high computational burden of (non-linear) MPC (3) can potentially be overcome by using the differential flatness property of quadcopters as in [79]. Yet, to the best of the authors' knowledge there are no MPC strategies currently available that provide all these favorable features. Indeed, MPC setups have been used for quadcopters in, for example, [152], exploiting differential flatness to achieve a convex optimization problem that is solved in real-time; however no stability guarantees are provided. In [22], a hierarchical MPC strategy was developed that uses linearization around the trajectory to make the problem computationally feasible, however tracking error convergence guarantees are not given.

A common element in many quadcopter control approaches is a cascaded control structure, where the control of the orientation of the quadcopter, known as the inner-loop, is separated from the position and velocity control, referred to as the outer-loop [108]. A strategy combining a thrust prioritizing inner-loop controller with an outer-loop controller that satisfies constraints on some of the states was presented in [114], again without stability and convergence guarantees. Control strategies following other approaches also do not provide all the desired features. For instance, the strategies provided in [67, 131] meet 3) and 4), however 1) and 2) are not satisfied.

Motivated by this gap in the literature, in this chapter a new MPC approach is proposed based on a general fourth-order quadcopter model as presented in [66]. A cascaded control design is employed, in which the control of the translational system, consisting of position and velocity kinematics and dynamics, is re-formulated as a linear problem by considering a virtual acceleration as input. This is referred to as the outer-loop tracking problem and an MPC strategy is designed that allows for meeting the desired features mentioned above. In order for the system to track the desired virtual acceleration generated by the MPC in the outer-loop, a desired thrust vector is generated, which is converted into a desired attitude that is tracked by using the attitude controller presented in [123]. The attitude tracking problem is referred to as the inner-loop, and the adopted controller requires that the desired virtual acceleration is twice differentiable, which is ensured by considering a linear fourth-order model for the outer-loop. It is shown how the constraints for the original nonlinear model can be translated into constraints for the linear fourth-order model, although this translation involves some degree of conservatism, as will be explained. Considering a given class of reference inputs, it is shown that the outer-loop MPC control strategy results in uniform global asymptotic stability (UGAS) for the tracking error. This convergence proof relies on new technical contributions that rely on state and input transformations and on recent results on globally stable MPC strategies for linear systems with input constraints. The advantages of the proposed cascaded control scheme are show in a numerical case study.

Compared to the preliminary results presented in [9], here: (i) a more complete, fourth-order model of the quadcopter as proposed in [66] is considered, (ii) trajectory tracking guarantees are provided for the full cascaded system rather than setpoint guarantees only for the outer-loop controller, and (iii) all the proofs and extensive explanations, not available in [9], are provided.

The remainder of this chapter is structured as follows. First, the dynamic model is discussed and the problem is defined in Section 5.3, after which the method is outlined in Section 5.4. The inner- and outer-loop controllers are presented in Sections 5.5 and 5.6, which are combined to provide the overall controller in Section 5.7, together with the proofs of the main results. Simulations results are provided in Section 5.8. Section 5.9 provides concluding remarks.

5.2 Preliminaries

In this section the notation used in this chapter is introduced. Let $e_i \in \mathbb{R}^3$ for $i \in \{1, 2, 3\}$ denote the standard unit vectors. The trace of a matrix A is denoted by $\mathrm{tr}\,(A)$ for a square matrix $A \in \mathbb{R}^{n\times n}$. The Euclidean, or two-norm, of a vector is denoted by $\|v\|$ for $v \in \mathbb{R}^n$, i.e., $\|v\| = \sqrt{v^\top v}$. The induced Euclidean matrix norm for $A \in \mathbb{R}^{n\times n}$ is denoted similarly by $\|A\|$, which is equal to $\sqrt{\lambda_{\max}(A^\top A)}$, where $\lambda_{\max}(A^\top A)$ denotes the largest eigenvalue of $A^\top A$. A positive definite and

semi-positive definite matrix A are denoted by $A \succ 0$ and $A \succeq 0$, respectively. A diagonal matrix with vectors of entries a, b on the diagonal is denoted as $\text{diag}(a, b)$.

Consider the non-autonomous system

$$\dot{x} = f(t, x), \tag{5.1}$$

with state x taking values in $\mathbb{R}^n$, time $t \in \mathbb{R}_{\geq 0}$ and $f : \mathbb{R}_{\geq 0} \times \mathbb{R}^n \to \mathbb{R}^n$ is piecewise continuous in t and locally Lipschitz in x. The origing $x = 0$ is an equilibrium point of (5.1) at $t = 0$ if

$$f(t, 0) = 0, \text{ for all } t \in \mathbb{R}_{\geq 0}. \tag{5.2}$$

In this chapter uniform global asymptotic stability (UGAS), uniform global exponential stability (UGES) and uniform local exponential stability (ULES) are considered, for which definitions are given in [107] and repeated here for convenience.

Definition 5.1 (cf. [107])**.** *The equilibrium point $x = 0$ of (5.1) is*

- *stable if, for each $\epsilon > 0$, there is $\delta = \delta(\epsilon, t_0) > 0$ such that*

$$\|x(t_0)\| < \delta \implies \|x(t\| < \epsilon, \text{ for all } t \geq t_0 \geq 0. \tag{5.3}$$

- *unstable if it is not stable.*
- *uniformly stable if, for each $\epsilon > 0$, there is $\delta = \delta(\epsilon) > 0$, independent of t_0, such that (5.3) is satisfied.*
- *asymptotically stable if it is stable and there is a positive constant $c = c(t_0)$ such that $x(t) \to 0$ as $t \to \infty$, for all $\|x(t_0)\| < c$.*
- *uniformly asymptotically stable if it is uniformly stable and there is a positive constant c, independent of t_0, such that for all $\|x(t_0)\| < c$, $x(t) \to 0$ as $t \to \infty$, uniformly in t_0; that is, for each $\eta > 0$, there is $T = T(\eta) > 0$ such that*

$$\|x(t)\| < \eta, \text{ for all } t \geq t_0 + T(\eta), \text{ for all } \|x(t_0)\| < c. \tag{5.4}$$

- *uniformly globally asymptotically stable (UGAS) if it is uniformly asymptotically stable, $\delta(\epsilon)$ can be chosen to satisfy $\lim_{\epsilon \to \infty} \delta(\epsilon) = \infty$, and, for each pair of positive numbers η and c, there is $T = T(\eta, c) > 0$ such that*

$$\|x(t)\| < \eta, \text{ for all } t \geq t_0 + T(\eta, c), \text{ for all } \|x(t_0)\| < c. \tag{5.5}$$

- *uniformly locally exponentially stable (ULES) if there exist positive constants c, k, and λ such that*

$$\|x(t)\| \leq k \|x(t_0)\| e^{-\lambda(t - t_0)}, \text{ for all } \|x(t_0)\| < c. \tag{5.6}$$

- *uniformly globally exponentially stable (UGES) if* (5.6) *is satisfied for any initial state* $x(t_0) \in \mathbb{R}^n$.

Global stabilization of a quadcopter involves global stabilization of its attitude on $SO(3)$ and global stabilization of its linear position. Since global stabilization on $SO(3)$ using a continuous control input can not be achieved, cf. [24], we need to relax such stability notion. In fact, uniform *almost* global asymptotic stability is the aim in this chapter, which is defined as follows:

Definition 5.2. *The equilibrium point* $x = 0$ *of* (5.1) *is uniformly almost globally asymptotically stable (UaGAS), if it is UGAS, except for initial conditions in a set of measure zero. That is,* (5.5) *holds for every* $x(t_0) \in \mathbb{R}^n \setminus \{M\}$, *where* M *is a set of measure zero.*

5.3 Quadcopter dynamics and problem formulation

In this section a model of the quadcopter dynamics is first introduced, followed by the introduction of feasible reference trajectories. Based on these two ingredients, a problem statement is provided.

5.3.1 Dynamic model of quadcopter

The model that is used here is based on [102]. However, in contrast to [102], stiff rotors and no external wind are considered here (as was also done in [66]). To present the resulting model, suitable coordinate frames are needed. To introduce them, let W denote a right-handed inertial (or world) frame according to the North-East-Down (NED) convention, with unit vectors along the axes denoted by $\{x_W, y_W, z_W\}$, forming an orthonormal basis. Let B denote a right-handed body-fixed frame with unit vectors $\{x_B, y_B, z_B\}$ forming an orthonormal basis, where these vectors are the axes of B with respect to W. The origin of the body-fixed frame coincides with the center of mass of the quadrotor, and z_B is aligned with z_W and the gravitational vector when the quadrotor is at hover, see Figure 5.1. The orientation of B with respect to W is represented by the rotation matrix $R = [x_B, y_B, z_B] \in SO(3)$. Let $\omega = [\omega_1, \omega_2, \omega_3]^\top$ denote the angular velocities of B relative to W, expressed in B. The position and linear velocity of the center of mass of the quadrotor with respect to W are denoted by $p = [p_x, p_y, p_z]^\top$ and $v = [v_x, v_y, v_z]^\top$, respectively.

Using the above variables, the model can now be described by the equations

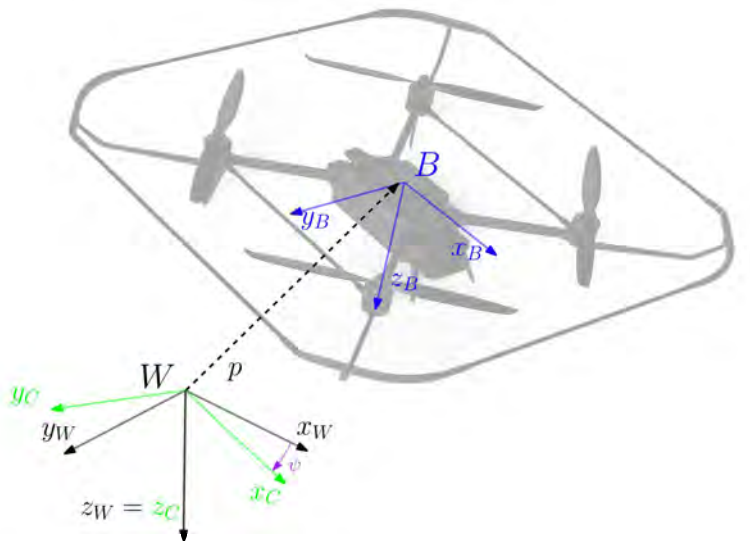


Figure 5.1: The inertial reference frame $\{W\}$ and body fixed frame $\{B\}$.

$$\dot{p} = v, \tag{5.7a}$$
$$\dot{v} = gz_W - Tz_B - RDR^\top v, \tag{5.7b}$$
$$\dot{R} = RS(\omega), \tag{5.7c}$$
$$J\dot{\omega} = S(J\omega)\omega - \tau_g - AR^\top v - C\omega + \tau. \tag{5.7d}$$

The forces acting on the translational dynamics ((5.7a)-(5.7b)) of the quadrotor consist of the gravity, given by gz_W, where g is the gravitational constant, the thrust force $-Tz_B$, where $T \geq 0$ denotes the magnitude of the combined thrust of the four propellers (mass-normalized), and a drag force as a result of rotor drag $-RDR^\top v = -d(R)v$, where $D = \text{diag}(d_x, d_y, d_z)$, $d_x, d_y, d_z > 0$ are the mass-normalized rotor drag coefficients and $d(R) = RDR^\top$.

The rotation of the quadcopter is characterized by the attitude kinematics given in (5.7c), where $S(a)$ represents a skew-symmetric matrix such that $S(a)b = a \times b$ for any vectors $a, b \in \mathbb{R}^3$ and the dynamics given in (5.7d), where $J \in \mathbb{R}^{3\times3}$ is the inertia matrix, $\tau_g \in \mathbb{R}^3$ are torques resulting from gyroscopic effects, A and C are constant matrices and $\tau = [\tau_1, \tau_2, \tau_3]^\top \in \mathbb{R}^3$ is the torque input.

The thrust is considered to be non-negative and limited according to

$$0 \leq T(t) \leq T_{\max}, \text{ for all } t \in \mathbb{R}_{\geq 0}, \tag{5.8}$$

where $T_{\max} > g$ is the maximal thrust. This is a physical restriction dictated by the fact that the propellers can only generate limited thrust upwards and must be capable of counteracting the gravitational force. Note that the thrust T and the torque τ are considered as the control inputs of the quadcopter. Effects such as motor dynamics and propeller aerodynamics are omitted.

5.3.2 Reference trajectory

In this chapter the focus is on a certain class of feasible reference trajectories. In fact, a reference trajectory $(\bar{p}, \bar{v}, \bar{R}, \bar{\omega}, \bar{T}, \bar{\tau}) : \mathbb{R}_{\geq 0} \to \mathbb{R}^3 \times \mathbb{R}^3 \times SO(3) \times \mathbb{R}^3 \times \mathbb{R} \times \mathbb{R}^3$ is called *feasible*, if it satisfies the dynamics (5.7) in the sense that for all $t \in \mathbb{R}_{\geq 0}$:

$$\dot{\bar{p}} = \bar{v}, \tag{5.9a}$$
$$\dot{\bar{v}} = g z_W - \bar{T} \bar{z}_B - \bar{R} D \bar{R}^\top \bar{v}, \tag{5.9b}$$
$$\dot{\bar{R}} = \bar{R} S(\bar{\omega}), \tag{5.9c}$$
$$J \dot{\bar{\omega}} = S(J\bar{\omega})\bar{\omega} - \tau_g - A \bar{R}^\top \bar{v} - C\bar{\omega} + \bar{\tau}, \tag{5.9d}$$

and

$$0 < \underline{\epsilon} \leq \bar{T}(t) \leq T_{\max} - \bar{\epsilon}, \tag{5.10}$$

for fixed $\bar{\epsilon}, \underline{\epsilon} > 0$. Note that the thrust of a feasible reference must be strictly greater than zero and have a maximum that is strictly smaller than the maximal thrust $T_{\max}$ of the actual quadcopter.

5.3.3 Problem statement

Given a feasible reference trajectory, the error coordinates can be defined as

$$\tilde{p} = \bar{p} - p, \tag{5.11a}$$
$$\tilde{v} = \bar{v} - v, \tag{5.11b}$$
$$\tilde{R} = \bar{R}^\top R, \tag{5.11c}$$
$$\tilde{\omega} = \omega - \tilde{R}^\top \bar{\omega}, \tag{5.11d}$$

which can be used to formulate the main problem of this chapter as follows.

Problem 1. *Given a feasible reference trajectory* $(\bar{p}, \bar{v}, \bar{R}, \bar{\omega}, \bar{T}, \bar{\tau})$*, find control laws*

$$\begin{aligned} T &= T(p, v, R, \omega, \bar{p}, \bar{v}, \bar{R}, \bar{\omega}, \bar{T}, \bar{\tau}), \\ \tau &= \tau(p, v, R, \omega, \bar{p}, \bar{v}, \bar{R}, \bar{\omega}, \bar{T}, \bar{\tau}), \end{aligned} \tag{5.12}$$

such that (5.8) *holds and such that for the closed-loop system* (5.7), (5.9), (5.12)

$$\lim_{t\to\infty}(\tilde{p}(t),\tilde{v}(t),\tilde{R}(t),\tilde{\omega}(t)) = (0,0,I,0), \tag{5.13}$$

for almost all initial conditions $(p(0),v(0),R(0),\omega(0)) \in \mathbb{R}^3\times\mathbb{R}^3\times SO(3)\setminus\{M\}\times\mathbb{R}^3$*, where* M *is a set of measure zero.*

Assumption 5.3. *It is assumed that the effects of rotation in the drag force are negligible, i.e., that* $d(R) = RDR^\top \approx D$*, for all trajectories considered.*

Remark 5.4. *While Problem 1 assumes, for simplicity, a continuous-time controller implementation by making* T *and* τ *at time* t *a function of the state and reference at time* t*, the actual proposed controller will take a sample and hold form. This is required due to the MPC approach. More formally, we can state that* T,τ *are functions of the state and reference from time* 0 *to time* t*.*

5.4 Methodology

In order to solve the nonlinear tracking problem defined in Problem 1, a cascaded controller design is employed. The cascade consists of an outer loop and an inner loop, that contain the translational ((5.7a)-(5.7b)) and rotational ((5.7c)-(5.7d)) subsystems, respectively, as depicted in Figure 5.2. First the setup of the cascade is discussed, followed by the resulting constraints on the subsystems that follow from the setup and the thrust constraint (5.8). These constraints also ensure that the variables used in this section are well-defined. The section is concluded with the problem definitions related to the inner loop and outer loop, that are solved in the subsequent sections.

5.4.1 Cascaded trajectory tracking setup

Considering the definition of the position and the velocity errors in (5.11a)-(5.11b), their dynamics can be found by subtracting (5.9a)-(5.9b) from (5.7a)-(5.7b), giving

$$\begin{aligned}\dot{\tilde{p}} &= \tilde{v},\\ \dot{\tilde{v}} &= -D\tilde{v} + Tz_B - \bar{T}\bar{z}_B.\end{aligned}$$

Introducing a new virtual input $a_d \in \mathbb{R}^3$, referred to as the desired acceleration (error), the actual acceleration error is replaced by a_d, i.e.,

$$a_d = Tz_B - \bar{T}\bar{z}_B, \tag{5.14}$$

which leads to

$$\dot{\tilde{p}} = \tilde{v}, \tag{5.15a}$$

$$\dot{\tilde{v}} = -D\tilde{v} + a_d. \tag{5.15b}$$

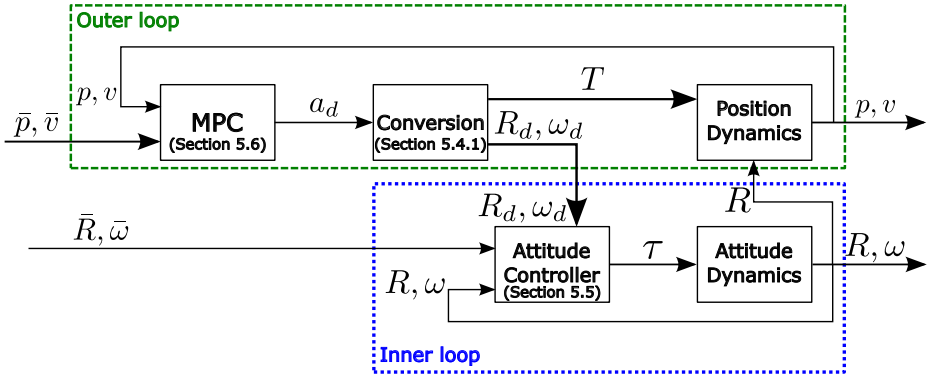


Figure 5.2: Overview of the proposed control strategy. In the outer loop, MPC is used to generate desired accelerations a_d, as discussed in Section 5.6. As discussed in Section 5.4.1, these accelerations are subsequently converted to thrust inputs T that are applied to the quadcopter as well as desired attitudes R_d and angular velocities ω_d. These are then combined with the reference and measured attitudes and angular velocities, $\bar{R}, \bar{\omega}$ and R, ω, respectively, in the attitude controller in the inner loop, resulting in torque inputs τ that are applied to the quadcopter, see Section 5.5. The outer-loop and inner-loop strategies are combined in Section 5.7, where the stability of the overall system is discussed as well.

In Section 5.6 an MPC strategy is used to find a desired acceleration a_d such that the dynamics in (5.15) are stabilized and the constraint in (5.8) is satisfied. Based on this desired acceleration, the inputs τ and T are then used to have the actual acceleration converge to the desired acceleration, i.e., to have the error

$$\tilde{a} = a_d - T z_B + \bar{T} \bar{z}_B. \tag{5.16}$$

converge to zero. By setting (5.16) to zero, it follows that

$$T z_B = a_d + \bar{T} \bar{z}_B, \tag{5.17}$$

and by setting the thrust as the magnitude of the vector on the right-hand side, i.e.,

$$T = \left\| a_d + \bar{T} \bar{z}_B \right\|, \tag{5.18}$$

the first input is determined. Note that in order to ensure the correct direction of the vector on the right-hand side in (5.17), the vector z_B can be used. However, this is not a direct control input of the quadcopter. Therefore, instead, the desired rotation of the quadcopter, denoted by R_d, is determined next. First, note that pre-multiplication of (5.17) with $\bar{R}^\top$ results in

$$T \bar{R}^\top R e_3 = \bar{R}^\top a_d + \bar{T} \bar{R}^\top \bar{R} e_3,$$

or, equivalently,

$$T\tilde{R}e_3 = \bar{R}^\top a_d + \bar{T}e_3. \tag{5.19}$$

The desired thrust direction is then set to

$$z_{B,d} = \begin{bmatrix} z_{B,d_1} \\ z_{B,d_2} \\ z_{B,d_3} \end{bmatrix} = \frac{\bar{R}^\top a_d + \bar{T}e_3}{\left\| \bar{R}^\top a_d + \bar{T}e_3 \right\|}, \tag{5.20}$$

and the remaining columns of the desired quadcopter orientation are set to

$$y_{B,d} = \frac{z_{B,d} \times e_1}{\|z_{B,d} \times e_1\|} = \begin{bmatrix} 0 & \frac{z_{B,d_3}}{\alpha} & \frac{-z_{B,d_2}}{\alpha} \end{bmatrix}^\top,$$
$$x_{B,d} = y_{B,d} \times z_{B,d} = \begin{bmatrix} \alpha & -\frac{z_{B,d_1} z_{B,d_2}}{\alpha} & -\frac{z_{B,d_1} z_{B,d_3}}{\alpha} \end{bmatrix}^\top,$$

for $\alpha = \sqrt{z_{B,d_2}^2 + z_{B,d_3}^2}$. The desired attitude of the quadcopter is then given by

$$R_d = \begin{bmatrix} x_{B,d}, & y_{B,d}, & z_{B,d} \end{bmatrix}. \tag{5.21}$$

Roughly speaking, this ensures that when the errors in (5.15) converge to zero, a_d converges to zero and $z_{B,d} \to e_3$, making $R_d \to I$.

Note that the Euclidean norm is invariant under rotation, so that the thrust defined in (5.18) can be written as

$$T = \left\| a_d + \bar{T}\bar{z}_B \right\| = \left\| \bar{R}^\top a_d + \bar{T}e_3 \right\|.$$

Combined with (5.20) and (5.21), this shows that (5.19) can be written as

$$T\tilde{R}e_3 = \bar{R}^\top a_d + \bar{T}e_3 = TR_d e_3. \tag{5.22}$$

From this it becomes clear that by making $\tilde{R}$ converge to R_d, the desired acceleration is achieved. This will be achieved using the torque τ, which is generated by the attitude tracking controller presented in Section 5.5. This controller will require differentiable setpoints for the desired angular velocity, which are found by first noting that from $\dot{R}_d = R_d S(\omega_d)$ it follows that

$$\dot{x}_{B,d} = \omega_{d_3} y_{B,d} - \omega_{d_2} z_{B,d}, \tag{5.23a}$$
$$\dot{y}_{B,d} = -\omega_{d_3} x_{B,d} + \omega_{d_1} z_{B,d}, \tag{5.23b}$$
$$\dot{z}_{B,d} = \omega_{d_2} x_{B,d} - \omega_{d_1} y_{B,d}. \tag{5.23c}$$

Pre-multiplying (5.23b) with $x_{B,d}^\top$ and (5.23c) with $x_{B,d}^\top$ and $y_{B,d}^\top$ results in an expression for the desired angular velocity as

$$\omega_d = \begin{bmatrix} -y_{B,d}^\top \dot{z}_{B,d} \\ x_{B,d}^\top \dot{z}_{B,d} \\ -x_{B,d}^\top \dot{y}_{B,d} \end{bmatrix}. \tag{5.24}$$

The above derivations show that the desired acceleration a_d can be achieved by generating the desired attitude in (5.21) and using the thrust as in (5.18). Next, it is discussed how the constraint on the thrust in (5.8) can be converted to constraints on the desired acceleration a_d, which is to be generated by the MPC. Moreover, the constraints on the desired acceleration will also ensure that the variables in this section are well-defined.

5.4.2 Constraints

The magnitude of the thrust vector is constrained according to (5.8) and for (5.20) to be defined it is required that $T \neq 0$. Furthermore, in order for $y_{B,d}$ to be well-defined it is required that $z_{B,d}$ and e_1 are never parallel, which is achieved by requiring $z_{B,d_3} > 0$, which also ensures $T \neq 0$. Note that from (5.20) it follows that ensuring $z_{B,d_3} > 0$ is equivalent to ensuring $\bar{z}_B^\top a_d + \bar{T} > 0$. To this effect, $a_d \in \mathcal{A}$ is constrained and the set of admissible values that the desired acceleration can take is defined as

$$\mathcal{A}(\bar{R}, \bar{T}) := \{a_d \in \mathbb{R}^3 | 0 < \left\| a_d + \bar{T}\bar{z}_B \right\| \leq T_{\max}, \bar{z}_B^\top a_d + \bar{T} > 0\}. \tag{5.25}$$

The inner-loop controller requires setpoints for the desired angular velocity and its derivative, which means that the desired acceleration needs to be twice differentiable, see (5.24).

5.4.3 Cascaded problem definition

The original tracking control problem as defined in Problem 1 is now split into two subproblems, namely an outer-loop and an inner-loop problem. The outer-loop problem is formulated as

Problem 2 (Outer-loop problem)**.** *Find a twice differentiable virtual acceleration control law $a_d = a_d(p, v, \bar{p}, \bar{v}, \bar{T})$, such that the origin $(\tilde{p}(t), \tilde{v}(t)) = (0, 0)$ of the system (*(5.15a)-(5.15b)*) is UGAS and such that $a_d \in \mathcal{A}(\bar{R}, \bar{T})$, for all $t \in \mathbb{R}_{\geq 0}$, where $\mathcal{A}(\bar{R}, \bar{T})$ is defined in* (5.25)*.*

Since it is desired to steer $\tilde{R}$ to R_d, the attitude error and angular velocity error considered in the inner-loop problem are defined as

$$R_e = R_d^\top \tilde{R} \tag{5.26}$$

and

$$\omega_e = \omega - \tilde{R}^\top \bar{\omega} - R_e^\top \omega_d, \tag{5.27}$$

respectively. The inner-loop problem is now formulated as

Problem 3 (Inner-loop problem)**.** *Find a control law $\tau = \tau(R, \omega, \bar{R}, \bar{\omega}, \bar{T}, a_d$, such that the origin $(R_e(t), \omega_e(t)) = (I, 0)$ of the system (*(5.26)-(5.27)*) is UaGAS.*

In Section 5.7 it will be shown that by solving Problems 2 and 3 a solution to Problem 1 can be obtained.

Remark 5.5. *Similarly to Remark 5.4, we can state that the control laws in Problem 2 and 3 are rather a function of state and reference variables from time* 0 *to time* t*, to accent the sampled-data approach.*

5.5 Inner-loop tracking

As mentioned in the introduction, many controllers for stabilizing the attitude dynamics of quadcopters have been proposed over the years. Here, a controller similar to the one proposed in [123] is employed, because it provides ULES and UaGAS for the attitude dynamics. The dynamics of the error variables in ((5.26),(5.27)) are given by

$$\begin{aligned}
\dot{R}_e &= R_e S(\omega_e), \\
J\dot{\omega}_e &= S(J\omega)\omega - \tau_g - AR^\top v - C\omega + \tau \\
&\quad - J\tilde{R}^\top J^{-1}\left(S(J\bar{\omega})\bar{\omega} - \tau_g - A\bar{R}^\top \bar{v} - C\bar{\omega} + \bar{\tau}\right) \\
&\quad + J\left[\left(S(\omega)\tilde{R}^\top - \tilde{R}^\top S(\bar{\omega})\right)\bar{\omega} + S(\omega_e)R_e^\top \omega_d - R_e^\top \dot{\omega}_d\right],
\end{aligned}$$

which, combined with the input

$$\begin{aligned}
\tau &= -K_\omega \omega_e + K_R \sum_{i=1}^{3} k_i (e_i \times R_e^\top e_i) \\
&\quad - S(J\omega)\omega + \tau_g + AR^\top v + C\omega \\
&\quad + J\tilde{R}^\top J^{-1}\left(S(J\bar{\omega})\bar{\omega} - \tau_g - A\bar{R}^\top \bar{v} - C\bar{\omega} + \bar{\tau}\right) \\
&\quad - J\left[\left(S(\omega)\tilde{R}^\top - \tilde{R}^\top S(\bar{\omega})\right)\bar{\omega} + S(\omega_e)R_e^\top \omega_d - R_e^\top \dot{\omega}_d\right],
\end{aligned} \tag{5.28}$$

result in the closed-loop system

$$\dot{R}_e = R_e S(\omega_e), \tag{5.29a}$$

$$J\dot{\omega}_e = -K_\omega \omega_e + K_R \sum_{i=1}^{3} k_i (e_i \times R_e^\top e_i), \tag{5.29b}$$

with distinct $k_i > 0$, $K_\omega \succ 0$ and $K_R \succ 0$. The following theorem asserts stability for this closed-loop system:

Theorem 5.6 (cf. [123], Theorem 4)**.** *The system in* (5.29) *has* $(I, 0)$ *as a ULES and UaGAS equilibrium. That is, let* $E_c = \{I, diag(1, -1, -1), diag(-1, 1, -1),$

diag$(-1,-1,1)\}$*, then* R_e *converges to* E_c *and* ω_e *converges to zero. The equilibria* $(R_e, 0)$ *of* (5.29)*, where* $R_e \in E_c \setminus \{I\}$*, are unstable and the set of all initial conditions converging to the equilibria* $(R_e, 0)$*, where* $R_e \in E_c \setminus \{I\}$ *form a lower dimensional manifold.*

Since R_e converges to I for almost all initial conditions it follows from (5.26) that $\tilde{R}$ converges to R_d for almost all initial conditions. Moreover, as $\omega_e \to 0$ it follows from (5.27) that $\tilde{\omega} \to R_e^\top \omega_d$, which combined with $R_e \to I$ results in $\tilde{\omega} \to \omega_d$. This solves Problem 3.

5.6 Outer-loop tracking

For the outer-loop control problem defined in Problem 2 a model predictive control (MPC) strategy is used, that allows for the desired acceleration to be constrained to the set defined in (5.25), while still providing appropriate stability guarantees. As is common for many MPC strategies, the MPC law used is formulated in discrete time. An overview of the MPC strategy is provided in Figure 5.3. In this section, the actual optimal control problem (OCP) that is solved is presented in a stepwise manner, starting with the discretization of the dynamics ((5.15a)-(5.15b)) in Section 5.6.1. In this same section, the constraints on the continuous states are replaced by constraints on the discrete input, which still ensure the continuous-time satisfaction of the constraint on the desired acceleration a_d in (5.25). An MPC law is formulated based on the resulting discretized system in Section 5.6.4. This MPC law uses the fact that a globally stabilizing control law is known, which is introduced just before, in Section 5.6.3. In Section 5.6.5 it will be shown that stability and constraint satisfaction is achieved for the continuous time system as well.

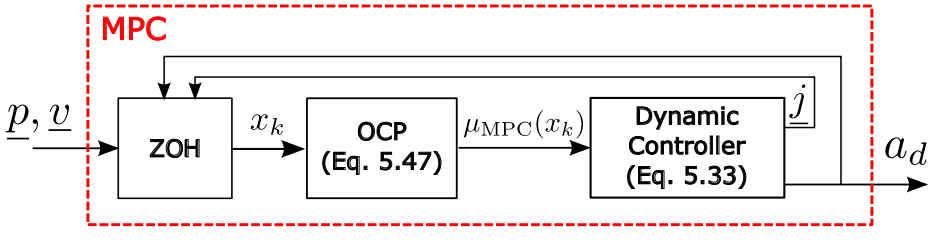


Figure 5.3: Overview of the proposed MPC strategy, for one of the three axes. The position and velocity error dynamics are extended into a dynamic controller structure in (5.33), which is subsequently discretized using exact discretization with a ZOH input. This discretized system forms the basis for the optimal control problem (OCP) that is solved for the MPC strategy in (5.47).

5.6.1 Discretization and input transformation

As mentioned, the MPC law used is formulated in discrete-time, and in order to be able to provide twice differentiable desired accelerations for the inner-loop the following extended version of ((5.15a)-(5.15b)) is considered:

$$\dot{\tilde{p}} = \tilde{v}, \tag{5.30a}$$

$$\dot{\tilde{v}} = -D\tilde{v} + a_d, \tag{5.30b}$$

$$\dot{a}_d = -\frac{1}{\gamma}(a_d + j), \tag{5.30c}$$

$$\dot{j} = -\frac{1}{\gamma}(j + s), \tag{5.30d}$$

where $\gamma > 0$, $\tilde{p}, \tilde{v}, a_d, j \in \mathbb{R}^3$ and the snap $s \in \mathbb{R}^3$ is considered as the input. By designing a piecewise constant (zero-order-hold (ZOH)) control law for s in this model, instead of for a_d directly, it is ensured that a_d is twice differentiable for the original model (5.15). Recall that twice differentiability of the desired acceleration a_d is required by the inner-loop controller. Note that for the purpose of ensuring differentiability, integrator dynamics would have sufficed, however the first-order filter dynamics will be used at the end of this subsection to provide constraint satisfaction in between sample times.

Next, a_d is constrained to lie in a more conservative set than strictly necessary, i.e., contained in the set defined in (5.25). However, with the benefit that this idea will result in an easier control design to generate a_d by decoupling the constraint in (5.25). First note that

$$\left\| a_d + \bar{T}\bar{z}_B \right\| \leq \|a_d\| + \bar{T}.$$

Hence, if

$$\|a_d\| + \bar{T} \leq T_{\max},$$

then the upper-bound in (5.25), i.e., $\left\| a_d + \bar{T}\bar{z}_B \right\| \leq T_{\max}$, is met. Furthermore, it also required that $\bar{z}_B^\top a_d + \bar{T} > 0$, which can be ensured by requiring

$$\|a_d\| \leq \bar{T} - \delta,$$

for some small $0 < \delta < \underline{\epsilon}$. These last two conditions describe a sphere for the desired acceleration a_d to be in of radius $T_{\max} - \bar{T}$ and $\bar{T} - \delta$, respectively. However, similarly to [152], a more conservative, box approximation $\mathcal{A}_L(\bar{T}) \subset \mathcal{A}(\bar{R}, \bar{T})$ is considered, where $\mathcal{A}(\bar{R}, \bar{T})$ as defined in (5.25) and

$$\mathcal{A}_L(\bar{T}(t)) := \{a_d \in \mathbb{R}^3 | -L \leq a_{d,i} \leq L, i \in \{1,2,3\}\}, \tag{5.31}$$

with

$$L(t) = \frac{1}{\sqrt{3}} \min\left(\bar{T}(t) - \delta, T_{\max} - \bar{T}(t)\right), \tag{5.32}$$

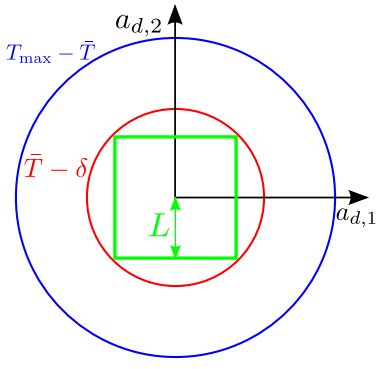


Figure 5.4: Schematic depiction of the constraints on the desired acceleration a_d in 2D (so that circles and squares are considered, instead of spheres and cubes, respectively). The blue and red circles have radii of $T_{\max} - \bar{T}$ and $\bar{T} - \delta$, respectively. Note that the radius of each circle changes as the reference thrust $\bar{T}$ changes, so that it is not always the case that the red circle is smaller than the blue circle. In this case however, the red circle is the smallest, and thus the most restricting for the desired acceleration a_d to have to lie in. In this circle the largest square that fits is depicted in green, and the constraint on the desired acceleration can now be decoupled by limiting each of the components to smaller than $\pm L$.

which can be viewed as the largest cube that fits in the sphere that is most restricting, see Figure 5.4. Note that $a_d \in \mathcal{A}_L(\bar{T})$ implies $a_d \in \mathcal{A}(\bar{R}, \bar{T})$. This allows for the consideration of three separate, constrained, scalar systems given by

$$\dot{\underline{p}}^i = \underline{v}^i, \tag{5.33a}$$
$$\dot{\underline{v}}^i = -d^i \underline{v}^i + \underline{a}^i, \tag{5.33b}$$
$$\dot{\underline{a}}^i = -\frac{1}{\gamma}(\underline{a}^i + \underline{j}^i), \tag{5.33c}$$
$$\dot{\underline{j}}^i = -\frac{1}{\gamma}(\underline{j}^i + \underline{s}^i), \tag{5.33d}$$

with $i \in \{1, 2, 3\}$, $\underline{p}^i, \underline{v}^i, \underline{a}^i, \underline{j}^i, \underline{s}^i \in \mathbb{R}$, d^i the corresponding component of D and $\underline{a}^i \in \underline{\mathcal{A}}_L(\bar{T}(t)) := \{a \in \mathbb{R} | -L(t) \leq a \leq L(t)\}$. The index i will be omitted from here on whenever possible to improve readability, since the three scalar systems are of the same form.

Using exact discretization with a ZOH input, i.e.,

$$\underline{s}(t) = \underline{s}(t_k), \quad t \in [t_k, t_{k+1}), \tag{5.34}$$

with $t_k = kh$, $k \in \mathbb{N}_{\geq 0}$, and $h > 0$ the sample time, results in the discrete-time system

$$x_{k+1} = Ax_k + Bu_k \tag{5.35}$$

where $x_k = x(t_k)$, $x = \begin{bmatrix} \underline{p} & \underline{v} & \underline{a} & \underline{j} \end{bmatrix}^\top$ taking values in $\mathbb{R}^4$, $u_k = \underline{s}(t_k)$ taking values in $\mathbb{R}$ and $k \in \mathbb{N}_{\geq 0}$. The system matrices are of the form

$$A = \begin{bmatrix} 1 & c_1 & c_2 & c_3 \\ 0 & c_4 & c_5 & c_6 \\ 0 & 0 & c_7 & c_8 \\ 0 & 0 & 0 & c_7 \end{bmatrix}, \quad B = \begin{bmatrix} c_9 \\ c_{10} \\ c_{11} \\ c_{12} \end{bmatrix}, \tag{5.36}$$

where $c_i > 0$, for $i = 1, 2, \ldots, 12$. The time-varying constraint in (5.32) can be lower bounded by a positive constant as

$$0 < \Delta = \inf_{t \in \mathbb{R}_{\geq 0}} L(t), \tag{5.37}$$

since $\bar{T}(t)$ is limited as in (5.8) and $\delta < \underline{\epsilon}$. The constraint $\underline{a} \in \underline{\mathcal{A}}_L(\bar{T}(t))$ is then ensured by initializing the controller such that $|\underline{a}(0)| \leq \Delta$, $|\underline{j}(0)| \leq \Delta$ and restricting the input as

$$-\Delta \leq u_k \leq \Delta, \text{ for all } k \in \mathbb{N}_{\geq 0}. \tag{5.38}$$

This choice of controller saturation results in the satisfaction of the constraint (5.32) as can be seen by considering the evolution of the acceleration and the jerk of the system (5.33) in between sample times, which is given by

$$\begin{aligned} \underline{a}(t) &= \alpha(t)\underline{a}_k + \beta(t)\underline{j}_k + [1 - \alpha(t) - \beta(t)]u_k, \\ \underline{j}(t) &= \alpha(t)\underline{j}_k + [1 - \alpha(t)]u_k, \end{aligned}$$

for $t \in [t_k, t_{k+1})$, where $\alpha(t) = e^{-t/\gamma}$ and $\beta(t) = \frac{t}{\gamma}e^{-t/\gamma}$. From these equations it follows that during the first sample period the jerk is limited by

$$|\underline{j}(t)| \leq \alpha(t)\Delta + [1 - \alpha(t)]\Delta = \Delta, \tag{5.39}$$

for $t \in [t_k, t_{k+1})$, so that the jerk is limited at the sample point, as well as between sample points. Similarly, the acceleration is bounded as

$$|\underline{a}(t)| \leq \alpha(t)\Delta + \beta(t)\Delta + [1 - \alpha(t) - \beta(t)]\Delta = \Delta, \tag{5.40}$$

for $t \in [t_k, t_{k+1})$. This also holds for all subsequent sample periods, so that it can be concluded that the constraint in (5.31), and thus the constraint in (5.25), is satisfied for all $t \in \mathbb{R}_{\geq 0}$.

5.6.2 MPC formulation

The MPC law for the scalar system (5.35) consists of solving the following optimal control problem (OCP) at each time step:

$$\begin{aligned} \min_{U_k} \;& J(x_k, U_k) = l_T(x_{N|k}) + \sum_{\ell=0}^{N-1} l(x_{\ell|k}, \hat{u}_{\ell|k}), \\ \text{s.t. } & x_{0|k} = x_k, \\ & x_{\ell+1|k} = Ax_{\ell|k} + Bu_{\ell|k},\ \ell \in \{0, 1, \ldots, N-1\}, \\ & u_{\ell|k} \in \mathbb{U}_{\ell|k},\ \ell \in \{0, 1, \ldots, N-1\}, \end{aligned} \tag{5.41}$$

where $U_k = \begin{bmatrix} u_{0|k} & \ldots & u_{N-1|k} \end{bmatrix}^\top$ contains the predicted future control inputs. Moreover, $N \in \mathbb{N}_{\geq 1}$ is the prediction horizon, $l : \mathbb{R}^4 \times \mathbb{R} \to \mathbb{R}_{\geq 0}$ is the stage/running cost, $l_T : \mathbb{R}^4 \to \mathbb{R}_{\geq 0}$ is the terminal cost and $x_{\ell|k}, u_{\ell|k}$ denote the prediction of the state and input at time step $\ell + k$, made at time k, respectively. The time-varying set $\mathbb{U}_{\ell|k}$ is chosen such that it contains the constraints on the input in (5.38) and some additional constraints to ensure stability as introduced in the sequel.

The first input of the optimal U_k, denoted as $U_k^* = \begin{bmatrix} u_{0|k}^* & \ldots & u_{N-1|k}^* \end{bmatrix}^\top$, is then applied to the system, yielding the MPC policy $u_k = \mu_{MPC}(x_k)$ as a nonlinear function of the state, where

$$\mu_{MPC}(x_k) = u_{0|k}^*. \tag{5.42}$$

The input to the trajectory tracking dynamics in (5.15a)-(5.15b) is now obtained by solving the OCP at each $k \in \mathbb{N}$ for each scalar system $i \in \{1, 2, 3\}$ resulting in $\mu_{MPC,i}(x_k)$. Using these inputs as $\underline{s}^i(t) = \mu_{MPC,i}(x_k)$ for $t \in [t_k, t_{k+1})$ (see also (5.34)) in the linear system (5.33) yields $\underline{a}^i(t)$ and $a_d(t) = [\underline{a}^1(t), \underline{a}^2(t), \underline{a}^3(t)]^\top$.

In the rest of this section exact expressions will be provided for the OCP that ensure that the origin of (5.15a)-(5.15b) is globally asymptotically stable (GAS).

5.6.3 Low-gain feedback controller

The MPC law as proposed in [205] is used to stabilize the system (5.35)-(5.36). This MPC law is based on the fact that a globally stabilizing control law is known for this system that provides a quadratic Lyapunov function. By imposing extra constraints on the first step of the input, a decrease in the Lyapunov function is guaranteed.

This stabilizing control law will be introduced shortly, however it is first shown that there exists a control law that renders the origin of the system (5.35)-(5.36) globally asymptotically stable in the following lemma.

Lemma 5.7. *There exists a control law $u_k = u_k(x_k)$ that renders the origin of the system (5.35)-(5.36) globally asymptotically stable and that satisfies the input constraint $|u_k| \leq \Delta$, $k \in \mathbb{N}_{\geq 0}$.*

Proof. From [206] it follows that the necessary and sufficient conditions for a linear system subject to input saturation to be globally asymptotically stabilizable are

1. The pair (A, B) is stabilizable, and
2. A has as its eigenvalues in the closed unit disc.

The system (5.35)-(5.36) has four eigenvalues:

$$\lambda_1 = 1, \quad \lambda_2 = e^{-dh}, \quad \lambda_3 = \lambda_4 = e^{-h/\gamma}, \tag{5.43}$$

where $d, h, \gamma > 0$. These all lie in the closed unit disk, thus satisfying the second condition. Moreover, by computing the controllability matrix

$$\mathcal{C} = \begin{bmatrix} B & AB & A^2B & A^3B \end{bmatrix}$$

of the system and calculating its determinant yields that it is nonzero for all $d, h, \gamma > 0$, so that the system is controllable and thus stabilizable. □

A globally stabilizing control law, known as a low-gain controller, was proposed in [94] as

$$u_L = F_{\varepsilon(x_k)} x_k = -(R + B^\top P_{\varepsilon(x_k)} B)^{-1} B^\top P_{\varepsilon(x_k)} A x_k, \tag{5.44}$$

where $R \succ 0$ and $\varepsilon(x_k) > 0$ is the low-gain parameter, which is scheduled such that the controller provides global asymptotic stability. The dependency of $\varepsilon(x_k)$ on x_k is omitted hereafter for brevity. The matrix $P_\varepsilon \succ 0$ is determined by solving the parametric discrete-time algebraic Ricatti equation (DARE):

$$P_\varepsilon = A^\top P_\varepsilon A - A^\top P_\varepsilon B(R_L + B^\top P_\varepsilon B)^{-1} B^\top P_\varepsilon A + Q_\varepsilon, \tag{5.45}$$

with $Q_\varepsilon \succ 0$ a parametrized matrix such that $\lim_{\varepsilon \to 0} Q_\varepsilon = 0$ and $\frac{\mathrm{d}Q_\varepsilon}{\mathrm{d}t} > 0$, e.g., $Q_\varepsilon = \varepsilon I$. The low-gain parameter is scheduled as

$$\varepsilon(x_k) = \max \left\{ r \in (0, 1] \;\middle|\; (x_k^\top P_r x_k) \mathrm{tr}(P_r) \leq \frac{\Delta^2}{b} \right\}, \tag{5.46}$$

where $b = 2\mathrm{tr}(BB^\top)$, P_r is the unique positive definite solution of (5.45) with $\varepsilon = r$ and $\mathrm{tr}(\cdot)$ denotes the trace of a matrix. For this scheduling it is assumed that $R = I$, which can always be achieved by appropriate scaling.

In [206] it was shown that the low-gain controller in (5.44) with P_ε the solution of (5.45) and the scheduling (5.46) provides global asymptotic stability for linear discrete-time systems subject to input saturation. The following theorem states the same result for the system under consideration.

Theorem 5.8 (cf. [206]). *The system* (5.35) *with input* (5.44) *with* $R = I$, P_ε *the solution of* (5.45) *and the scheduling given in* (5.46) *satisfies the constraint* (5.37) *and is globally asymptotically stable.*

5.6.4 Model predictive controller

Next, the same Lyapunov function candidate as was used in the proof of Theorem 5.8 is considered, namely $V_k = V(x_k) = x_k^\top P_k x_k$ and the shorthand notations $P_k = P_{\varepsilon(x_k)}$, $Q_k = Q_{\varepsilon(x_k)}$, $F_k = F_{\varepsilon(x_k)}$ and $R = I$, however the evolution along trajectories for any $|u_k| \leq \Delta$ is evaluated as

$$\begin{aligned} V_{k+1} - V_k &= -x_k^\top Q_k x_k - u_k^\top u_k + x_{k+1}^\top (P_{k+1} - P_k) x_{k+1} \\ &\quad + (u_k - F_k x_k)^\top (I + B^\top P_k B)(u_k - F_k x_k), \\ &= -x_k^\top Q_k x_k + x_{k+1}^\top (P_{k+1} - P_k) x_{k+1} \\ &\quad + (u_k - F_k x_k)^\top B^\top P_k B (u_k - F_k x_k) \\ &\quad - x_k^\top F_k^\top F_k x_k - 2 x_k^\top F_k^\top (u_k - F_k x_k). \end{aligned}$$

In [205] it was established that $V_{k+1} - V_k$ and $x_{k+1}^\top (P_{k+1} - P_k) x_{k+1}$ cannot have the same sign for the scheduling in (5.46), thus a decreasing Lyapunov function can be ensured by having

$$2 x_k^\top F_k (u_k - F_k x_k) - (u_k - F_k x_k)^\top B^\top P_k B (u_k - F_k x_k) \geq 0.$$

By writing this constraint as

$$\left[2 x_k^\top F_k - (u_k - F_k x_k)^\top B^\top P_k B \right] (u_k - F_k x_k) \geq 0,$$

it becomes clear that the term in the square brackets must have the same sign as $(u_k - F_k x_k)$[1]. This can be achieved by requiring

$$\begin{aligned} \operatorname{sign}(u_k - F_k x_k)(u_k - F_k x_k) &\geq 0, \\ \operatorname{sign}(u_k - F_k x_k)(2F_k - B^\top P_k B (u_k - F_k x_k)) &\geq 0, \end{aligned}$$

or by using the following linear constraints on the input as in [205]:

$$\begin{aligned} \operatorname{sign}(F_k x_k)(u_k - F_k x_k) &\geq 0, \\ \operatorname{sign}(F_k x_k)(2F_k - B^\top P_k B (u_k - F_k x_k)) &\geq 0. \end{aligned}$$

[1] When both terms are nonzero

The MPC law then consists of solving the following optimal control problem (OCP) at each time step

$$\begin{aligned}
\min_{U_k} \;& J(x_k, U_k) = l_T(x_{N|k}) + \sum_{\ell=0}^{N-1} l(x_{\ell|k}, u_{\ell|k}) \\
\text{s.t. } & x_{0|k} = x_k \\
& x_{\ell+1|k} = Ax_{\ell|k} + Bu_{\ell|k},\ \ell \in \{0, \dots, N-1\} \\
& |u_{\ell|k}| \leq \Delta,\ \ell \in \{0, \dots, N-1\} \\
& 0 \leq \operatorname{sign}(F_k x_{0|k})(u_{0|k} - F_k x_{0|k}), \\
& 0 \leq \operatorname{sign}(F_k x_{0|k})(2F_k - B^\top P_k B(u_{0|k} - F_k x_{0|k})).
\end{aligned} \tag{5.47}$$

The first input is then applied to the system according to (5.42), where $u^*_{0|k}$ is obtained from (5.47). The following result asserts the stability of the MPC policy [205, 212]:

Theorem 5.9. *The origin of the system* (5.35) *with input* (5.42)*, obtained from* (5.47)*, for any $l : \mathbb{R}^4 \times \mathbb{R} \to \mathbb{R}_{\geq 0}$, $l_T : \mathbb{R}^4 \to \mathbb{R}_{\geq 0}$ and $N \in \mathbb{N}_{\geq 1}$, is globally asymptotically stable.*

Proof. The optimization in (5.47) guarantees, by construction, that $V_{k+1} - V_k < 0$, for sufficiently small ϵ and all $x_k \neq 0$. Moreover, the optimization problem is always feasible, since a known solution in the form of (5.44) is feasible. □

Remark 5.10. *Note that only the first input of the horizon $u_{0|k}$ in* (5.47) *needs to be constrained by Δ, whereas for the rest of the horizon the constraint on the input can be chosen freely, whilst still guaranteeing feasibility and stability of the closed-loop system. This is because only the first input in the horizon is applied to the system, and there always exists a solution in the form of the low-gain feedback controller.*

Remark 5.11. *Note that the choice of cost function in* (5.47) *is not mentioned in the above theorem, i.e., any cost function and any horizon $N \in \mathbb{N}_{\geq 1}$ can be chosen whilst still guaranteeing global asymptotic stability of the system* (5.35)*. In fact, global stability follows from the constraints imposed in* (5.47)*. However, the performance of the MPC strategy does depend on the choice of cost function, since the controller is free to generate an input that satisfies the input constraint and makes the Lyapunov function decrease in the next timestep, but yields a larger decrease in the Lyapunov function than the low-gain feedback controller.*

Remark 5.12. *Although the choice of cost function in* (5.47) *is not relevant for providing stability guarantees in Theorem 5.9, in the stability proof of the cascade in Section 5.7 uniqueness of solutions is required. Therefore, the cost function in* (5.47) *is limited to be convex in the rest of this chapter. Note that this is not a major restriction, since in practice convex cost functions are typically preferred as they often lead to lower computation times.*

5.6.5 Stability of the continuous-time system

Although GAS has been concluded for the system (5.35), care must be taken in concluding GAS for the continuous-time system in (5.30). The state in (5.35) is exactly the state (5.30) due to the ZOH and the exact discretization used. However, the behavior of the continuous-time system in between sampling times needs to be considered as well.

Consider the continuous-time linear system in (5.30), which can be written as

$$\dot{\bar{x}}(t) = \bar{A}x(t) + \bar{B}u(t), \tag{5.48}$$

where $\bar{x}(t) = [\tilde{p}^\top, \tilde{v}^\top, a_d^\top, j^\top]^\top \in \mathbb{R}^{12}$ and the input is generated by solving (5.47) for each axis and setting

$$u(t) = \begin{bmatrix} \mu_{MPC,1}(\bar{x}(kh)) \\ \mu_{MPC,2}(\bar{x}(kh)) \\ \mu_{MPC,3}(\bar{x}(kh)) \end{bmatrix}, \text{ for } t \in [kh, kh+h). \tag{5.49}$$

Then, for $t \in [kh, kh+h]$:

$$\bar{x}(t) = e^{\bar{A}(t-kh)}x(kh) + \int_0^{t-kh} e^{\bar{A}s}\,\mathrm{d}s\ \bar{B}u(kh),$$

and therefore

$$\begin{aligned} \|\bar{x}(t)\| &= \|e^{\bar{A}(t-kh)}\bar{x}(kh) + \int_0^{t-kh} e^{\bar{A}s}\,\mathrm{d}s\ \bar{B}u(kh)\| \\ &\leq \|e^{\bar{A}(t-kh)}\|\|\bar{x}(kh)\| + \|\int_0^{t-kh} e^{\bar{A}s}\,\mathrm{d}s\bar{B}\|\,\|u(kh)\| \\ &\leq c_1\|\bar{x}(kh)\| + c_2\,\|u(kh)\|\,. \end{aligned}$$

As both $\|\bar{x}(kh)\|$ and $\|u(kh)\|$ go to zero it follows that $\lim_{t\to\infty}\|\bar{x}(t)\| = 0$. Therefore, the system (5.48) in closed loop with the ZOH-input given by (5.42) is globally asymptotically stable.

5.7 Cascaded trajectory tracking controller

In the previous sections a desired acceleration for asymptotic stability of the position dynamics in the outer-loop problem was derived together with a controller that uses the torque and thrust to acquire this desired control action asymptotically in the inner-loop problem. In order to conclude stability of the closed-loop system using the proposed strategy, the cascaded system is now examined. To conclude stability the following theorem will be used:

Theorem 5.13 (cf. [166],[123]). *Consider a cascaded system $\dot{x} = f(t,x)$ with $f(t,0) = 0$, for all $t \in \mathbb{R}_{\geq 0}$, that can be written as*

$$\dot{x}_1 = f_1(t,x_1) + g(t,x_1,x_2)x_2, \tag{5.50a}$$
$$\dot{x}_2 = f_2(t,x_2), \tag{5.50b}$$

with x_1 and x_2 taking values in $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively. This system is a cascade of the systems

$$\dot{x}_1 = f_1(t,x_1) \tag{5.51}$$

and (5.50b). *If the origins of the systems* (5.51) *and* (5.50b) *are UGAS and the solutions to* (5.50) *remain bounded, then the origin of the system* (5.50) *is UGAS.*

Remark 5.14. *In [166] it is assumed that $f_1(t,x_1)$ is continuously differentiable in (t,x_1) and $f_2(t,x_2)$, $g(t,x_1,x_2)$ are continuous in their arguments, and locally Lipschitz in x_2 and (x_1,x_2) respectively. However, only uniqueness of solutions (see Remark 5.12) is used in the proof of Theorem 5.13, so that the same theorem can be employed here in the stability proof for this system.*

Consider the dynamics (5.7) and reference (5.9) in closed loop with the inputs (5.18), (5.28) and (5.49). The closed-loop system is then given by

$$\dot{\tilde{p}} = \tilde{v}, \tag{5.52a}$$
$$\dot{\tilde{v}} = -D\tilde{v} + a_d(t) + TR(I - R_e^\top)e_3, \tag{5.52b}$$
$$\dot{a}_d = -\frac{1}{\gamma}(a_d + j), \tag{5.52c}$$
$$\dot{j} = -\frac{1}{\gamma}(j + u), \tag{5.52d}$$
$$\dot{R}_e = R_e S(\omega_e), \tag{5.52e}$$
$$J\dot{\omega}_e = -K_\omega \omega_e + K_R \sum_{i=1}^{3} k_i (e_i \times R_e^\top e_i). \tag{5.52f}$$

Theorem 5.15. *The origin $(\tilde{p},\tilde{v},R_e,\omega_e) = (0,0,I,0)$ of* (5.52) *is UaGAS.*

Proof. First note that (5.52) is a cascade of the systems ((5.52a)-(5.52d)) and ((5.52e)-(5.52f)). Since ((5.52e)-(5.52f)) is UaGAS, we consider our stability analysis on $\mathbb{R}^6 \times G$, where $G \subset SO(3) \times \mathbb{R}^3$ is the uniformly almost global region of attraction of ((5.52e)-(5.52f)).

Since ((5.52e)-(5.52f)) and (5.48) are UGAS on $\mathbb{R}^6 \times G$, it is only required to show that the solutions remain bounded to conclude UGAS of (5.52) on $\mathbb{R}^6 \times G$ according to Theorem 5.13. The dynamics ((5.52e)-(5.52f)) are bounded since

Description	Symbol	Value
Gravitational constant	g	9.81
Inertia matrix	J	$\text{diag}(0.08, 0.08, 0.15)$
Translational drag coefficients	D	$\text{diag}(3.3, 1.7, 1.7)$
Gyroscopic torques	τ_g	$[0, 0, 0]^\top$
Cross drag coefficients	A	$0.1I$
Rotational Drag coefficients	C	$0.5I$
Maximum torque	$T_{\max}$	$1.6g = 15.696$

Table 5.1: Quadcopter parameters used in simulations.

they are UGAS on G and for ((5.52a)-(5.52d)) the boundedness of solutions is shown in Appendix 5.A. The result follows from Theorem 5.13. Moreover, the origin $(\tilde{p}, \tilde{v}, R_e, \omega_e) = (0, 0, I, 0)$ is UGAS except for initial conditions in a set of measure zero, so that UaGAS of (5.52) can be concluded according to Definition 5.2. □

Finally, it is shown that when UaGAS of (5.52) is realized, a solution to Problem 1 is found.

Corollary 5.16. *The controller consisting of the inputs obtained from* (5.18), (5.28) *and* (5.49) *solves Problem 1.*

Proof. From Theorem 5.15 it directly follows that $(\tilde{p}(t), \tilde{v}(t)) \to (0, 0)$, which combined with ((5.15a)-(5.15b)) results in $a_d(t) \to 0$. Then, by using (5.20), it follows that $z_{B,d}(t) \to e_3$ and from (5.21) it then follows that $R_d(t) \to I$. Combining this with $R_e(t) \to I$ results in $\tilde{R}(t) \to I$. Finally, since $\dot{a}_d(t) \to 0$, it follows that $\omega_d(t) \to 0$, which, combined with $\omega_e(t) \to 0$ and (5.27), results in $\tilde{\omega}(t) \to 0$, concluding the proof. □

5.8 Numerical case study

In this section the effectiveness of the proposed strategy is illustrated through numerical examples. The dynamics in (5.7) are considered with the parameters provided in Table 5.1. For the reference trajectory, we consider

$$\bar{p}(t) = \begin{bmatrix} \cos(t) & \sin(t) & -5\sin(2t) \end{bmatrix}^\top, \tag{5.53}$$

together with a heading angle $\psi = 0.2t$, which is the angle between the projection of x_B onto the $x_W - y_W$ plane and the x_W axis, see Figure 5.1. This angle is

only well defined when the thrust vector is limited by $0 \leq z_W^\top \bar{z}_B(t) < 1$, for $t \in \mathbb{R}_{\geq 0}$, i.e., when the quadcopter is upright, which is the case for this reference trajectory. This trajectory fully defines the states and inputs of the reference model in (5.9) by following the differential flatness method employed in [66]. The initial conditions are set to

$$\begin{aligned} p(0) &= \begin{bmatrix} 0 & 1 & -6 \end{bmatrix}^\top, \\ v(0) &= \bar{v}(0), \\ R(0) &= R_x\left(170\frac{\pi}{180}\right) R_y\left(30\frac{\pi}{180}\right) R_z\left(20\frac{\pi}{180}\right), \\ \omega(0) &= \bar{\omega}(0), \end{aligned}$$

where $R_x(\theta)$ denotes a rotation around the x-axis according to

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(-\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}$$

and R_y, R_z are rotations about the y and z axes defined similarly.

For all simulations the inner-loop gains are set to $K_\omega = 30J$, $K_R = 70J$ with J the inertia matrix and $[k_1, k_2, k_3] = [0.9, 1, 1.1]$. Furthermore, a sample time of $h = 0.1$ seconds is used and $\gamma = 0.01$ is set for the model used in the MPC controller.

5.8.1 Outer-loop verification

First the scalar, linear discrete-time system (5.35) is simulated for the z-axis, with the input generated by solving the OCP in (5.47) and transforming back to the system using (5.42). The following quadratic cost function is used

$$J = x_{N|k}^\top Q_N x_{N|k} + f_N x_{N|k} + \sum_{\ell=0}^{N-1} x_{\ell|k}^\top Q x_{\ell|k} + f x_{\ell|k} + u_{\ell|k}^\top R \hat{u}_{\ell|k} + g u_{\ell|k}, \quad (5.54)$$

with $Q = \mathrm{diag}(100, 1, 1, 1)$, $Q_N = 0.01Q$, $R = 0.1$ and $f_N = f = g = -1$. The settings for the low-gain controller that provides the constraints by solving the DARE in (5.45) are $Q_\varepsilon = \varepsilon Q$ and $R_L = 0.1$, where ε is obtained from the scheduling in (5.46). Note that the settings for the low-gain controller can be chosen independently of those of the cost function (cf. Remarks 5.11 and 5.12).

To show the importance of the stabilizing constraints, the horizon is set to $N = 1$ and the solutions for the OCP are computed with and without the stabilizing constraints. Figure 5.5 shows the position errors for both cases, showing that the stabilizing constraints ensure stability and fast convergence. The scheduled low-gain parameter obtained from (5.46) and corresponding Lyapunov function are shown for both cases in Figure 5.6, confirming that the constraint on the first input only results in a decreasing Lyapunov function for the system.

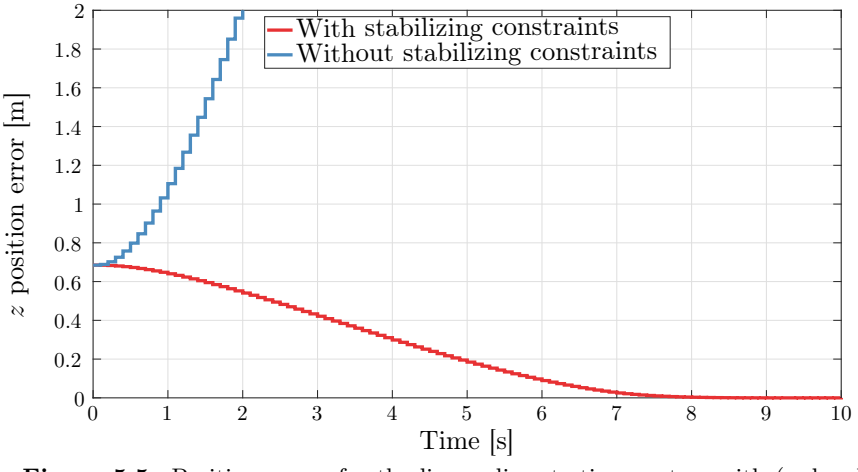


Figure 5.5: Position errors for the linear discrete-time system with (red, solid) and without (blue, dashed) constraints.

5.8.2 Quadcopter Simulation

A simulation of the cascaded system, consisting of the outer and inner loop, is performed next. Most settings remain the same for each subsystem corresponding to the x, y, z axes, except for the prediction horizon and the terminal cost, which are set to $N = 20$ and $Q_N = 10Q_d$, where Q_d is obtained by solving the regular DARE, i.e. (5.45) for $\varepsilon = 1$, with Q and R from the cost function. Moreover, f_N, f and g are set to zero, i.e., $f_N = f = g = -1$.

The resulting trajectory is visualized in Figure 5.10, where it can be seen that the quadcopter is able to recover from an upside-down initial attitude and converges to the reference trajectory. The corresponding position and attitude errors are shown in Figures 5.7 and 5.8, respectively. For the attitude errors the same metric as in the proof of Theorem 5.13 is used, i.e., $\gamma(R_1, R_2) = \|R_1 - R_2\|_F$ and the distance of both the desired attitude R_d and the reference attitude error $\tilde{R}$ to I are evaluated. Lastly, Figure 5.9 displays the thrust during the trajectory, showing that it stays below the maximum value and thus satisfies the constraint (5.8).

5.9 Conclusions

In this chapter a new cascaded controller was presented, that enables guaranteed trajectory tracking for quadcopters while taking into account the limited thrust capabilities of quadcopters. The method uses the differential flatness property of the quadcopter dynamics in combination with a uniformly almost globally asymptotically stable inner-loop controller and a novel, MPC-based, uniformly

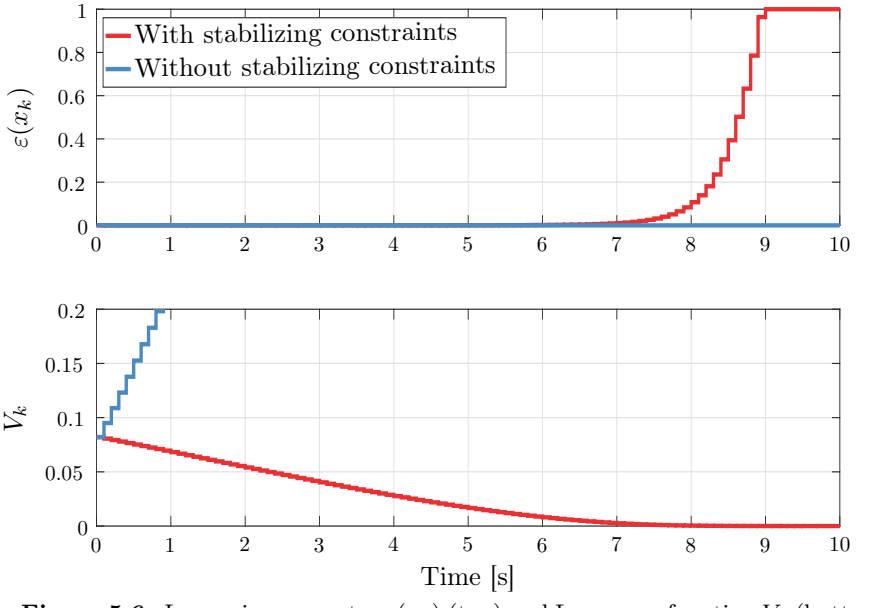


Figure 5.6: Low-gain parameter $\varepsilon(x_k)$ (top) and Lyapunov function V_k (bottom) for the linear discrete-time system with (red, solid) and without (blue, dashed) constraints.

globally asymptotically stable outer-loop controller. This combination allows for providing convergence guarantees for trajectory tracking, incorporation of future reference information and constraint handling. Moreover, the methodology used allows the MPC strategy to be formulated as three quadratic problems, each with just four states and one input, and only linear constraints on the first input of the horizon, allowing for fast computation times. The advantages of the method are shown in a numerical case study.

Suggestions for future research include the development of less conservative constraints on the desired acceleration a_d, by, for example, relaxing the need for decoupling and/or providing tighter bounds on the evolution of the desired acceleration in between sample times. An experimental case study and demonstration of the real-time capability of the suggested control strategy is also of interest. Finally, the incorporation of measurement inaccuracies in the development of the controller will make the method more applicable in practice.

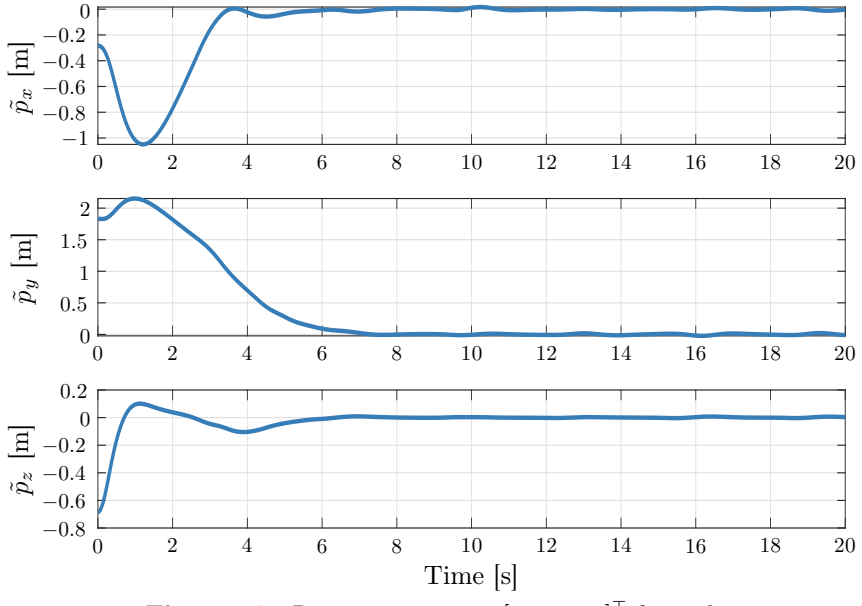


Figure 5.7: Position errors $\tilde{p} = [\tilde{p}_x, \tilde{p}_y, \tilde{p}_z]^\top$ for each axis.

5.A Proof of theorem 5.15

Consider the dynamics in (5.52), which, using (5.48), can be written as

$$\dot{\bar{x}}(t) = \bar{A}\bar{x}(t) + \bar{B}u(\bar{x}(kh)) + \begin{bmatrix} 0 \\ I_3 \\ 0 \\ 0 \end{bmatrix} TR(I - R_e^T)e_3$$

for $t \in [kh, kh + h)$ with $\bar{x}(t) \in \mathbb{R}^{12}$. Using the observation that the x, y and z dynamics were decoupled, these dynamics can be considered separately, and using the fact that T is (globally) bounded and the (R_e, ω_e) dynamics are locally exponentially stable, only boundedness (and therefore stability) of the following dynamics needs to be studied:

$$\dot{x}(t) = Ax(t) + B\mu_{MPC}(x(kh)) + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \gamma(t)$$

for $t \in [kh, kh + h)$ with $x(t) \in \mathbb{R}^4$, A and B are the system matrices for a single axis and where $|\gamma(t)| \leq ce^{-\lambda t}$ for some constants $\lambda > 0$ and $c > 0$, where c depends on the initial condition $(R_e(t_0), \omega_e(t_0))$.

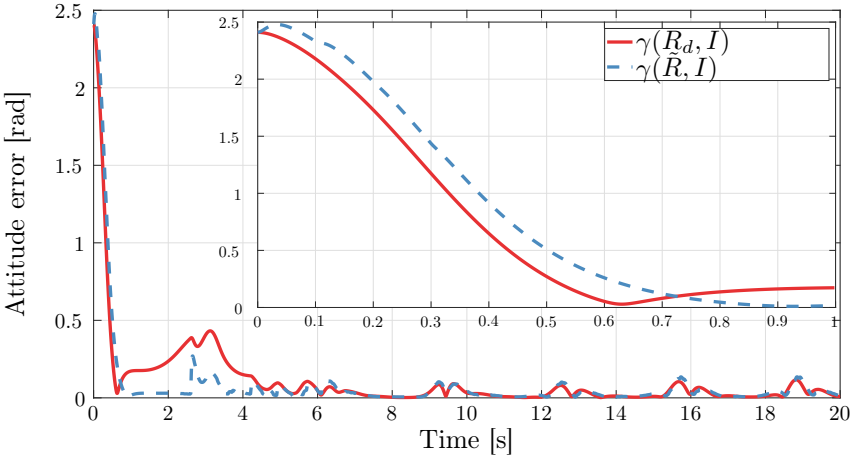


Figure 5.8: Attitude errors for the desired attitude $\gamma(R_d, I) = \|R_d - I\|_F$ (red, solid) and the reference attitude $\gamma(\tilde{R}, I) = \|R_d - I\|_F$ (blue, dashed). The insert highlights the first second.

The aim is to achieve this by showing that the function $V(x) = x^\top P_{\varepsilon(x)} x$ remains bounded, by considering this function at sample times. The following properties will be used, which follow from the scheduling in (5.46) (cf. [177], Theorem 4.52):

1. For any $x_1, x_2 \in \mathbb{R}^n$: $x_1^\top P_{\varepsilon(x_1)} x_1 \leq x_2^\top P_{\varepsilon(x_2)} x_2$ implies $\varepsilon(x_1) \geq \varepsilon(x_2)$.
2. P_ε is continuously differentiable with respect to ε and $\frac{\mathrm{d}P_\varepsilon}{\mathrm{d}\varepsilon} > 0$ for any $\varepsilon \in (0, 1]$.

It follows that:

$$x(kh+h) = [Ax(kh) + B\mu_{MPC}(x(kh))] + \int_0^h e^{\bar{A}s}\gamma(kh+h-s)\,\mathrm{d}s \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}.$$

Furthermore,

$$\left\| \int_0^h e^{\bar{A}s}\gamma(kh+h-s)\,\mathrm{d}s \right\| \leq c_4 e^{-\lambda kh}.$$

From the proof of Theorem 5.9 it is known that

$$V(Ax(kh) + B\mu_{MPC}(x(kh))) \leq V(x(kh)).$$

Assume that

$$V(x(kh+h)) \geq V(Ax(kh) + B\mu_{MPC}(x(kh))).$$

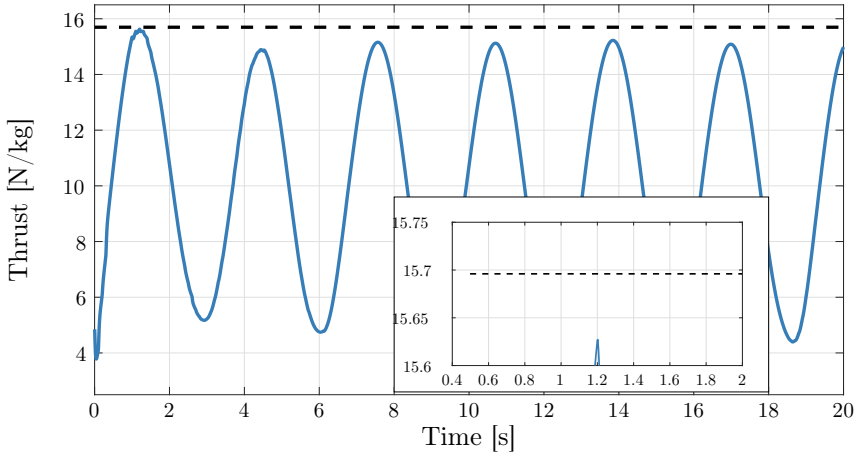


Figure 5.9: Thrust T during the trajectory (blue, solid) and the maximum thrust $T_{\max}$ (black, dashed). The insert shows the thrust stays below the maximum value and thus satisfies the constraint (5.8).

Then $\varepsilon(x(kh+h)) \leq \varepsilon(Ax(kh) + B\mu_{MPC}(x(kh)))$ and $P_{\varepsilon(x(kh+h))} \leq P_{\varepsilon(Ax(kh)+B\mu_{MPC}(x(kh)))}$. Therefore,

$$\begin{aligned} V(x(kh+h)) &= x(kh+h)^\top P_{\varepsilon(x(kh+h))} x(kh+h) \\ &\leq x(kh+h)^\top P_{\varepsilon(Ax(kh)+B\mu_{MPC}(x(kh)))} x(kh+h) \\ &\leq V(x(kh)) + 2\sqrt{\|P_{\varepsilon(Ax(kh)+B\mu_{MPC}(x(kh)))}\|V(x(kh))}c_4 e^{-\lambda kh} \\ &\quad + \|P_{\varepsilon(Ax(kh)+B\mu_{MPC}(x(kh)))}\|c_4^2 e^{-2\lambda kh} \\ &\leq V(x(kh)) + c_5\sqrt{V(x(kh))}e^{-\lambda kh} + c_6 e^{-2\lambda kh}. \end{aligned}$$

In case

$$V(x(kh+h)) < V(Ax(kh) + B\mu_{MPC}(x(kh)))$$

the latter bound holds even with $c_5 = c_6 = 0$, since

$$V(Ax(kh) + B\mu_{MPC}(x(kh))) \leq V(x(kh)).$$

Observe that the difference equation

$$F_{k+1} = F_k + 2\alpha\sqrt{F_k}e^{-\lambda kh} + \alpha^2 e^{-2\lambda kh}$$

is solved by

$$F_k = \left(\alpha\frac{1-e^{-\lambda kh}}{1-e^{-\lambda h}} + \sqrt{F_0}\right)^2.$$

Therefore, it is obtained that

$$\begin{aligned} V(x(kh)) &\leq \left(\alpha \frac{1-e^{-\lambda kh}}{1-e^{-\lambda h}} + \sqrt{V(x(0))}\right)^2 \\ &\leq \left(\frac{\alpha}{1-e^{-\lambda h}} + \sqrt{V(x(0))}\right)^2, \end{aligned}$$

where $\alpha = \max(c_5/2, \sqrt{c_6})$.

Since $V(x(kh))$ is bounded, boundedness of $x(kh)$ is obtained, from which boundedness of $x(t)$ follows as well.

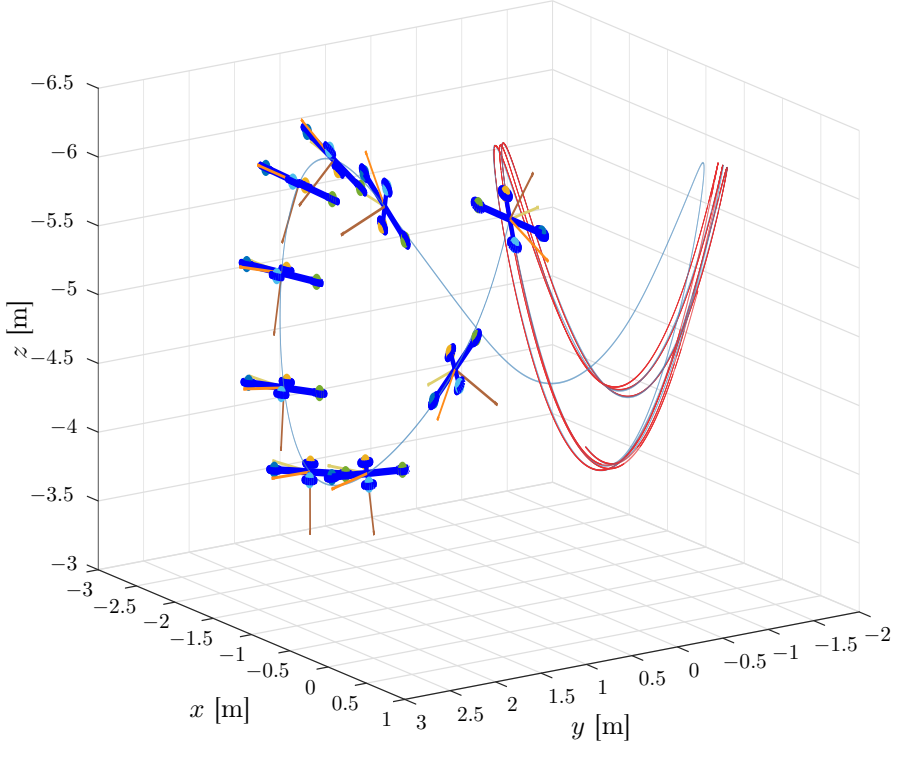


Figure 5.10: Three dimensional plot of the quadcopter trajectory. The position trajectory of the reference and quadcopter are shown by the red and blue lines, respectively. Snapshots of the quadcopter during the first 3 seconds of the trajectory are shown, where the orange, yellow and brown lines depict the x_B, y_B and z_B axes, respectively. The right most snapshot is the initial condition. Note that the quadcopter is able to recover from an initial upside-down attitude and converges to the reference.

Chapter 6

Time and Energy Efficient Descent Trajectories for Quadcopters that Avoid the Vortex Ring State

In this chapter, the problem of computing time-optimal and energy-optimal vertical descent trajectories for quadcopters, while avoiding the Vortex Ring State (VRS), is investigated. The VRS is a region in the velocity space of the quadcopter in which disturbances on the thrust produced by the propellers result in a reduced control effectiveness and can eventually lead to a crash. A new model is proposed for the VRS dynamics for quadcopters, which is shown to be more complete in comparison with existing approaches. Based on this novel model, time-optimal and energy-optimal trajectories are computed using GPOPS II and compared in a numerical case study.

This chapter is based on [10]

6.1 Introduction

The Vortex Ring State (VRS) is a well-known phenomenon in helicopters [190]. It captures a loss of thrust due to the formation of vortices at the blade tips during fast descent trajectories, see Fig. 6.1. In normal operation, the horizontal rotors of a quadcopter produce a downward induced flow (Fig. 6.1a). When the quadcopter starts to descend, this flow is disturbed by the upward flow of the descending motion. This results in disruption of the laminar flow around the rotor disks. If the descent rate approaches the rotor induced velocity, vortices form around the rotor in a vortex ring (Fig. 6.1b). These vortices cause disturbances in the thrust and torque that the rotors produce. This in turn can lead to reduced control effectiveness and even total loss of control of the quadcopter, with a crash as a possible consequence. The VRS can also occur in cases when the quadcopter is not descending fast, but the wind causes the quadcopter to enter the VRS. A well-known example of the latter scenario can take place in the industrial inspection of chimneys [134], where the quadcopter can enter VRS even when it is hovering due to a large upward airflow caused by the wind. If the vertical velocity is further increased, beyond the VRS region, the Turbulent Wake State (TWS) is entered, which has an even more detrimental effect on the control effectiveness than the VRS (Fig. 6.1c). At some point the flow around the rotors will even become laminar again, although in the direction opposite of that in normal operation (Fig. 6.1d). The velocity region where this occurs is referred to as the Windmill Brake State (WBS). In the WBS, helicopters can perform maneuvers that allow them to land safely even when motor failure has occurred, by changing the pitch of the rotor blades. In quadcopters, the propellers are typically fixed-pitch, so that it is deemed important to avoid this region as well. In the rest of this chapter the complete set of regions that need to be avoided, i.e. the VRS, TWS and WBS, will be referred to simply as the VRS.

The characteristics of the VRS in helicopters have been studied in detail [2, 100, 52, 209, 200, 146]. However, in quadcopters not much is known about this effect and there are only a few recent studies [39, 68, 193, 192].

In particular, in [193] a first attempt to describe the VRS in quadcopters is made. The effect is modeled and experimentally validated, and a constraint in the velocity space of the quadcopter's propeller frames is proposed. Time-optimal descent trajectories that avoid this constraint are then provided by solving a minimum time-optimal control problem using GPOPS [167], a general purpose optimal control solver. Although the results are impressive, the velocity in the propeller frame as a result of the angular velocity and the offset of the propeller from the body-fixed quadcopter frame are not taken into account. In [192], this component of the blade disk velocity is taken into account, however only pure yaw rotation is considered to avoid the VRS.

In this chapter this gap in the literature will be addressed and the full dynam-

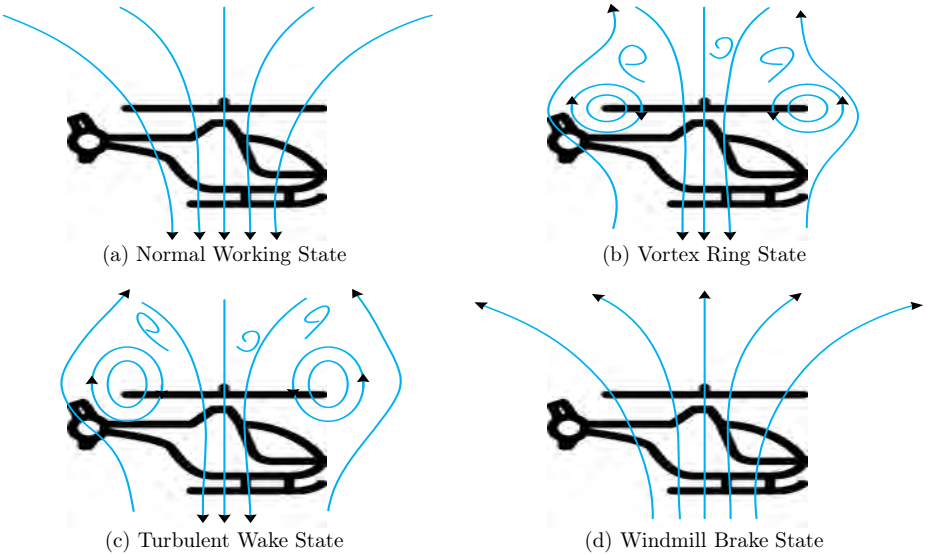


Figure 6.1: Schematic depiction of a helicopter in different operating regimes, based on [165]. The descent rate of the helicopter increases through (a)-(d).

ics of the propeller frames is considered, which results in a more difficult problem setting due to the coupling between the rotational and translational velocities in the constraint. Moreover, the challenge of computing not only time-optimal trajectories is considered, but energy-optimal descent trajectories as well.

The computation of time-optimal trajectories has been studied extensively in quadcopters following various approaches, including nonlinear programming [95], real-time planning subject to constraints [220], the PRONTO algorithm with barrier functions [186], piecewise polynomial trajectories formulated as a quadratic program (QP) [96] and direct multiple shooting [72]. A performance benchmark is provided in [89].

The computation of energy-optimal trajectories has also received quite some attention from the research community. For instance, in [132] energy-optimal trajectories are computed using GPOPS II by assuming that the quadcopter undergoes three phases, namely: acceleration, constant speed, and deceleration. In [214] models for the brushless direct current (BLDC) motors and the battery typically used in quadcopters are provided, as well as a calculation of the energy consumed by the quadcopter under deterministic wind disturbances. In [38] the power usage of the drone is calculated by explicitly considering the efficiency of the battery depending on its state-of-charge. An empirical study for a specific quadcopter's energy usage was performed in [1], resulting in a model of the total energy consumption with specific values that are only applicable to that quad-

copter, making it unsuitable in this study. However, from the results in [1] it is clear that the energy used for computation and communication processes is low compared to that of flying, which is assumed here as well. In [187] a comparison between drone-based delivery and truck-based delivery is provided. The potential of quadcopters to be more efficient in commercial delivery applications is shown, showcasing the importance of energy-efficient planning.

The contribution in this chapter is twofold; first a new, improved model for the VRS and corresponding constraint are developed, and, second time-optimal and energy-optimal trajectories that take into account the VRS are computed. To the best of the authors' knowledge, these problems have not been considered elsewhere. Moreover, the energy-optimal trajectories are considered with free end time, in contrast to other recent papers [150, 213, 214] that considered fixed end time. A extensive numerical case study is provided as well.

The remainder of the chapter is organized as follows. In Section 6.2 the considered quadcopter dynamics, including a BLDC motor model, are presented, together with a measure of the energy usage of the quadcopter and a problem formulation is given. Section 6.3 provides a new model of the blade-disk dynamics and compares it to the previously used methods. Section 6.4 presents the use of GPOPS II to solve the trajectory planning problems and Section 6.5 provides the numerical case study. The chapter is concluded in Section 6.6 and several directions for future work are proposed.

6.2 Quadcopter Model

6.2.1 Dynamic model of quadcopter

Let W denote a right-handed inertial (or world) frame according to the North-East-Down (NED) convention, with unit vectors along the axes denoted by $\{x_W, y_W, z_W\}$. Let B denote a right-handed body-fixed frame with unit vectors $\{x_B, y_B, z_B\}$, where these vectors are the axes of B represented in W. Let D_i, $i \in \{1, 2, 3, 4\}$ denote right-handed frames centered at each of the four propeller hubs and let $l_i = [l_{i,x}, l_{i,y}, 0]^\top \in \mathbb{R}^3$ denote the position of the origin of frame D_i with respect to B. The origin of the body-fixed frame coincides with the center of mass of the quadcopter, and z_B is aligned with z_W and the gravitational vector when the quadcopter is at hover, see Fig. 6.2. The orientation of B with respect to W is represented by the rotation matrix $R = [x_B, y_B, z_B] \in SO(3)$. Let $\omega = [\omega_1, \omega_2, \omega_3]^\top \in B$ denote the angular velocities of B relative to W, expressed in B. The position and linear velocity of the center of mass of the quadcopter with respect to W are denoted by $p = [p_x, p_y, p_z]^\top$ and $v = [v_x, v_y, v_z]^\top$, respectively.

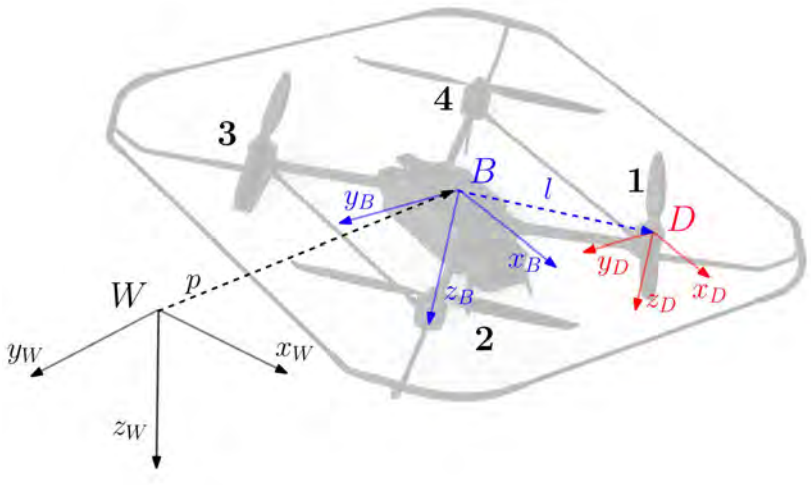


Figure 6.2: The inertial W, body-fixed B and blade disk D_1 frames.

The model used was developed in [66] and is given by the equations

$$\dot{p} = v, \tag{6.1a}$$

$$\dot{v} = gz_W - Tz_B - RD_rR^\top v, \tag{6.1b}$$

$$\dot{R} = RS(\omega), \tag{6.1c}$$

$$J\dot{\omega} = \tau - S(\omega)J\omega - \tau_r. \tag{6.1d}$$

The forces acting on the translational dynamics of the quadcopter consist of the gravity, given by gz_W, where g is the gravitational constant, the mass-normalized thrust force $-Tz_B$, where $T \geq 0$ denotes the magnitude of the combined thrust of the four propellers, and a drag force as a result of rotor drag where $D_r = \text{diag}(d_x, d_y, d_z)$, $d_x, d_y, d_z > 0$ are the mass-normalized rotor drag coefficients.

The rotation of the quadcopter is characterized by the attitude kinematics given in (6.1c), where $S(a)$ represents a skew-symmetric matrix such that $S(a)b = a \times b$ for any vectors $a, b \in \mathbb{R}^3$ and the dynamics given in (6.1d), where $J \in \mathbb{R}^{3\times3}$ is the inertia matrix, $\tau_r = J_r S(\omega)\Omega_r e_3 \in \mathbb{R}^3$ are torques resulting from the inertia of the propellers, J_r is the inertia of the propeller and motor, $\Omega_r = \sum_{i=1}^{4} \omega_{r,i}$, $\omega_{r,i}$ is the angular velocity of propeller i and $\tau = [\tau_1, \tau_2, \tau_3]^\top \in \mathbb{R}^3$ is the torque input.

Alternatively, the orientation of the quadcopter can be represented using quaternions instead of rotation matrices as considered above. The use of quaternions over rotation matrices results in numerical advantages, which is why quaternions are used in the computation of trajectories in Section 6.5. In order to provide more insight in the quadcopter dynamics and the VRS, rotation matrices will be used for now and the introduction of the quaternion form of the dynamics will be postponed to Section 6.4.

6.2.2 Propeller thrust relation

The thrust and torques exerted on the quadcopter originate from the thrust generated by each of its four propellers, which in turn are (typically) powered by BLDC motors. The thrust and torque generated by each of the four propellers are modeled as

$$T_{p,i} = k_T \omega_{r,i}^2, \quad \text{for } i \in \{1,2,3,4\}, \tag{6.2}$$

$$\tau_{p,i} = k_\tau \omega_{r,i}^2, \quad \text{for } i \in \{1,2,3,4\}, \tag{6.3}$$

where $T_{p,i}$ and $\tau_{p,i}$ are the thrust and torque generated by each propeller, respectively. The constants k_T and k_τ are the thrust and aerodynamic drag factors of the propellers and are given by $k_T = \frac{C_T}{m}\rho A r^2$, $k_\tau = C_Q \rho A r^3$, where m is the mass of the vehicle, $A = \pi r^2$ and r are the blade disk area and radius, respectively, ρ is the air density, $C_Q = C_T\sqrt{C_T/2}$ is the propeller torque coefficient and $C_T > 0$ is the propeller thrust coefficient.

The propeller speeds are then related to the thrust and torque of the quadcopter as

$$\begin{bmatrix} T \\ \tau \end{bmatrix} = \begin{bmatrix} k_T & k_T & k_T & k_T \\ l_y k_T & -l_y k_T & -l_y k_T & l_y k_T \\ l_x k_T & l_x k_T & -l_x k_T & -l_x k_T \\ k_\tau & -k_\tau & k_\tau & -k_\tau \end{bmatrix} \begin{bmatrix} \omega_{r,1}^2 \\ \omega_{r,2}^2 \\ \omega_{r,3}^2 \\ \omega_{r,4}^2 \end{bmatrix}. \tag{6.4}$$

6.2.3 Brushless DC Motor Model

In order to estimate the energy usage of the quadcopter the method as proposed in [150] is used, which neglects battery effects and the effects of commonly used Electronic Speed Controllers (ESCs). The method uses a model that represents the brushless DC motors used in quadcopters as a resistor and voltage generator in series and neglects the transient response of the electrical part of the motor, since it is of a faster timescale than that of the mechanical system. The model for a single motor is then given by

$$v_m = R_m i_m + k_e \omega_r, \tag{6.5a}$$

$$k_e i_m = J_r \dot{\omega}_r + k_\tau \omega_r^2 + d_\tau \omega_r + \tau_f, \tag{6.5b}$$

where R_m, J_r, k_e, τ_f, d_τ are motor constants and v_m and i_m are the voltage and current across the motor, respectively. The power usage of a single motor is given by

$$P = v_m i_m = R_m i_m^2 + k_e \omega_r i_m, \tag{6.6}$$

which, using (6.5b), can be written as a function of the angular velocity and its derivative as

$$P = c_1\dot{\omega}_r^2 + c_2\dot{\omega}_r\omega_r^2 + c_3\dot{\omega}_r\omega_r + c_4\dot{\omega}_r + c_5\omega_r^4 + c_6\omega_r^3 + c_7\omega_r^2 + c_8\omega_r + c_9, \tag{6.7}$$

where

$$\begin{aligned}
c_1 &= \frac{R_m}{k_e^2}J_r^2, \quad c_2 = 2\frac{R_m}{k_e^2}J_r k_\tau, \quad c_3 = 2\frac{R_m}{k_e^2}J_r d_\tau + J_r,\\
c_4 &= 2\frac{R_m}{k_e^2}J_r\tau_f, \quad c_5 = \frac{R_m}{k_e^2}k_\tau^2, \quad c_6 = 2\frac{R_m}{k_e^2}k_\tau d_\tau + k_\tau,\\
c_7 &= \frac{R_m}{k_e^2}(2k_\tau\tau_f + d_\tau^2) + d_\tau, \quad c_8 = 2\frac{R_m}{k_e^2}\tau_f d_\tau + \tau_f,\\
c_9 &= \frac{R_m}{k_e^2}\tau_f^2.
\end{aligned}$$

6.2.4 Problem statement

The objective in this chapter is to calculate time-optimal and energy-optimal trajectories that move from an initial state to a goal state (which is below the initial state), while avoiding the VRS. The corresponding optimal control problem formulations are posed first and the constraints for avoiding the VRS are discussed in the next section.

The energy usage of a quadcopter over a time window can be found by integrating the sum of the power usage of all propellers over that same window, i.e., $E = \int_0^{t_f} \sum_{i=1}^4 P_i \, \mathrm{d}t$. The general form of the problem tackled in this chapter is given by

Problem 4. *Given an initial* $(p_0, v_0, R_0, \omega_0)$ *and final* $(p_{t_f}, v_{t_f}, R_{t_f}, \omega_{t_f})$ *condition, find a solution to the following optimal control problem*

$$\begin{aligned}
\min_{\substack{\omega_{r,i}(t),\\ i\in\{1,2,3,4\}}} \quad & \mathcal{J}\\
\text{s.t.} \quad & \text{(6.1a)} - \text{(6.1d)} \ \textit{(dynamics)},\\
& \text{(6.4)} \ \textit{(thrust/torque-propeller relation)},\\
& (p(0), v(0), R(0), \omega(0)) = (p_0, v_0, R_0, \omega_0),\\
& (p(t_f), v(t_f), R(t_f), \omega(t_f)) = (p_{t_f}, v_{t_f}, R_{t_f}, \omega_{t_f}),\\
& 0 \le \omega_{r,i}(t) \le \omega_{r,max}, \ \forall t \in [0, t_f], \ i \in \{1,2,3,4\}\\
& v_{D,i} \in \mathcal{V}, \ i \in \{1,2,3,4\} \ \textit{(VRS constraint)},
\end{aligned} \tag{6.8}$$

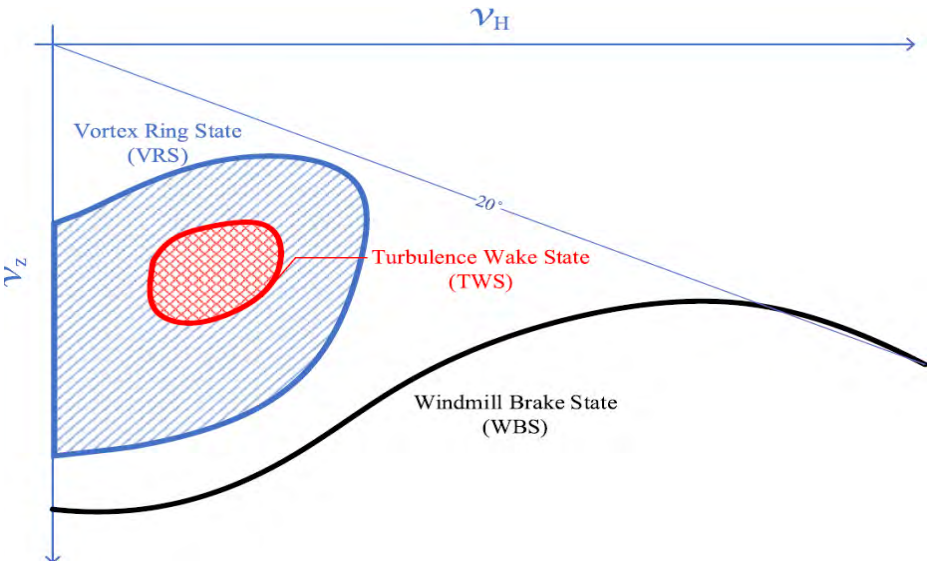


Figure 6.3: Sketch of different operating regimes of a propeller in the blade disk velocity space. The horizontal (ν_H) and vertical (ν_z) velocities are normalized by the induced velocity at hover [180]. Both the TWS and the WBS are to be avoided and are referred to, together with the VRS region, as simply the VRS in this chapter.

where $\omega_{r,max} > 0$ *is the maximum rotation speed of the propellers and* t_f *is the final time,* $v_{D,i}$ *and* $\mathcal{V}$ *are the blade disk velocities and VRS constraint, respectively, and will be introduced in the next section. The cost function is given by*

$$J = \begin{cases} \int_0^{t_f} \sum_{i=1}^{4} P_i \, \mathrm{d}t, & \textit{for energy optimization,} \\ t_f, & \textit{for time optimization.} \end{cases} \tag{6.9}$$

6.3 Vortex Ring State

The VRS is an effect that occurs when the quadcopter enters a specific region of the velocity state-space during descent maneuvers (see Fig. 6.1), resulting in high fluctuations in the thrust that the quadcopter can provide. The aerodynamics in the VRS are generally considered turbulent and difficult to model, therefore the aim is to avoid this region altogether and it is modeled as a hard constraint on the system.

As stated in the introduction, the different operating regimes have been extensively studied in helicopters. The regions can be characterized by the horizontal and vertical velocities in the blade disk frame, normalized by the induced

velocity at hover, as sketched in Fig. 6.3. The velocity of the origin of the blade disk frame belonging to the i^{th} propeller, i.e. D_i, expressed in the same blade disk frame, is given by

$$v_{D,i} = v_B + \omega \times l_i = R^\top v + \omega \times l_i, \tag{6.10}$$

where it is assumed that the blade disk frame and the body-fixed frame have the same orientation. The horizontal and vertical velocities in the blade disk frame, normalized by the induced velocity at hover are given by

$$\nu_{H,i} = \frac{\sqrt{v_{D,i,x}^2 + v_{D,i,y}^2}}{v_h}, \qquad \nu_{z,i} = \frac{v_{D,i,z}}{v_h}, \tag{6.11}$$

respectively. Here, $v_h = \sqrt{T_{\text{hover}}/2\rho A}$ is the induced velocity at hover, $v_{D,i} = [v_{D,i,x}, v_{D,i,y}, v_{D,i,z}]^\top$ and T_{hover}, ρ and A are the thrust of the propeller at hover, air density and blade disk area, respectively.

In [193] wind-tunnel experiments were performed with a quadcopter to identify the VRS regions for fixed pitch blade angles and brushless motors, as they are commonly used in quadcopter, contrary to helicopters that use variable pitch blades. As also stated in [193] the WBS is to be avoided in quadcopters, since due to the fixed pitch blades quadcopters are not able to go into auto-rotation as helicopters can. The results of the wind-tunnel experiments are shown in Fig. 6.4.

In [193] the region to be avoided was modeled as the region below the green line in Fig. 6.4, i.e., the blade disk velocity was constrained to only lie in the region defined as

$$v_{D,i} \in \mathcal{V}_1 := \{v_{D,i} \in \mathbb{R}^3 | \nu_{z,i} \leq \nu_{H,i} \tan(20°)\}, \tag{6.12}$$

which can be viewed as a cone into which the velocity vector cannot enter, see Fig. 6.6. Both 2D and 3D time-optimal trajectories were computed. However, the component of the blade disk frame that is due to the rotational velocity of the quadcopter was not taken into account, i.e., in [193] ω was (implicitly) considered to be zero in (6.10). To realize a more accurate description, here the rotational component is considered, which, especially for 2D trajectories, means that the constraint in (6.12) is not suitable. To see this, consider Fig. 6.5, where a sketch of the 2D model of the quadcopter as used in [193] is shown, together with the velocity constraint regions generated from (6.12). It becomes clear that in order to descend, the VRS region must be entered, since due to the rotational component at least one of the blade disks enters the constraint. For 3D trajectories the constraint in (6.12) entails that in order to descend, the quadcopter must, at least in the beginning, perform a yaw rotation. This is not the case in practice, where slow, purely vertical descent trajectories are possible for quadcopters.

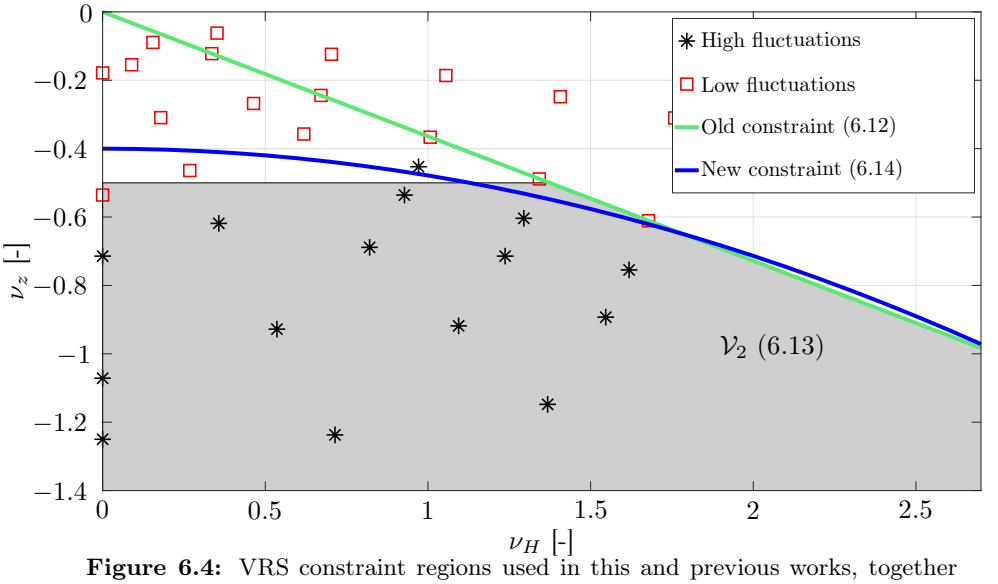


Figure 6.4: VRS constraint regions used in this and previous works, together with wind-tunnel measurements performed in [193]. The black asterisks and red squares represent measurements with high and low fluctuations of the propeller thrust, respectively. The area below the green line represents the constraint used in [193] (6.12), the gray area is the constraint used in [192] (6.13), and the area below the blue line is used in this chapter (6.14).

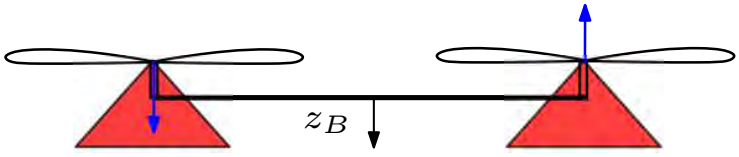


Figure 6.5: Sketch of a 2D quadcopter, with in red the VRS regions according to (6.12). It is clear that neither downward, nor rotational motion (for which the blue arrows depict the corresponding blade disk velocities) is possible without violating the VRS constraint.

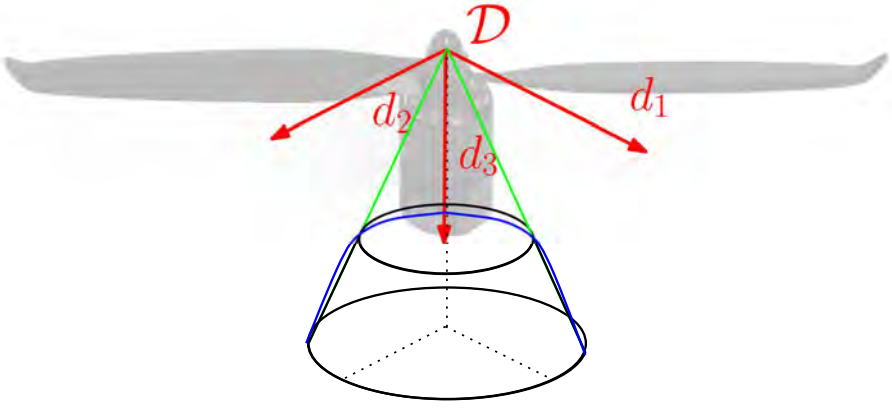


Figure 6.6: Blade disk $\{\mathcal{D}\}$ frame and VRS constraint regions. The velocity of the blade disk frame cannot enter the depicted green cone ($\mathcal{V}_1$) (6.12), black truncated cone ($\mathcal{V}_2$) (6.13) or blue paraboloid ($\mathcal{V}$) (6.14). In this chapter a smooth approximation of the truncated cone, represented by the blue paraboloid, is considered, in order to improve numerical stability.

In [192] a modified constraint was used, indicated by the gray colored polygon in Fig. 6.4. In 3D this yields the constraint set given by

$$v_{D,i} \in \mathcal{V}_2 := \{v_{D,i} \in \mathbb{R}^3 | \nu_{z,i} \leq \nu_{H,i} \tan(20^\circ) \cup \nu_{z,i} \leq 0.5\}, \tag{6.13}$$

which can be viewed as a truncated cone into which the velocity vector cannot enter, see Fig. 6.6. In this chapter a smooth approximation of this region is considered, in order to improve numerical stability. The approximated region is indicated by the area below the blue curve in Fig. 6.4 and is given by

$$v_{D,i} \in \mathcal{V} := \{v_{D,i} \in \mathbb{R}^3 | \nu_{z,i} \leq 0.28\nu_{H,i}^2 + 0.4\}. \tag{6.14}$$

This region can be viewed as a paraboloid, as shown in Fig. 6.6.

6.4 Computing time- and energy efficient trajectories

The optimal control problem in (6.8) is hard to solve due to several reasons. First, note that the dynamics in (6.1) are highly nonlinear. The constraint in (6.14) is non-convex in the blade disk velocity space, and since it is a nonlinear combination of the other states (see (6.10)), the constraint on the original states becomes even more complex. Moreover, the constraint needs to be considered for *each* propeller, further complicating matters.

Because of the complexity of the problems and the path constraints considered, a general purpose solver known as GPOPS-II [167] is employed. It is a *direct method*, i.e., where the original, continuous optimal control problem is discretized and transformed into a nonlinear programming (NLP) problem. More specifically, it employs a variable-order adaptive Legendre-Gauss-Radau quadrature collocation method.

In order to solve the problem using GPOPS-II, first the dynamics in (6.1) are written into quaternion form. The main advantage is that quaternions require only 4 parameters, whereas rotation matrices require 9. Let q denote the unit quaternion that represents the orientation of frame B with respect to frame W. The rotation of a vector $a \in \mathbb{R}^3$ by the quaternion q, resulting in a new vector $a' \in \mathbb{R}^3$, is then given by

$$a' = \mathcal{R}_q(a) = q \otimes \begin{bmatrix} 0 \\ a \end{bmatrix} \otimes q^*, \tag{6.15}$$

where $\otimes$ is the quaternion product and q^* the quaternion conjugate. The dynamics in (6.1) can then be written as

$$\dot{p} = v, \tag{6.16a}$$
$$\dot{v} = g z_W - \mathcal{R}_q(T e_3) - \mathcal{R}_q(D_r \mathcal{R}_{q^*}(v)), \tag{6.16b}$$
$$\dot{q} = \frac{1}{2} q \otimes \omega, \tag{6.16c}$$
$$J\dot{\omega} = \tau - S(\omega) J \omega - \tau_r. \tag{6.16d}$$

Next, the optimal control problem in (6.8) is written in a common form. Let the state and input vectors be defined by

$$x = [p, v, q, \omega, \omega_r]^\top \in \mathbb{R}^n, \quad u = \dot{\omega}_r \in \mathbb{R}^m \tag{6.17}$$

where $n = 17$ and $m = 4$. Consider the following optimal control problem

$$\begin{aligned} \min_{u(t)} \quad & \mathcal{J} \\ \text{s.t. } & \dot{x}(t) = f(x(t), u(t)), \\ & x(0) = x_0, \quad x(t_f) = x_{tf}, \\ & 0 \le \omega_{r,i}(t) \le \omega_{r,\max}, \ \forall t \in [0, t_f], \ i \in \{1,2,3,4\} \\ & v_{D,i} \in \mathcal{V}, \ i \in \{1,2,3,4\} \text{ (VRS constraint)}, \end{aligned} \tag{6.18}$$

Table 6.1: Parameters used in simulations [150].

$R_m = 0.2$ [Ohm]	$\rho = 1.225$ [kg/m^3]
$k_e = 0.0104$[Vs/rad]	$J = \text{diag}(0.081, 0.081, 0.142)$ [kgm^2]
$C_T = 0.0048$ [-]	$D_r = \text{diag}(0.3846, 0.3846, 0.1538)$ [s^{-1}]
$\tau_f = 4 \times 10^{-2}$ [Nm]	$J_m = 4.9 \times 10^{-6}$ [kgm^2]
$m = 1.3$ [kg]	$J_p = 3.7 \times 10^{-5}$ [kgm^2]
$r = 0.12$ [m]	$d_\tau = 2 \times 10^{-4}$ [Nm s/rad]
$l = 0.175$ [m]	$\omega_{r,\max} = 1047.2$ [rad/s]
$g = 9.81$ [m/s^2]	$\dot{\omega}_{r,\max} = 10000$ [rad/s^2]

Table 6.2: Initial and final conditions used for the simulations.

Variable	Unit	Initial Value	Final Value
p	[m]	$[0, 0, -30]^\top$	$[0, 0, 0]^\top$
v	[m/s]	$[0, 0, 0]^\top$	$[0, 0, 0]^\top$
q	[-]	$[1, 0, 0, 0]^\top$	$[1, 0, 0, 0]^\top$
ω	[rad/s]	$[0, 0, 0]^\top$	$[0, 0, 0]^\top$
ω_r	[rad/s]	$[912, 912, 912, 912]^\top$	$[912, 912, 912, 912]^\top$

where $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is given by (6.16) and the cost function is defined as in (6.9).

6.5 Numerical Case Study

The optimal control problem introduced in (6.18) is considered, with the parameters of a DJI Phantom 2 quadrotor [56], given in Table 6.1. Two scenarios are considered, both having the same initial and final conditions, where the quadcopter starts at a height of 30 meters and has to descend to 0 meters, see Table 6.2. The quadcopter is required to be at rest in both the initial and final point, in which case the propeller velocity is given by $\omega_{r,\text{hover}} = \sqrt{T_{\text{hover}}/k_T}$. In the first scenario the quadcopter is not constrained in its horizontal movement (apart from the VRS), whereas in the second scenario the horizontal positions of the quadcopter are restricted to lie in the interval $[-3, 3]$ meters. In both cases the quadcopter's roll and pitch angles are limited to ± 35 degrees, since many quadcopters used in industry are limited to low angle movements.

The resulting position trajectories are shown in Fig. 6.7 for the restricted and unrestricted horizontal positions, for both energy and time as optimization criteria. Moreover, the trajectories resulting from optimization without the VRS constraint are also shown, which are straight lines, as expected. Note that the time and energy trajectories are actually quite similar, they are almost mirror images of each other.

Table 6.3: Simulation results.

Restricted x, y	VRS	Criterion	Energy [kJ]	Time [s]
No	Yes	Energy	9.909	7.18
No	Yes	Time	10.62	7.02
No	Old	Energy	9.720	7.21
No	Old	Time	10.28	7.00
No	No	Energy	5.523	4.76
No	No	Time	5.636	4.74
Yes	Yes	Energy	12.34	8.87
Yes	Yes	Time	12.85	8.83
Yes	Old	Energy	11.91	8.68
Yes	Old	Time	12.71	8.57
Yes	No	Energy	5.523	4.76
Yes	No	Time	5.637	4.74

Numerical results from twelve different settings are compared in Table 6.3, where the 'Restricted' column refers to the restriction of horizontal positions. The column 'VRS' specifies whether the constraint (6.14) is used ('Yes') or not ('No') and 'Old' refers to the case where the constraint is applied but the angular velocity is not considered in (6.10). The 'Criterion' column specifies whether the trajectory is optimized for energy or time.

The results displayed in Fig. 6.7 and Table 6.3 show that the time and energy efficient trajectories are quite similar. This can be explained by considering that the final time is also an optimization variable in the computation of energy efficient trajectories, thus spending less time to reach the final configuration results in a lower energy use. Moreover, the trajectories that do not take the angular velocity in (6.10) into account ('Old'), are faster and more energy efficient, however they do violate the VRS in (6.14), as shown in Fig. 6.8 for the time-optimal, horizontally restricted case. The trajectories that ignore the VRS constraint are the fastest and most energy efficient, which is to be expected.

Fig. 6.9 shows the 3D position trajectory of the unrestricted, energy optimized case which respects the VRS constraint in (6.14), together with several snapshots of the quadcopter along the trajectory. From this figure more insight in the shape of the trajectory and the angles that the quadcopter goes through is obtained, from which it becomes clear that the quadcopter also uses the yaw angle to avoid the VRS region. In Fig. 6.10 this can also be seen by looking at the angles of the quadcopter over time, which also respect the constraint of ± 35 degrees for the roll and pitch angles as shown.

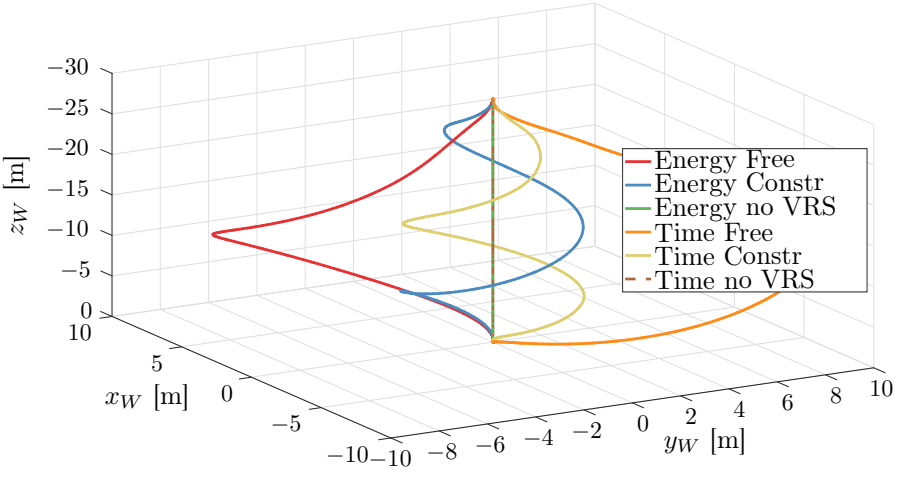


Figure 6.7: Three dimensional plot of the quadcopter trajectory. The results of energy optimization in the unrestricted translational, restricted translational and without the VRS cases are shown by the red, blue and green lines, respectively. Similarly, the orange, yellow and dashed brown lines show the unrestricted translational, restricted translational and without the VRS trajectories for time optimization. Note that the cases for energy and time optimal paths overlap in the case that the VRS is not considered.

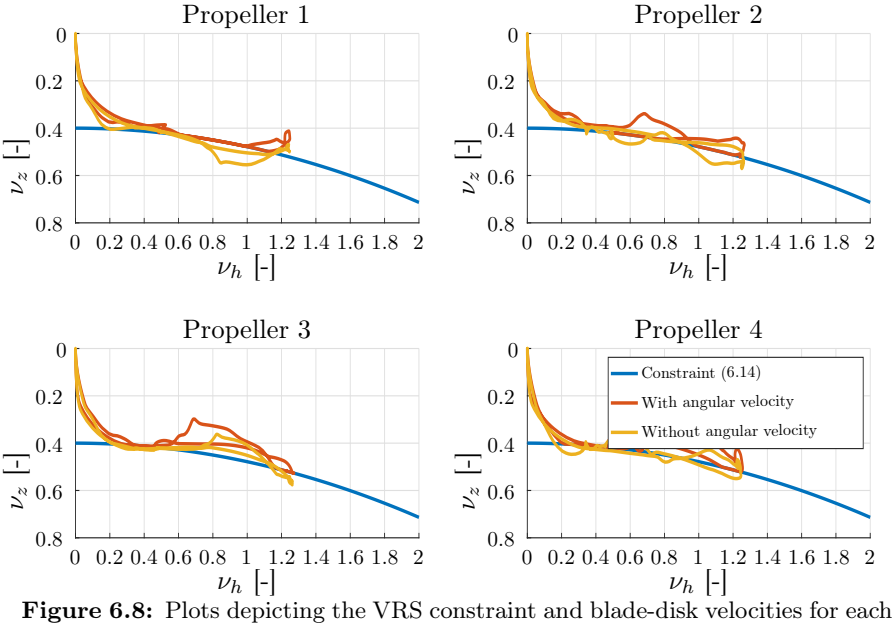


Figure 6.8: Plots depicting the VRS constraint and blade-disk velocities for each propeller for the time-optimal, horizontally restricted case.

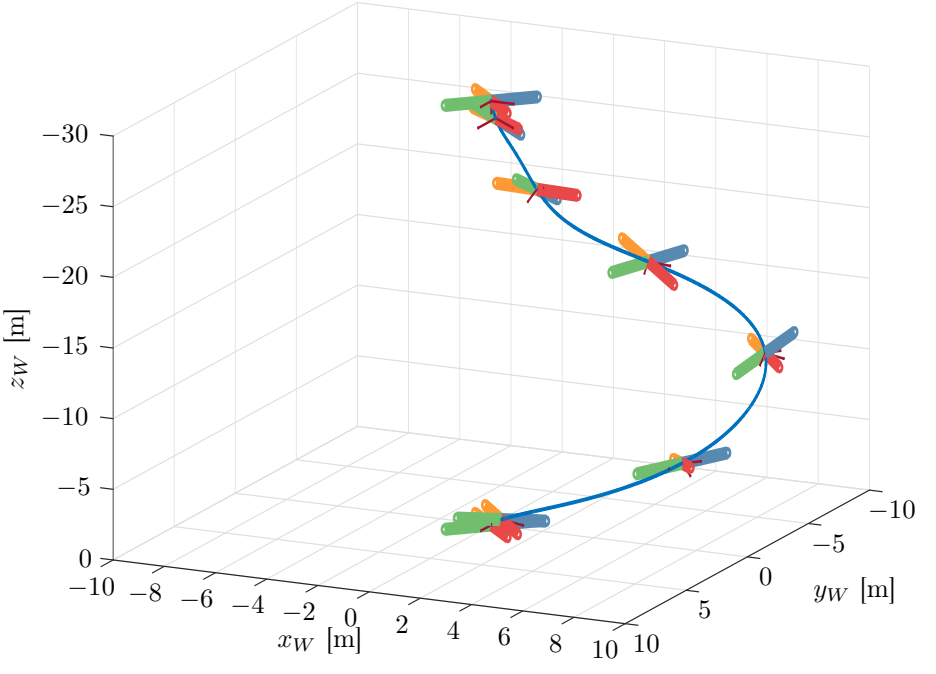


Figure 6.9: Three dimensional plot of the trajectory for the unrestricted, energy optimized case which respects the VRS constraint in (6.14). Several snapshots of the quadcopter along the trajectory are shown, yielding insight in the attitude of the quadcopter during the maneuver.

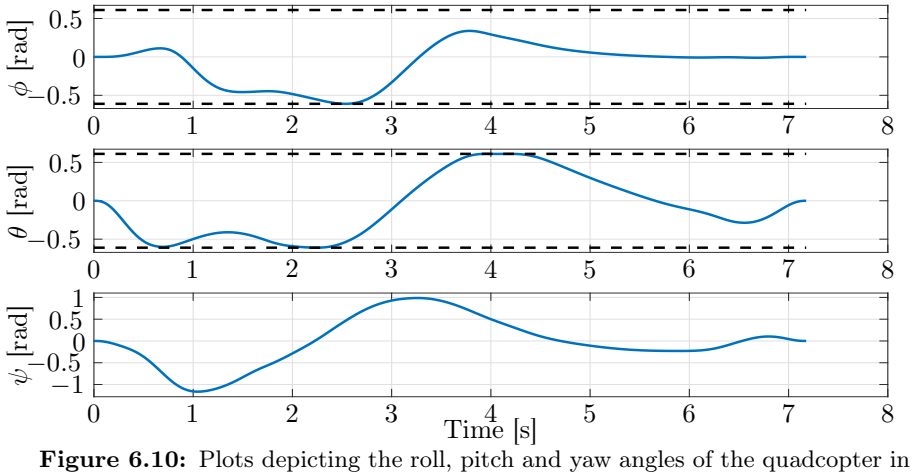


Figure 6.10: Plots depicting the roll, pitch and yaw angles of the quadcopter in radians, over time. The dashed, black lines show the constraint of ±35 degrees for the roll and pitch angles.

6.6 Conclusions and future directions

In this chapter the dynamics of the blade-disk velocity have been carefully modeled and taken into account to ensure the VRS region is avoided. Both time-optimal and energy-optimal trajectories that avoid the VRS have been computed using GPOPS II.

This chapter can be viewed as a starting point for many interesting research directions. For instance, the combination of nonlinear dynamics and non-convex constraint resulting from the VRS make for a difficult problem to solve. Further research can be performed into computing optimal trajectories using less computational power, allowing for online optimal control frameworks such as model predictive control to be applied. The VRS in quadcopters can itself be investigated further by performing more extensive experiments, as well as considering data-driven methods that allow for the (online) learning of the region and/or trajectories that avoid it.

Part IV

Closing

Chapter 7

Conclusions and Recommendations

In this dissertation, multiple steps have been taken to address important challenges on speed, efficiency and robustness of quadcopters. These steps are essential for quadcopters to become a fully integrated part of our society. This chapter provides the conclusions that have been reached in solving these challenges and provides suggestions for further research.

This thesis contains novel solutions to challenges that cover many aspects of quadcopters. The conclusions and recommendations related to each contribution presented in Section 1.6 are discussed separately.

7.1 Adaptive Complementary Filter

In the second chapter of this thesis the challenge of providing accurate attitude estimates using gyroscopic and vector-based measurements, where the vector-based measurements consist of accelerometer and magnetometer measurements, has been considered. It was observed that in real-world applications, the vector-based measurements suffer from disturbances that are often not taken into account in current attitude filter design, resulting in poor estimates. These disturbances are caused by, for instance, accelerations of the quadcopter that are not considered in the design of the filter or ferromagnetic materials in the environment that lead to disturbed measurements of the Earth's magnetic field by the magnetometer.

In order to provide more accurate estimates when these disturbances take place, a new attitude filter was proposed directly on the Special Orthogonal Group, in the form of an adaptive complementary filter, where the filter gain is adapted. This adaptation is based on the observation that when the attitude estimates of the gyroscope and vector-based measurements are of a similar shape in a certain time window, the vector-based measurements can be considered accurate. To quantify this, a similarity measure was introduced, which combined with the adaptation scheme, results in proven convergence of the filter estimate to the true attitude. It was shown that the filter is more robust to the considered disturbances, while not losing performance in non-disturbed cases.

Recommendations It is recommended to investigate the application of the adaptive component of the filter to other, more advanced complementary filter designs that have been proposed in the literature, although care must be taken to ensure that convergence can still be guaranteed. Another direction for future work is the usage of other sensors present on the quadcopter, such as cameras, which can further increase the accuracy of the obtained attitude estimate. Furthermore, knowledge of the quadcopter model, combined with sensors that provide position and/or velocity measurements, can also be used for improved attitude estimation. Finally, it could prove beneficial to use other adaptation rules based on, for instance, different insights or data-driven methods such as Gaussian Processes [169] or (Deep) Neural Networks [75].

7.2 Estimation and identification for Markov Jump Linear Systems

In chapters three and four, Markov Jump Linear Systems (MJLSs) have been studied in the context of both estimation and identification. MJLSs form a class

of switched linear systems, where the switching occurs according to a Markov chain, i.e., the mode of the system switches based on some probability, which is only related to the previous mode the system was in.

In Chapter 3 the problem of maximum likelihood estimation of both the state and the mode sequence over a certain horizon was considered. Estimates that satisfy this requirement are referred to as optimal joint maximum a posteriori probability (JMAP) estimates. It is known that computing the optimal JMAP estimate for MJLSs is computationally intractable, and, therefore, a novel, sub-optimal method was proposed. The proposed method uses a technique from the optimal control community known as relaxed dynamic programming, which allows to obtain estimates that are guaranteed to be within a pre-specified bound of the optimal estimates.

It was shown in Chapter 4 that the identification of several classes of Markov switched systems can be transcribed to the same estimation problem as was tackled in Chapter 3. In particular, this includes the following classes:

- switched autoregressive models with exogenous inputs (sARX), where the switching is governed by a Markov chain;
- jump Markov nonlinear systems with subsystem dynamics that depend affinely on a set of known state and control dependent features and the coefficients of the affine combination are unknown;
- MJLSs with full state information;

Hence, for these classes the same method can also be applied to identification, yielding the same guarantees on near-optimality.

The method provides the first theoretical guarantees on near-optimality in the literature, which is an important step forward compared to existing algorithms that are prone to getting stuck in local optima.

Recommendations The simulations performed on both the estimation and identification problems suggest that there is room to tighten the theoretical bound further. That is, the pre-specified bound currently needs to be set quite large to ensure that the problems remain computationally tractable, while the resulting estimates are still very close to optimal. It is recommended to improve upon the existing theoretical bound, in order to provide more insight and perhaps allow for different pruning methods.

Future work can be performed into finding a relation between a certain maximum number of hypotheses being retained and a corresponding bound on the cost. This would make the method more interesting for practical use, since a user can then weigh the cost of keeping more hypotheses (and therefore using more computation) with the benefit of being closer to the optimal estimates. An interesting research direction towards this relation between complexity and optimality is to compute, at each timestep, the smallest possible bound that

keeps the desired number of hypotheses. The guaranteed bound on the cost is then replaced with this smallest bound at the current timestep, if this smallest bound is larger than the previously computed guaranteed bound.

7.3 Model Predictive Control for Quadcopters

In Chapter 5 a novel model predictive control (MPC) strategy for quadcopter reference tracking was presented, that explicitly takes the limited thrust, which the quadcopter can provide, into account. The strategy consists of an inner-loop, concerning the attitude and angular velocity dynamics, and an outer-loop, which handles the position and translational velocity. By using this cascaded strategy, the outer-loop can be viewed as a linear system with a new, virtual input, which is transformed to setpoints for the inner-loop to ensure that this virtual input is actually achieved. For the inner-loop a nonlinear controller is used that provides almost global asymptotic stability. In the outer-loop, a system extension is first performed to ensure that, even when using a zero-order-hold sampling on the input, sufficiently smooth setpoints can be generated for the inner-loop. An input transformation is subsequently applied to transform the constraints on the state to constraints on the input only. For this system, a globally stabilizing controller is known, which is used in the formulation of an MPC scheme that provides global stability guarantees.

The MPC strategy provides the following main advantages:

1. (Almost) global asymptotic stability guarantees for the whole cascaded system, where it must be noted that these guarantees hold for the original, continuous-time system, even though a zero-order-hold controller is applied.

2. Guaranteed constraint satisfaction and incorporation of future reference information due to the usage of an MPC strategy.

3. Complete freedom in choosing the cost function used by the MPC scheme, while still providing stability guarantees. Moreover, the stability guarantees follow from *linear constraints on the first input only*. This means that fast computation times can be achieved by the MPC controller and real-time implementation is possible.

These three features make the proposed tracking controller a promising solution for future quadcopters.

Recommendations Although the presented strategy provides many advantages, there is room for improvement. For instance, the inner-loop controller uses feedforward signals to cancel many terms, which might not always be known or measurable. The assumption of full state information can also be investigated,

i.e., incorporating estimation (errors) in the design of the strategy. While the limited thrust capability of the quadcopter is taken into account, the torque that the quadcopter can generated around its axes is also limited in practice, which could be considered in a control strategy as well. This can perhaps be achieved by using MPC for the inner loop as well. Finally, the constraints considered in the MPC controller are very conservative in order to guarantee stability. It is recommended to investigate how these can be relaxed. If this is successful, it can be expected that higher performance can be achieved.

7.4 Fast Landing for Quadcopters while Avoiding the Vortex Ring State

In Chapter 6 the aerodynamic effect known as the Vortex Ring State has been studied for quadcopters. This effect occurs during (fast) descent trajectories, when the descent rate of the quadcopter approaches the induced velocity of the rotors, resulting in a loss of lift of some, or all, of the propellers. This, in turn, leads to a reduced control effectiveness and can lead to instability and uncontrollability of the quadcopter, with a crash as a common result.

Since this effect is detrimental to the safe operation of quadcopters and the dynamics during this effect are hard to model, the VRS is often avoided altogether by constraining the quadcopter to not enter the region, where the VRS might occur. A more complete modeling of the blade-disk velocities, which determine when the VRS region is entered, was performed, where the angular velocities of the quadcopter are taken into account. This is in contrast to previous work, where the effects of the angular velocities on the blade-disk velocities were not considered. The improved model was subsequently used in the planning of time-optimal and energy-optimal descent trajectories for quadcopters that avoid the VRS region.

Recommendations Due to the nonlinear dynamics of the quadcopter in combination with the non-convex constraint related to the VRS region, the computation of time-optimal and energy-optimal trajectories currently takes several hours. It is of interest to find approximations that allow for faster computation of time-optimal and energy-optimal trajectories, so that the trajectories could be computed online, resulting in a more practical application. This would also make for landing maneuvers that are more robust against disturbances and/or modeling errors due to online re-planning of trajectories. Possible avenues for finding useful approximations include the differential flatness property of quadcopters [66] as well as MPC, where in the latter case care must be taken to ensure recursive feasibility and optimality.

The VRS effect itself also invites further investigation, as the effect in quadcopters has only received little attention from the research community so far. In doing so, perhaps the detrimental effects caused by the VRS can be avoided

by not viewing the region as a hard constraint, but as a disturbance on the dynamics, which can be compensated for by some controller. Finally, data-driven methods could be of use as well in order to identify the VRS region or VRS dynamics, and it is recommended to pursue these directions in the future.

7.5 Final Thoughts

Quadcopters show great promise in providing more time-, cost- and energy-efficient solutions to existing and future problems in application fields such as agriculture, (industrial) inspection, law enforcement, delivery (in remote locations), construction, security, environmental studies and human transport. In order for quadcopters to become increasingly useful in many of these areas, it is important that some challenges are overcome. One of the disadvantages of quadcopters at this stage is their high energy usage and resulting limited flight time. If improvements can be made in this avenue, their usability would increase tremendously. Another interesting area is the guaranteed safety of quadcopters and their environment during normal use, as well as when damage occurs to the quadcopter.

In this thesis, new methods have been proposed that improve the capabilities of quadcopters. These methods cover many of the most important aspects of quadcopters, including estimation, identification, planning and control. By using these results as starting points for future work as suggested in the recommendations, quadcopters can become an important part of our everyday lives.

Bibliography

[1] H. V. Abeywickrama, B. A. Jayawickrama, Y. He, and E. Dutkiewicz, "Comprehensive energy consumption model for unmanned aerial vehicles, based on empirical studies of battery performance," *IEEE Access*, vol. 6, pp. 58 383–58 394, 2018.

[2] G. A. Ahlin, "The fluid dynamics of the helicopter vortex ring phenomenon," Ph.D. dissertation, 2007.

[3] H. Al-Jlailaty and M. M. Mansour, "Efficient attitude estimators: A tutorial and survey," *Journal of Signal Processing Systems*, 5 2021. [Online]. Available: https://doi.org/10.1007/s11265-020-01620-4

[4] A. R. P. Andriën and D. J. Antunes, "Near-optimal map estimation for markov jump linear systems using relaxed dynamic programming," *IEEE Control Systems Letters*, vol. 4, no. 4, pp. 815–820, 2020.

[5] A. R. P. Andriën, E. Lefeber, D. J. Antunes, and W. Heemels, "Model predictive controller for quadcopters with almost global trajectory tracking guarantees," *To be submitted*, 2021.

[6] A. Andriën and D. Antunes, "Filtering and smoothing in the presence of outliers using duality and relaxed dynamic programming," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 6038–6043.

[7] ——, "Near-optimal recursive identification for markov switched systems," in *Accepted to 2021 IEEE 60th Conference on Decision and Control (CDC)*, 2021.

[8] A. Andriën, D. Antunes, M. v. d. Molengraft, and W. Heemels, "Similarity-based adaptive complementary filter for imu fusion," in *2018 European Control Conference (ECC)*, 2018, pp. 3044–3049.

[9] A. Andriën, D. Kremers, D. Kooijman, and D. Antunes, "Model predictive tracking controller for quadcopters with setpoint convergence guarantees," in *2020 American Control Conference (ACC)*, 2020, pp. 3205–3210.

[10] A. Andriën, A. Talaeizadeh, D. Antunes, and W. Heemels, "Time and energy efficient descent trajectories for quadcopters that avoid the vortex ring state," in *Submitted to 2022 IEEE Conference on Robotics and Automation (ICRA)*, 2022.

[11] C. V. Angelino, V. R. Baraniello, and L. Cicala, "Uav position and attitude estimation using imu, gnss and camera," in *2012 15th International Conference on Information Fusion*, 2012, pp. 735–742.

[12] D. J. Antunes and H. Qu, "Frequency domain analysis of networked control systems modelled by markov jump linear systems," *IEEE Transactions on Control of Network Systems*, pp. 1–1, 2021.

[13] A. J. Baerveldt and R. Klang, "A low-cost and low-weight attitude estimation system for an autonomous helicopter," in *Proceedings of IEEE International Conference on Intelligent Engineering Systems*, 9 1997, pp. 391–395.

[14] M. Balasingam, "Drones in medicine—the rise of the machines," *International Journal of Clinical Practice*, vol. 71, no. 9, p. e12989, 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/ijcp.12989

[15] M. P. Balenzuela, A. G. Wills, C. Renton, and B. Ninness, "A variational expectation-maximisation algorithm for learning jump markov linear systems," 2020, [arXiv preprint arXiv:2004.08564, April 2020].

[16] ——, "Bayesian parameter identification for jump markov linear systems," 2021, [arXiv preprint arXiv:2004.08565, Februari 2021].

[17] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[18] M. Barao and J. S. Marques, "Offline bayesian identification of jump markov nonlinear systems," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 7761–7766, 2011, 18th IFAC World Congress. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S147466701644855X

[19] B. Barshan and H. F. Durrant-Whyte, "Inertial navigation systems for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 3, pp. 328–342, 6 1995.

[20] B. Barshan and H. Durrant-Whyte, "Evaluation of a solid-state gyroscope for robotics applications," *IEEE Transactions on Instrumentation and Measurement*, vol. 44, no. 1, pp. 61–67, 1995.

[21] H. Bavle, J. L. Sanchez-Lopez, P. De la Puente, A. Rodriguez-Ramos, C. Sampedro, and P. Campoy, "Fast and robust flight altitude estimation of multirotor uavs in dynamic unstructured environments using 3d point cloud sensors," *Aerospace*, vol. 5, no. 3, 2018. [Online]. Available: https://www.mdpi.com/2226-4310/5/3/94

[22] A. Bemporad, C. Pascucci, and C. Rocchi, "Hierarchical and hybrid model predictive control of quadcopter air vehicles," *IFAC Proceedings Volumes*, vol. 42, no. 17, pp. 14–19, 2009, 3rd IFAC Conference on Analysis and Design of Hybrid Systems. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667015307308

[23] M. Beul and S. Behnke, "Analytical time-optimal trajectory generation and control for multirotors," in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2016, pp. 87–96.

[24] S. P. Bhat and D. S. Bernstein, "A topological obstruction to continuous global stabilization of rotational motion and the unwinding phenomenon," *Systems & Control Letters*, vol. 39, no. 1, pp. 63–70, 2000. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167691199000900

[25] M. Billio, A. Monfort, and C. Robert, "Bayesian estimation of switching arma models," *Journal of Econometrics*, vol. 93, no. 2, pp. 229–255, 1999.

[26] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon path planning for 3d exploration and surface inspection," *Autonomous Robots*, vol. 42, no. 2, pp. 291–306, 11 2016. [Online]. Available: https://doi.org/10.1007/s10514-016-9610-0

[27] L. Blackmore, S. Gil, Seung Chung, and B. Williams, "Model learning for switching linear systems with autonomous mode transitions," in *2007 46th IEEE Conference on Decision and Control*, 2007, pp. 4648–4655.

[28] W. P. Blair and D. D. Sworder, "Continuous-time regulation of a class of econometric models," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-5, no. 3, pp. 341–346, 5 1975.

[29] S. Bouabdallah and R. Siegwart, "Backstepping and sliding-mode techniques applied to an indoor micro quadrotor," in *2005 IEEE Int. Conference on Robotics and Automation*, 4 2005, pp. 2247–2252.

[30] S. Bouabdallah, A. Noth, and R. Siegwart, "Pid vs lq control techniques applied to an indoor micro quadrotor," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, 2004, pp. 2451–2456 vol.3.

[31] Y. Bouktir, M. Haddad, and T. Chettibi, "Trajectory planning for a quadrotor helicopter," in *2008 16th Mediterranean Conference on Control and Automation*, 2008, pp. 1258–1263.

[32] P.-J. Bristeau, F. Callou, D. Vissière, and N. Petit, "The navigation and control technology inside the ar.drone micro uav," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 1477–1484, 2011, 18th IFAC World Congress. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667016438188

[33] F. Bullo and A. D. Lewis, *Geometric control of mechanical systems : modeling, analysis, and design for simple mechanical control systems.* New York, NY: Springer, 2005.

[34] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[35] G. Cai, J. Dias, and L. Seneviratne, "A survey of small-scale unmanned aerial vehicles: Recent advances and future development trends," *Unmanned Systems*, vol. 02, no. 02, pp. 175–199, 2014.

[36] D. Chen, L. Bako, and S. Lecœuche, "A recursive sparse learning method: Application to jump markov linear systems," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 3198–3203, 2011, 18th IFAC World Congress. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667016441030

[37] X. Chen, S. Zhao, and F. Liu, "Identification of time-delay markov jump autoregressive exogenous systems with expectation-maximization algorithm," *International Journal of Adaptive Control and Signal Processing*, vol. 31, no. 12, pp. 1920–1933, 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/acs.2807

[38] Y. Chen, D. Baek, A. Bocca, A. Macii, E. Macii, and M. Poncino, "A case for a battery-aware model of drone energy consumption," in *2018 IEEE International Telecommunications Energy Conference (INTELEC)*, 2018, pp. 1–8.

[39] L. Chenglong, F. Zhou, W. Jiafang, and Z. Xiang, "A vortex-ring-state-avoiding descending control strategy for multi-rotor uavs," in *2015 34th Chinese Control Conference (CCC)*, 2015, pp. 4465–4471.

[40] A. Cherian, J. Andersh, V. Morellas, N. Papanikolopoulos, and B. Mettler, "Autonomous altitude estimation of a uav using a single onboard camera,"

in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 3900–3905.

[41] E. Cinquemani, R. Porreca, G. Ferrari-Trecate, and J. Lygeros, "A general framework for the identification of jump markov linear systems," in *2007 46th IEEE Conference on Decision and Control*, 2007, pp. 5737–5742.

[42] P. Cohn, A. Green, M. Langstaff, and M. Roller, "Commercial drones are here: The future of unmanned aerial systems," *McKinsey & Company*, 2017.

[43] P. Corke, *Robotics, Vision and Control*. Springer International Publishing, 2017. [Online]. Available: https://doi.org/10.1007/978-3-319-54413-7

[44] P. Corke, J. Lobo, and J. Dias, "An introduction to inertial and visual sensing," *The International Journal of Robotics*, vol. 26, pp. 519–535, 2007.

[45] O. Costa, J. do Val, and J. Geromel, "Continuous-time state-feedback h_2-control of markovian jump linear systems via convex analysis," *Automatica*, vol. 35, no. 2, pp. 259 – 268, 1999.

[46] O. L. V. Costa, M. D. Fragoso, and R. P. Marques, *Discrete-time Markov jump linear systems*. Springer Science, 2006.

[47] I. D. Cowling, O. A. Yakimenko, J. F. Whidborne, and A. K. Cooke, "A prototype of an autonomous controller for a quadrotor uav," in *2007 European Control Conference (ECC)*, 2007, pp. 4001–4008.

[48] B. Coxworth. (2013, 4) Dji announces stabilized hero mount, and camera-equipped phantom quadcopter. New Atlas. [Online]. Available: https://newatlas.com/dji-phantom-vision-zenmuse-quadcopter/27028/

[49] T. Crouch, S. N. Air, and S. Museum, *Wings: A History of Aviation from Kites to the Space Age*. Smithsonian National Air and Space Museum, 2003.

[50] R. Cunha, D. Antunes, P. Gomes, and C. Silvestre, "A path-following preview controller for autonomous air vehicles," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*. American Institute of Aeronautics and Astronautics, 6 2006. [Online]. Available: https://doi.org/10.2514/6.2006-6715

[51] A. L. da Silva and J. J. da Cruz, "Fuzzy adaptive extended kalman filter for uav ins/gps data fusion," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 38, no. 6, pp. 1671–1688, 8 2016.

[52] B. Dang-Vu, "Vortex ring state protection flight control law." in *39th European Rotorcraft Forum*, MOSCOU, Russia, Sep. 2013. [Online]. Available: https://hal-onera.archives-ouvertes.fr/hal-01061133

[53] H. G. de Marina, F. J. Pereda, J. M. Giron-Sierra, and F. Espinosa, "Uav attitude estimation using unscented kalman filter and triad," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4465–4474, 11 2012.

[54] B. de Miguel Molina and M. Segarra Oña, *The Drone Sector in Europe*. Cham: Springer International Publishing, 2018, pp. 7–33. [Online]. Available: https://doi.org/10.1007/978-3-319-71087-7_2

[55] C. E. de Souza and M. D. Fragoso, "h_∞ filtering for discrete-time linear systems with markovian jumping parameters," *Int. Journal of Robust and Nonlinear Control*, vol. 13, no. 14, pp. 1299–1316, 2003.

[56] DJI. Dji phantom 2. DJI. [Online]. Available: https://www.dji.com/nl/phantom-2

[57] A. Doucet and C. Andrieu, "Iterative algorithms for state estimation of jump markov linear systems," *IEEE Transactions on Signal Processing*, vol. 49, no. 6, pp. 1216–1227, 6 2001.

[58] A. Doucet, A. Logothetis, and V. Krishnamurthy, "Stochastic sampling algorithms for state estimation of jump markov linear systems," *IEEE Trans. on Automatic Control*, vol. 45, no. 2, pp. 188–202, 2 2000.

[59] T. M. Drews, P. G. Kry, J. R. Forbes, and C. Verbrugge, "Sequential pose estimation using linearized rotation matrices," in *2013 International Conference on Computer and Robot Vision*, 2013, pp. 113–120.

[60] S. Driessen, N. Janssen, L. Wang, J. Palmer, and H. Nijmeijer, "Experimentally validated extended kalman filter for uav state estimation using low-cost sensors," *IFAC-PapersOnLine*, vol. 51, no. 15, pp. 43–48, 2018, 18th IFAC Symposium on System Identification SYSID 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896318317488

[61] H. Durrant-Whyte, N. Roy, and P. Abbeel, *Construction of Cubic Structures with Quadrotor Teams*. MITP, 2012, pp. 177–184.

[62] A. El Hadri and A. Benallegue, "Sliding mode observer to estimate both the attitude and the gyro-bias by using low-cost sensors," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 2867–2872.

[63] B. J. Emran and H. Najjaran, "A review of quadrotor: An underactuated mechanical system," *Annual Reviews in Control*, vol. 46, pp. 165–180, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1367578818300932

[64] D. Eynard, P. Vasseur, C. Demonceaux, and V. Frémont, "Uav altitude estimation by mixed stereoscopic vision," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 646–651.

[65] D. Eynard, P. Vasseur, C. Demonceaux, and V. Frémont, "Real time UAV altitude, attitude and motion estimation from hybrid stereovision," *Autonomous Robots*, vol. 33, no. 1-2, pp. 157–172, 3 2012. [Online]. Available: https://doi.org/10.1007/s10514-012-9285-0

[66] M. Faessler, A. Franchi, and D. Scaramuzza, "Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 620–626, 2018.

[67] I. Fantoni, R. Lozano, and F. Kendoul, "Asymptotic stability of hierarchical inner-outer loop-based flight controllers," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 1741 – 1746, 2008, 17th IFAC World Congress.

[68] J. V. Foster and D. Hartman, "High-fidelity multi-rotor unmanned aircraft system (UAS) simulation development for trajectory prediction under off-nominal flight dynamics," in *17th AIAA Aviation Technology, Integration, and Operations Conference*. American Institute of Aeronautics and Astronautics, Jun. 2017. [Online]. Available: https://doi.org/10.2514/6.2017-3271

[69] E. Foxlin, "Inertial head-tracker sensor fusion by a complementary separate-bias kalman filter," in *Proceedings of the IEEE 1996 Virtual Reality Annual International Symposium*, 3 1996, pp. 185–194, 267.

[70] S. Garrido-Jurado, R. M. noz Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280 – 2292, 2014.

[71] P. Gąsior, S. Gardecki, J. Gośliński, and W. Giernacki, "Estimation of altitude and vertical velocity for multirotor aerial vehicle using kalman filter," in *Recent Advances in Automation, Robotics and Measuring Techniques*. Springer International Publishing, 2014, pp. 377–385. [Online]. Available: https://doi.org/10.1007/978-3-319-05353-0_36

[72] M. Geisert and N. Mansard, "Trajectory generation for quadrotor based systems using numerical optimal control," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 2958–2964.

[73] W. Giernacki, M. Skwierczyński, W. Witwicki, P. Wroński, and P. Kozierski, "Crazyflie 2.0 quadrotor as a platform for research and education in robotics and control engineering," in *2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*, 2017, pp. 37–42.

[74] S. Gil and B. Williams, "Beyond local optimality: An improved approach to hybrid model learning," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, 2009, pp. 3938–3945.

[75] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[76] G. C. Goodwin, M. M. Seron, and J. A. De Doná, *Constrained Estimation*. London: Springer London, 2005, pp. 187–216.

[77] D. Görges, M. Izak, and S. Liu, "Optimal control and scheduling of switched systems," *IEEE Transactions on Automatic Control*, vol. 56, no. 1, pp. 135–140, 1 2011.

[78] R. Graves, *The Greek Myths: The Complete and Definitive Edition*. Penguin Books, Limited, 2017.

[79] M. Greeff and A. P. Schoellig, "Flatness-based model predictive control for quadrotor trajectory tracking," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 10 2018, pp. 6740–6745.

[80] H. F. Grip, T. I. Fossen, T. A. Johansen, and A. Saberi, "Globally exponentially stable attitude and gyro bias estimation with application to gnss/ins integration," *Automatica*, vol. 51, pp. 158–166, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109814004579

[81] A. Gynnild, "The robot eye witness," *Digital Journalism*, vol. 2, no. 3, pp. 334–343, 2014. [Online]. Available: https://doi.org/10.1080/21670811.2014.883184

[82] T. Hamel and R. Mahony, "Attitude estimation on so[3] based on direct inertial measurements," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 5 2006, pp. 2170–2175.

[83] J. D. Hamilton, "A new approach to the economic analysis of nonstationary time series and the business cycle," *Econometrica*, vol. 57, no. 2, pp. 357–384, 1989.

[84] G. Hartmann. Clément-bayard, sans peur et sans reproche. [Accessed 15-June-2021]. [Online]. Available: http://www.hydroretro.net/etudegh/clement-bayard.pdf

[85] M. Hassanalian, D. Rice, and A. Abdelkefi, "Evolution of space drones for planetary exploration: A review," *Progress in Aerospace Sciences*, vol. 97, pp. 61–105, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0376042117301884

[86] J. Hauser, "A projection operator approach to the optimization of trajectory functionals," *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 377–382, 2002, 15th IFAC World Congress. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667015387334

[87] M. Hehn and R. D'Andrea, "Quadrocopter trajectory generation and control," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 1485–1491, 2011, 18th IFAC World Congress. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S147466701643819X

[88] M. Hehn and R. D'Andrea, "A frequency domain iterative learning algorithm for high-performance, periodic quadrocopter maneuvers," *Mechatronics*, vol. 24, no. 8, pp. 954–965, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957415814001561

[89] M. Hehn, R. Ritz, and R. D'Andrea, "Performance benchmarking of quadrotor systems using time-optimal control," *Autonomous Robots*, vol. 33, no. 1-2, pp. 69–88, Mar. 2012. [Online]. Available: https://doi.org/10.1007/s10514-012-9282-3

[90] A. V. Heikki Hyyti, "A dcm based attitude estimation algorithm for low-cost mems imus," *International Journal of Navigation and Observation*, vol. 2015, 2015.

[91] T. A. Heppenheimer, *First flight : the Wright brothers and the invention of the airplane*. Hoboken, N.J: Wiley, 2003.

[92] S. Hojjatinia, C. M. Lagoa, and F. Dabbene, "A method for identification of markovian jump arx processes," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 14 088–14 093, 2017, 20th IFAC World Congress.

[93] ——, "Identification of switched autoregressive exogenous systems from large noisy datasets," *International Journal of Robust and Nonlinear Control*, vol. 30, no. 15, pp. 5777–5801, 2020.

[94] P. Hou, A. Saberi, Z. Lin, and P. Sannuti, "Simultaneous external and internal stabilization for continuous and discrete-time critically unstable linear systems with saturating actuators," *Automatica*, vol. 34, no. 12, pp. 1547 – 1557, 1998.

[95] B. Hu and S. Mishra, "Time-optimal trajectory generation for landing a quadrotor onto a moving platform," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 2, pp. 585–596, 2019.

[96] X. Hu, D. Olesen, and P. Knudsen, "Trajectory generation using semidefinite programming for multi-rotors," in *2019 18th European Control Conference (ECC)*, 2019, pp. 2577–2582.

[97] J. Hwangbo, I. Sa, R. Siegwart, and M. Hutter, "Control of a quadrotor with reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2096–2103, 2017.

[98] K. J. Jensen, "Generalized nonlinear complementary attitude filter," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 5, pp. 1588–1593, 2011. [Online]. Available: https://doi.org/10.2514/1.53467

[99] X. Jin and B. Huang, "Identification of switched markov autoregressive exogenous systems with hidden switching state," *Automatica*, vol. 48, no. 2, pp. 436–441, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109811005450

[100] W. Johnson, *Rotorcraft aeromechanics*. Cambridge New York, NY: Cambridge University Press, 2013.

[101] A. L. Juloski, S. Weiland, and W. P. M. H. Heemels, "A bayesian approach to identification of hybrid systems," *IEEE Transactions on Automatic Control*, vol. 50, no. 10, pp. 1520–1533, 2005.

[102] J.-M. Kai, G. Allibert, M.-D. Hua, and T. Hamel, "Nonlinear feedback control of quadrotors exploiting first-order drag effects," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 8189–8195, 2017, 20th IFAC World Congress. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896317317822

[103] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *Journal of basic engineering*, vol. 83, no. 1, pp. 95–108, 1961.

[104] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[105] V. Kangunde, R. S. Jamisola, and E. K. Theophilus, "A review on drones controlled in real-time," *International Journal of Dynamics and Control*, 1 2021. [Online]. Available: https://doi.org/10.1007/s40435-020-00737-5

[106] K. J. Keesman, *System identification : an introduction*. London New York: Springer, 2011.

[107] H. Khalil, *Nonlinear Systems*, ser. Pearson Education. Prentice Hall, 2002.

[108] M. Khan, M. Zafar, and A. Chatterjee, "Barrier functions in cascaded controller: Safe quadrotor control," in *2020 American Control Conference (ACC)*, 2020, pp. 1737–1742.

[109] J. Kim, S. A. Gadsden, and S. A. Wilkerson, "A comprehensive survey of control strategies for autonomous quadrotors," *Canadian Journal of Electrical and Computer Engineering*, vol. 43, no. 1, pp. 3–16, 2020.

[110] E. Kirschner, *Aerospace balloons : from Montgolfiere to space*. Fallbrook, CA: Aero, 1985.

[111] M. Kok, *Probabilistic modeling for sensor fusion with inertial measurements*. Linköping: Linköping University Electronic Press, 2016.

[112] M. Kok and T. B. Schön, "Magnetometer calibration using inertial sensors," *IEEE Sensors Journal*, vol. 16, no. 14, pp. 5679–5689, 2016.

[113] M. Kok, J. D. Hol, and T. B. Schön, "Indoor positioning using ultrawideband and inertial measurements," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1293–1303, 2015.

[114] D. Kooijman, A. P. Schoellig, and D. J. Antunes, "Trajectory tracking for quadrotors with attitude control on $\mathcal{S}^2 \times \mathcal{S}^1$," in *2019 18th European Control Conference (ECC)*, 6 2019, pp. 4002–4009.

[115] R. Kottath, P. Narkhede, V. Kumar, V. Karar, and S. Poddar, "Multiple model adaptive complementary filter for attitude estimation," *Aerospace Science and Technology*, vol. 69, no. Supplement C, pp. 574 – 581, 2017.

[116] V. Krishnamurthy, S. Dey, and J. P. LeBlanc, "Blind equalization of iir channels using hidden markov models and extended least squares," *IEEE Transactions on Signal Processing*, vol. 43, no. 12, pp. 2994–3006, 12 1995.

[117] B. Kuchera. (2010, 10) The $300 quadrocopter: your iphone is a remote control. ARS Technica. [Online]. Available: https://arstechnica.com/gaming/2010/10/the-300-quadrocopter-your-iphone-is-a-remote-control/

[118] F. Lauer and G. Bloch, *Hybrid System Identification: Theory and Algorithms for Learning Switching Models*. Lect. Notes Control, 2019.

[119] A. Ledergerber and R. D'andrea, "Calibrating away inaccuracies in ultra wideband range measurements: A maximum likelihood approach," *IEEE Access*, vol. 6, pp. 78 719–78 730, 2018.

[120] J. H. Ledet, V. P. Jilkov, and X. R. Li, "Recursive sliding-window algorithm for constrained multiple-model map estimation," in *2018 21st Int. Conference on Information Fusion*, 7 2018, pp. 1–5.

[121] T. Lee, "Bayesian attitude estimation with the matrix fisher distribution on so(3)," *IEEE Transactions on Automatic Control*, vol. 63, no. 10, pp. 3377–3392, 2018.

[122] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor uav on se(3)," in *49th IEEE Conference on Decision and Control (CDC)*, 2010, pp. 5420–5425.

[123] E. Lefeber, S. J. A. M. van den Eijnden, and H. Nijmeijer, "Almost global tracking control of a quadrotor uav on se(3)," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 12 2017, pp. 1175–1180.

[124] G. Lente and K. Ősz, "Barometric formulas: various derivations and comparisons to environmentally relevant observations," *ChemTexts*, vol. 6, no. 2, Apr. 2020. [Online]. Available: https://doi.org/10.1007/s40828-020-0111-6

[125] W. Li and J. Wang, "Effective adaptive kalman filter for mems-imu/magnetometers integrated attitude and heading reference systems," *Journal of Navigation*, vol. 66, no. 1, p. 99–113, 2013.

[126] D. Liberzon, *Switching in Systems and Control*. Boston, MA, USA: Birkhauser, 2003.

[127] C. F. Liew, D. DeLatte, N. Takeishi, and T. Yairi, "Recent developments in aerial robotics: A survey and prototypes overview," 2017, [arXiv:1711.10085, November 2017].

[128] J. Linchant, J. Lisein, J. Semeki, P. Lejeune, and C. Vermeulen, "Are unmanned aircraft systems (uass) the future of wildlife monitoring? a review of accomplishments and challenges," *Mammal Review*, vol. 45, no. 4, pp. 239–252, 2015. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/mam.12046

[129] B. Lincoln and A. Rantzer, "Relaxing dynamic programming," *IEEE Transactions on Automatic Control*, vol. 51, no. 8, pp. 1249–1260, 8 2006.

[130] A. Logothetis and V. Krishnamurthy, "Expectation maximization algorithms for map estimation of jump markov linear systems," *IEEE Trans. on Signal Processing*, vol. 47, no. 8, pp. 2139–2156, 8 1999.

[131] R. Lozano, A. Sanchez, S. Salazar-Cruz, I. Fantoni, and J. Torres, "Discrete-time stabilization of integrators in cascade: Real-time stabilization of a mini-rotorcraft," in *Proceedings of the 45th IEEE Conference on Decision and Control*, 12 2006, pp. 6265–6270.

[132] H. Lu, K. Chen, X. B. Zhai, B. Chen, and Y. Zhao, "Tradeoff between duration and energy optimization for speed control of quadrotor unmanned aerial vehicle," in *2018 IEEE Symposium on Product Compliance Engineering - Asia (ISPCE-CN)*, 2018, pp. 1–5.

[133] H. J. Luinge, P. H. Veltink, and C. T. M. Baten, "Estimation of orientation with gyroscopes and accelerometers," in *Proceedings of the First Joint BMES/EMBS Conference. 1999 IEEE Engineering in Medicine and Biology 21st Annual Conference and the 1999 Annual Fall Meeting of the Biomedical Engineering Society (Cat. N*, vol. 2, 10 1999, pp. 844 vol.2–.

[134] A. Maas. Hadek chimney inspection. Avular. [Online]. Available: https://avular.com/casestudies/hadek/

[135] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," in *2011 IEEE International Conference on Rehabilitation Robotics*, 6 2011, pp. 1–7.

[136] R. Mahony, T. Hamel, and J. M. Pflimlin, "Complementary filter design on the special orthogonal group so(3)," in *Proceedings of the 44th IEEE Conference on Decision and Control*, 12 2005, pp. 1477–1484.

[137] ——, "Nonlinear complementary filters on the special orthogonal group," *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1203–1218, 6 2008.

[138] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE Robotics Automation Magazine*, vol. 19, no. 3, pp. 20–32, 9 2012.

[139] P. Marantos, Y. Koveos, and K. J. Kyriakopoulos, "Uav state estimation using adaptive complementary filters," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1214–1226, 7 2016.

[140] J. L. Marins, X. Yun, E. R. Bachmann, R. B. McGhee, and M. J. Zyda, "An extended kalman filter for quaternion-based orientation estimation using marg sensors," in *Proceedings 2001 IEEE/RSJ International Conference*

on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180), vol. 4, 2001, pp. 2003–2011 vol.4.

[141] K. Masuya, T. Sugihara, and M. Yamamoto, "Design of complementary filter for high-fidelity attitude estimation based on sensor dynamics compensation with decoupled properties," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 606–611.

[142] C. G. Mayhew, R. G. Sanfelice, and A. R. Teel, "Quaternion-based hybrid control for robust global attitude tracking," *IEEE Transactions on Automatic Control*, vol. 56, no. 11, pp. 2555–2566, 11 2011. [Online]. Available: https://doi.org/10.1109/tac.2011.2108490

[143] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, Dec. 2014. [Online]. Available: https://doi.org/10.1016/j.automatica.2014.10.128

[144] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789 – 814, 2000.

[145] G. McLachlan and D. Peel, *Finite Mixture Models*. Wiley, 2000.

[146] J. McQuaid, A. Kolaei, G. Bramesfeld, and P. Walsh, "Early onset prediction for rotors in vortex ring state," *Journal of Aerospace Engineering*, vol. 33, no. 6, p. 04020081, 2020.

[147] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE International Conference on Robotics and Automation*, 5 2011, pp. 2520–2525.

[148] X. Meng and T. Chen, "Optimal sampling and performance comparison of periodic and event based impulse control," *IEEE Transactions on Automatic Control*, vol. 57, no. 12, pp. 3252 –3259, 12 2012.

[149] P. Monzón, "On necessary conditions for almost global stability," *IEEE Transactions on Automatic Control*, vol. 48, no. 4, pp. 631–634, 2003.

[150] F. Morbidi, R. Cano, and D. Lara, "Minimum-energy path generation for a quadrotor uav," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1492–1498.

[151] H. D. K. Motlagh, F. Lotfi, H. D. Taghirad, and S. B. Germi, "Position estimation for drones based on visual slam and imu in gps-denied environment," in *2019 7th International Conference on Robotics and Mechatronics (ICRoM)*, 2019, pp. 120–124.

[152] M. W. Mueller and R. D'Andrea, "A model predictive controller for quadrocopter state interception," in *2013 European Control Conference (ECC)*, 2013, pp. 1383–1389.

[153] M. W. Mueller, M. Hamer, and R. D'Andrea, "Fusing ultra-wideband range measurements with accelerometers and rate gyroscopes for quadrocopter state estimation," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 1730–1736.

[154] M. W. Mueller, M. Hehn, and R. D'Andrea, "A computationally efficient motion primitive for quadrocopter trajectory generation," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1294–1310, 2015.

[155] K. Munson, *Helicopters and other rotorcraft since 1907*. London: Blandford Publishing, 1968.

[156] J. Musić, P. M. Cecić, and P. V. Zanchi, "Real-time body orientation estimation based on two-layer stochastic filter architecture," *Automatika*, vol. 51, no. 3, pp. 264–274, 2010. [Online]. Available: https://doi.org/10.1080/00051144.2010.11828380

[157] M. Nazarahari and H. Rouhani, "40 years of sensor fusion for orientation tracking via magnetic and inertial measurement units: Methods, lessons learned, and future challenges," *Information Fusion*, vol. 68, pp. 67–84, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253520303997

[158] A. A. Neto, D. G. Macharet, V. C. da Silva Campos, and M. F. Montenegro Campos, "Adaptive complementary filtering algorithm for mobile robot localization," *Journal of the Brazilian Computer Society*, vol. 15, no. 3, pp. 19–31, 9 2009.

[159] P. Neto, J. N. Pires, and A. P. Moreira, "3-d position estimation from inertial sensing: Minimizing the error from the process of double integration of accelerations," in *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*, 2013, pp. 4026–4031.

[160] B. J. Njinwoua and A. V. Wouwer, "Cascade attitude control of a quadcopter in presence of motor asymmetry," *IFAC-PapersOnLine*, vol. 51, no. 4, pp. 113–118, 2018, 3rd IFAC Conference on Advances in Proportional-Integral-Derivative Control PID 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896318303513

[161] k. Odry, I. Kecskes, P. Sarcevic, Z. Vizvari, A. Toth, and P. Odry, "A novel fuzzy-adaptive extended kalman filter for real-time attitude estimation of mobile robots," *Sensors*, vol. 20, no. 3, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/3/803

[162] S. M. Oh, "Multisensor fusion for autonomous uav navigation based on the unscented kalman filter with sequential measurement updates," in *2010 IEEE Conference on Multisensor Fusion and Integration*, 9 2010, pp. 217–222.

[163] H. Ohlsson and L. Ljung, "Identification of switched linear regression models using sum-of-norms regularization," *Automatica*, vol. 49, no. 4, pp. 1045–1050, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109813000320

[164] E. Ozkan, F. Lindsten, C. Fritsche, and F. Gustafsson, "Recursive maximum likelihood identification of jump markov nonlinear systems," *IEEE Transactions on Signal Processing*, vol. 63, no. 3, pp. 754–765, 2015.

[165] G. Padfield, *Helicopter Flight Dynamics: The Theory and Application of Flying Qualities and Simulation Modeling*, ser. AIAA education series. American Institute of Aeronautics and Astronautics, 2007. [Online]. Available: https://books.google.nl/books?id=kTNMPgAACAAJ

[166] E. Panteley and A. Loría, "Growth rate conditions for uniform asymptotic stability of cascaded time-varying systems," *Automatica*, vol. 37, no. 3, pp. 453–460, 2001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109800001692

[167] M. A. Patterson and A. V. Rao, "Gpops-ii: A matlab software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming," *ACM Trans. Math. Softw.*, vol. 41, no. 1, Oct. 2014. [Online]. Available: https://doi.org/10.1145/2558904

[168] PwC. (2018, 11) Flying high. Engineering Council of India. [Online]. Available: https://www.pwc.in/assets/pdfs/publications/2018/flying-high.pdf

[169] C. Rasmussen and C. Williams, *Gaussian processes for machine learning*. Cambridge, Mass: MIT Press, 2006.

[170] R. Ritz, M. Hehn, S. Lupashin, and R. D'Andrea, "Quadrocopter performance benchmarking using optimal control," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 5179–5186.

[171] X. Rong Li and V. P. Jilkov, "Survey of maneuvering target tracking. part v. multiple-model methods," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1255–1321, 10 2005.

[172] G. Rousseau, C. S. Maniu, S. Tebbani, M. Babel, and N. Martin, "Quadcopter-performed cinematographic flight plans using minimum jerk trajectories and predictive camera control," in *2018 European Control Conference (ECC)*, 2018, pp. 2897–2903.

[173] B. Rubí, R. Pérez, and B. Morcego, "A survey of path following control strategies for UAVs focused on quadrotors," *Journal of Intelligent & Robotic Systems*, vol. 98, no. 2, pp. 241–265, 9 2019. [Online]. Available: https://doi.org/10.1007/s10846-019-01085-z

[174] S. Sabatelli, M. Galgani, L. Fanucci, and A. Rocchi, "A double-stage kalman filter for orientation tracking with an integrated processor in 9-d imu," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 3, pp. 590–598, 3 2013.

[175] A. Sabatini, "Quaternion-based extended kalman filter for determining orientation by inertial and magnetic sensing," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 7, pp. 1346–1356, 2006.

[176] A. M. Sabatini and V. Genovese, "A sensor fusion method for tracking vertical velocity and height based on inertial and barometric altimeter measurements," *Sensors*, vol. 14, no. 8, pp. 13 324–13 347, 2014. [Online]. Available: https://www.mdpi.com/1424-8220/14/8/13324

[177] A. Saberi, A. A. Stoorvogel, and P. Sannuti, *Internal and External Stabilization of Linear Systems with Constraints*. Birkhäuser Boston, 2012. [Online]. Available: https://doi.org/10.1007/978-0-8176-4787-2

[178] B. Sabetghadam, A. Alcántara, J. Capitán, R. Cunha, A. Ollero, and A. Pascoal, "Optimal trajectory planning for autonomous drone cinematography," in *2019 European Conference on Mobile Robots (ECMR)*, 2019, pp. 1–7.

[179] A. Sanyal and N. Chaturvedi, "Almost global robust attitude tracking control of spacecraft in gravity," in *AIAA Guidance, Navigation and Control Conference and Exhibit*. American Institute of Aeronautics and Astronautics, 6 2008. [Online]. Available: https://doi.org/10.2514/6.2008-6979

[180] G. Saunders, *Dynamics of helicopter flight*. New York: Wiley, 1975.

[181] M. G. Scott. That's what she said. The Office. [Online]. Available: https://youtu.be/1mipOFszqNE?t=105

[182] K. D. Sebesta and N. Boizot, "A real-time adaptive high-gain ekf, applied to a quadcopter inertial navigation system," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 1, pp. 495–503, 1 2014.

[183] B. Silver, M. Mazur, A. Wisniewski, and A. Babicz, "Welcome to the era of drone-powered solutions: a valuable source of new revenue streams for telecoms operators," *PWC Communications Review*, 2017.

[184] H. Simon, "Adaptive filter theory," *Prentice Hall*, vol. 2, pp. 478–481, 2002.

[185] A. M. Sjøberg and O. Egeland, "Lie algebraic unscented kalman filter for pose estimation," 2020, [arXiv:2005.00385, April 2020].

[186] S. Spedicato and G. Notarstefano, "Minimum-time trajectory generation for quadrotors in constrained environments," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 4, pp. 1335–1344, 2018.

[187] J. K. Stolaroff, C. Samaras, E. R. O'Neill, A. Lubers, A. S. Mitchell, and D. Ceperley, "Energy use and life cycle greenhouse gas emissions of drones for commercial package delivery," *Nature Communications*, vol. 9, no. 1, 2 2018. [Online]. Available: https://doi.org/10.1038/s41467-017-02411-5

[188] Q. Sun, C. Lim, P. Shi, and F. Liu, "Design and stability of moving horizon estimator for markov jump linear systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 3, pp. 1109–1124, 3 2019.

[189] A. Svensson, T. B. Schön, and F. Lindsten, "Identification of jump markov linear models using particle filters," in *53rd IEEE Conference on Decision and Control*, 2014, pp. 6504–6509.

[190] S. Taamallah, "A qualitative introduction to the vortex-ring-state, autorotation, and optimal autorotation," in *Proceedings of the 36th European rotorcraft forum*. National Aerospace Laboratory NLR, 2010.

[191] S. Taamallah, X. Bombois, and P. M. Van den Hof, "Trajectory planning and trajectory tracking for a small-scale helicopter in autorotation," *Control Engineering Practice*, vol. 58, pp. 88–106, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0967066116301782

[192] A. Talaeizadeh, H. N. Pishkenari, and A. Alasty, "Quadcopter fast pure descent maneuver avoiding vortex ring state using yaw-rate control scheme," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 927–934, 2021.

[193] A. Talaeizadeh, D. Antunes, H. N. Pishkenari, and A. Alasty, "Optimal-time quadcopter descent trajectories avoiding the vortex ring and autorotation states," *Mechatronics*, vol. 68, p. 102362, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957415820300428

[194] G. Tang, W. Sun, and K. Hauser, "Learning trajectories for real- time optimal control of quadrotors," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 3620–3625.

[195] T. Tomić, M. Maier, and S. Haddadin, "Learning quadrotor maneuvers from optimal control and generalizing in real-time," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1747–1754.

[196] J. Turner, P. Kenkel, R. B. Holcomb, and B. Arnall, "Economic Potential of Unmanned Aircraft in Agricultural and Rural Electric Cooperatives," Southern Agricultural Economics Association, 2016 Annual Meeting, February 6-9, 2016, San Antonio, Texas 230047, 2016.

[197] J. Vaganay and M. Aldon, "Attitude estimation for a vehicle using inertial sensors," *Control Engineering Practice*, vol. 2, no. 2, pp. 281–287, 1994. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0967066194902097

[198] R. G. Valenti, I. Dryanovski, and J. Xiao, "Keeping a good attitude: A quaternion-based orientation filter for imus and margs," *Sensors*, vol. 15, no. 8, pp. 19 302–19 330, 2015.

[199] M. van Nieuwstadt and R. M. Murray, "Real time trajectory generation for differentially flat systems," *IFAC Proceedings Volumes*, vol. 29, no. 1, pp. 2301 – 2306, 1996, 13th World Congress of IFAC, 1996, San Francisco USA, 30 June - 5 July.

[200] H. Van Vyve, "Simulation of a helicopter in vortex ring state through a coupled simulation of multi-body dynamics and aerodynamics," Ph.D. dissertation, UCL - Ecole polytechnique de Louvain, 2019. [Online]. Available: http://hdl.handle.net/2078.1/thesis:19581

[201] J. F. Vasconcelos, B. Cardeira, C. Silvestre, P. Oliveira, and P. Batista, "Discrete-time complementary filters for attitude and position estimation: Design, analysis and experimental validation," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 1, pp. 181–198, 2011.

[202] R. Vidal, "Recursive identification of switched arx systems," *Automatica*, vol. 44, no. 9, pp. 2274–2287, 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109808001088

[203] P. Voosen, "Nasa to fly drone on titan," *Science*, vol. 365, no. 6448, pp. 15–15, 2019. [Online]. Available: https://science.sciencemag.org/content/365/6448/15.1

[204] G. Wahba, "A least squares estimate of satellite attitude," *SIAM Review*, vol. 7, no. 3, pp. 409–409, 1965.

[205] X. Wang, H. F. Grip, A. Saberi, and T. A. Johansen, "A new low-and-high gain feedback design using mpc for global stabilization of linear systems

subject to input saturation," in *2012 American Control Conference (ACC)*, 6 2012, pp. 2337–2342.

[206] X. Wang, A. Saberi, A. A. Stoorvogel, and P. Sannuti, "Simultaneous global external and internal stabilization of linear time-invariant discrete-time systems subject to actuator saturation," *Automatica*, vol. 48, no. 5, pp. 699 – 711, 2012.

[207] D. Weber, C. Gühmann, and T. Seel, "Neural networks versus conventional filters for inertial-sensor-based attitude estimation," 2020, [arXiv preprint arXiv:2005.06897, June 2020].

[208] J.-Y. Wen and K. Kreutz-Delgado, "The attitude control problem," *IEEE Transactions on Automatic Control*, vol. 36, no. 10, pp. 1148–1162, 1991.

[209] O. Westbrook-Netherton and C. Toomer, "An investigation into predicting vortex ring state in rotary aircraft," 2014.

[210] A. Wisniewski and M. Mazur, "Clarity from above pwc global report on the commercial applications of drone technology," Tech. rep. May 2016. URL: http://www. pwc. pl/pl/pdf/clarity-from-above-pwc . . . , Tech. Rep., 2016.

[211] X. Wu and K. Ma, "Attitude estimation based on robust information cubature quaternion filter," *Circuits, Systems, and Signal Processing*, vol. 39, p. 2948–2967, 2020.

[212] M. Xue and I. A. Hiskens, "Alternative strategies for designing stabilizing model predictive controllers," in *52nd IEEE Conference on Decision and Control*, 12 2013, pp. 4491–4497.

[213] F. Yacef, N. Rizoug, L. Degaa, O. Bouhali, and M. Hamerlain, "Trajectory optimisation for a quadrotor helicopter considering energy consumption," in *2017 4th International Conference on Control, Decision and Information Technologies (CoDIT)*, 2017, pp. 1030–1035.

[214] F. Yacef, N. Rizoug, L. Degaa, and M. Hamerlain, "Energy-efficiency path planning for quadrotor uav under wind conditions," in *2020 7th International Conference on Control, Decision and Information Technologies (CoDIT)*, vol. 1, 2020, pp. 1133–1138.

[215] Q.-q. Yang, L.-l. Sun, and L. Yang, "A fast adaptive-gain complementary filter algorithm for attitude estimation of an unmanned aerial vehicle," *Journal of Navigation*, vol. 71, no. 6, p. 1478–1491, 2018.

[216] T. S. Yoo, S. K. Hong, H. M. Yoon, and S. Park, "Gain-scheduled complementary filter design for a mems based attitude and heading reference system," *Sensors*, vol. 11, no. 4, pp. 3816–3830, 2011.

[217] W. Yoo, E. Yu, and J. Jung, "Drone delivery: Factors affecting the public's attitude and intention to adopt," *Telematics and Informatics*, vol. 35, no. 6, pp. 1687–1700, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0736585318300388

[218] S. Zhang, S. Yu, C. Liu, X. Yuan, and S. Liu, "A dual-linear kalman filter for real-time orientation determination system using low-cost mems sensors," *Sensors*, vol. 16, no. 2, p. 264, 2 2016.

[219] W. Zhang, J. Hu, and A. Abate, "Infinite-horizon switched lqr problems in discrete time: A suboptimal algorithm with performance analysis," *IEEE Transactions on Automatic Control*, vol. 57, no. 7, pp. 1815–1821, 7 2012.

[220] X. Zhang, Y. Fang, X. Zhang, P. Shen, J. Jiang, and X. Chen, "Attitude-constrained time-optimal trajectory planning for rotorcrafts: Theory and application to visual servoing," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 4, pp. 1912–1921, 2020.

[221] P. Zhao, Y. Kang, and Y. Zhao, "A brief tutorial and survey on markovian jump systems: Stability and control," *IEEE Systems, Man, and Cybernetics Magazine*, vol. 5, no. 2, pp. 37–C3, 2019.

[222] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, "Robust and efficient quadrotor trajectory generation for fast autonomous flight," *CoRR*, vol. abs/1907.01531, 2019. [Online]. Available: http://arxiv.org/abs/1907.01531

[223] X. Zhou, Z. Yi, Y. Liu, K. Huang, and H. Huang, "Survey on path and view planning for uavs," *Virtual Reality & Intelligent Hardware*, vol. 2, no. 1, pp. 56–69, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2096579620300073

[224] A. Zulu and S. John, "A review of control algorithms for autonomous quadrotors," *Open Journal of Applied Sciences*, vol. 04, no. 14, pp. 547–556, 2014. [Online]. Available: https://doi.org/10.4236/ojapps.2014.414053

Summary

Optimization-based Estimation and Control Algorithms for Quadcopter Applications

This thesis presents novel optimization-based estimation and control algorithms. While these algorithms are primarily intended for quadcopter applications, some of these algorithms are more general and can be of interest in a broader range of applications. Considering the expected increase in the usage of quadcopters and their new application domains, demands on their performance in terms of speed, accuracy, reliability and robustness are likely to increase. In order to meet these new demands, improvements on current methods are required, of which this thesis provides several.

One of directions in which improvement is required is that of estimation of the state of the quadcopter, which is covered in the first part of the thesis. Accurate attitude and position estimation is important for the control of quadcopters, in order to know their configuration with respect to the environment they are operating in. In this thesis, novel estimation algorithms are proposed for both attitude and position estimation. For attitude estimation, the challenge of accurate estimation that is robust to disturbances in the accelerometer and magnetometer measurements is considered. An adaptive complementary filter is proposed that fuses measurements from a gyroscope and vector-based measurements, which are provided by accelerometers and/or magnetometers. The filter is posed directly on the Special Orthogonal Group, and estimates the disturbances in the vector-based measurements. These estimates are then used to adapt the gains of the filter to rely more on the measurements with low noise levels. As a result, the method enhances non-adaptive methods while still providing similar convergence guarantees for the estimation errors, as will be shown in this thesis.

Concerning position estimation, the issues arising from intermittent availability of positioning sensors such as GPS or Ultra-wideband (UWB) systems

are addressed. The estimation problem is formulated in a considerably broader framework than just the quadcopter context, namely that of state estimation for Markov Jump Linear Systems (MJLSs). A novel approach for this general problem relying on Relaxed Dynamic Programming techniques is proposed, leading to near-optimal estimates within a pre-specified bound. The proposed method is also applied to the identification of MJLSs.

Another important area of improvement in quadcopters is that of planning and control, which is covered in the second part of this thesis. Good planning algorithms are essential in order to ensure that the trajectories generated for the quadcopter are fast, efficient and safe. In this thesis, a new method is proposed for the computation of both time-optimal and energy-optimal vertical descent trajectories. These trajectories are computed while explicitly taking into account the aerodynamic phenomenon known as the Vortex Ring State (VRS). This effect occurs during fast descent trajectories and can cause a loss of control of the quadcopter, which can result in a crash. The VRS is carefully modeled and is explicitly taken into account in the determination of both time-optimal and energy-optimal vertical descent trajectories.

The control part of the thesis investigates trajectory tracking for quadcopters. Accurate and robust trajectory tracking is important to ensure that planned trajectories are followed even when disturbances are presented. For the trajectory tracking problem a novel model predictive control (MPC) strategy is proposed. The problem is separated into an outer-loop controller handling the translational dynamics and an inner-loop controller that handles the attitude dynamics. For the inner loop a nonlinear controller is used, while for the outer loop MPC is applied. This MPC strategy uses an input transformation that allows the use of results on globally stabilizing control for linear systems subject to input constraints. In this way the MPC strategy allows for reference tracking while respecting constraints on the thrust of the quadcopter. Results that render the trajectory tracking errors almost globally asymptotically stable are provided for the overall strategy.

Dankwoord

Ik heb veel dingen geleerd tijdens mijn tijd als promovendus, maar een van de dingen die (beginnend) wetenschappers altijd op hun hart wordt gedrukt is het citeren van je bronnen. Hieronder wil ik dan ook graag de belangrijkste (niet-wetenschappelijke) bronnen van mijn promotietijd benoemen.

First and foremost, I'd like to thank my copromotor and daily supervisor. Duarte, I can very honestly say that without you this dissertation would have never made it to completion. From the start it was you that offered me this position and from that moment onward you never lost faith in a successful outcome, even though there were moments where there were doubts, often coming from me, sometimes from others. Your incredibly positive attitude and kindness towards others and myself have continued to impress me these last five years, and together with your humor, dedication and (sometimes a bit chaotic) work ethic, have made working with you a real pleasure. The time you spent helping me, even when you had none, and the confidence and trust you gave me during this project were key to its successful completion. For this and all the good times together, muito obrigado.

Maurice, de afgelopen jaren heb jij als promotor altijd jouw oplossend vermogen ingezet als het weer eens wat minder ging met het onderzoek. Jouw positieve instelling, passie voor het vak en humor zorgden er altijd voor dat ik wijzer en met meer enthousiasme je kantoor verliet. Jouw ongelofelijk scherpe commentaar hebben onze publicaties en dit proefschrift naar een hoger niveau getild. Dat jij na zoveel jaren en zoveel papers nog altijd zo scherp op de details zit is echt indrukwekkend, alhoewel ik niet altijd even blij was met een met rode pen doorkruiste pdf ;). Bedankt voor alles Maurice!

I would like to thank Nathan van de Wouw, Bayu Jayawardhana, Boris Houska and Ivana Palunko for the time and effort spent reading and assessing this thesis, providing comments that improved its quality and participating in my defense committee.

Maarten, bedankt dat je als voorzitter wil plaatsnemen in mijn promotiecommissie en voor alle ondersteuning tijdens mijn promotie en master studie. Jouw vermogen om snel tot de kern van het probleem te komen zorgde ervoor dat ik

altijd na 10 minuten weer buiten je kantoor stond, maar dat al mijn vragen wel beantwoord waren. Door jouw focus op mensen was de CST groep altijd een fijne plek om te studeren en werken.

Erjen, bedankt voor de hulp met het oplossen van de (uiteindelijk toch wel vele) problemen in het MPC hoofdstuk en het rond krijgen van het bewijs. Ik heb veel geleerd en erg genoten van de vele uren die wij afgelopen jaar hebben doorgebracht in jouw kantoor om samen aan het bewijs te klussen.

I would like to thank all the current and former colleagues of the CST and D&C groups that I had the pleasure of working with for the many coffee breaks, great team outings and fun during the Benelux meetings. Nancy en Roos, bedankt voor alle hulp tijdens mijn promotie en het organiseren van alle team activiteiten, dankzij jullie blijft de sfeer in de groep, zelfs nu, heel goed. Mijn kantoorgenoten van afgelopen jaren, Frans, Robert, Wouter, Lu, Michiel, Roy, Koen en Manuel, wil ik graag bedanken voor de wetenschappelijke, en vooral de niet wetenschappelijke, gesprekken. Also thanks to my temporary, but very much valued, office mate Chyannie, for the support and for going to the struggle of finishing a thesis together with me. You beat me by a long shot, though ;). Ook de lunchwandeling groep, Max, Max, Leontine, Joey, Koen en zowel Nard als Noud, wil ik graag bedanken voor de gesprekken over de beurs, huizen en noedels. Maurice (Mo!), bedankt voor de filosofische gesprekken en het delen van mooie ervaringen, en natuurlijk de sokken! Sebastiaan, bedankt voor je hulp bij de bewijzen. Jordy, enne jong? Bedankt voor de vele (digitale) koffiepauzes en gesprekken over hoe alles zou moeten zijn op de TU/e en tijdens het promoveren, en voor de interesse in mijn avonturen.

My thanks also go to all the great people working at Avular, it has been a pleasure seeing the company grow from (less than?) 5 people to what I think is now around 40 during the course of my PhD. In het bijzonder wil ik de mannen van het eerste uur, Albert, Joop, Yuri en Pasquale bedanken voor de prettige samenwerking, de legendarische team unions, heerlijke humor, spontane Kanye platen en al het gouuud. Alhoewel mijn onderzoek al snel een andere kant op ging dan jullie bedrijf heb ik me altijd bijzonder welkom bij jullie gevoeld, dank daarvoor.

Dan gaat mijn volgende dank uit naar wat technisch gezien collega's waren, maar vooral mijn beste vrienden zijn. Leroy, Daniel, Robin, ik weet oprecht niet waar ik moet beginnen, en dat overkomt me niet vaak. Bedankt voor de hulp bij mijn onderzoek, jullie hebben als collega's was het beste van mijn promotie. Maar vooral bedankt voor alle avonden chillen, slap ouwehoeren, optimaal Catannen, beurspraat, logische kronkels, enzovoorts. Cameron hier hoor jij ook zeker bij, bedankt man! Leroy, jou wil ik nog in het bijzonder bedanken voor het zijn van paranimf, maar ook voor alle trips die wij afgelopen jaren samen hebben gedaan, van surfen en duiken in verre plekken, tot sterren kijken en mountainbiken in de buurt. Helaas is het er door wat ongelukkige omstandigheden de laatste tijd niet meer van gekomen, maar laten we dit snel weer doen!

Ik wil ook nog even de dames bedanken, Rosanne, Patricia, Sadia en Imane, bedankt voor de goede zorgen voor de mannen en mijzelf, de gastvrijheid en het ons (terecht) laten weten als we weer eens een ontzettende nerd discussie aan het voeren waren.

Natuurlijk kan ik de mannen van het Ardennen offensief / de wintersport niet vergeten, Pieter, Ivo D, Dirk Jan, Du, Jan, Marius, Sander, Dino, Ivo P, Geert en natuurlijk mijn beste afstudeerder $\|\text{Bas}\|_2$. Bedankt voor de vele mooie klus sessies, weekenden Ardennen, weken wintersport, daydrink activiteiten, sigaren vergaderingen, schreeuwende discussies, oudjaar vieringen en carnavalstradities. Oja en halve leos uiteraard, was het bijna vergeten.

De mannen van het dispuut wil ik bedanken voor jullie belangrijke aandeel in het vormen van de persoon die ik nu ben, en natuurlijk voor alle mooie activiteiten over de jaren: de borrels, het zeilen, de wintersporten, KMT weekenden, vakanties, MTB trips, immer mooie ALVs en inmiddels reünistenactiviteiten. Het was, en is, mij een waar genoegen.

I want to thank my fellow marranos, Jordan and David, for the best trips of my life (so far). Jordan, bedankt voor alle dingen die ik van je heb geleerd en het (soms letterlijk) tonen van de weg. David and Andrea, thank you for the hospitality you provided us in Guatemala and during your wedding, it was an honor to be one of your best men David! Shout-out to Maria and Keith, for going on an epic, but very tough trip with us (without complaining for a second!).

Ik wil de familie aan de Timmers kant bedanken voor de mooie vakanties door de jaren heen en de altijd gezellige koffie op zondagochtend bij oma. De familie aan de Andriën kant wil ik bedanken voor de gastvrijheid, alhoewel ik lang niet vaak genoeg langskom, voel ik me altijd zeer welkom in Limburg.

Als laatste wil ik mijn direct familie bedanken. Eduard, bedankt voor het zijn van een voorbeeld waar ik naar op kan kijken en het banen van de weg. Manouk, bedankt voor de goede zorgen voor Ed en Lou, en voor het zijn van een perfecte aanvulling op onze familie. Judith, bedankt voor alle steun over de afgelopen jaren en het mij met beide benen op de grond houden als dat nodig is. En weet dat ik trots op je ben. Pap, bedankt voor de kansen die je mij hebt gegeven en voor het altijd klaar staan, wat ik ook nodig heb. Mam, bedankt voor je mooie blik op het leven, het altijd willen aanhoren van mijn problemen en de liefde die je mij altijd hebt gegeven.

Alex Andriën
Eindhoven, 30 December 2021

About the author

Alex Andriën was born on June 22nd 1990, in Heeze, the Netherlands. After finishing his secondary education at 2College Durendael in Oisterwijk, he studied Mechanical Engineering at the Eindhoven University of Technology, where he received his Bachelor of Science degree in 2013. He then studied Systems and Control, for which he received his Master of Science degree in 2016 (with great appreciation). As part of his master's studies he performed a six-month internship at the Massachusetts Institute of Technology (MIT) in 2014 under the supervision of David Trumper, where he worked on a printed circuit board design of a three mass cantilever beam system as well as the development of a brushless DC motor controller, combined with a teaching assistant position for the Mechatronics course. His master's graduation project, entitled 'Lazy Motion Planning for Robotic Manipulators', was performed under the supervision of René van de Molengraft en Herman Bruyninckx and focused on using the environment to reduce the energy used by a 7-DOF robotic manipulator while performing grasping tasks. The project involved electronics, FPGA programming, (low-level) motor control, modeling and optimization.

In January 2017, Alex started his Ph.D. research in the Control Systems Technology group at the department of Mechanical Engineering at the Eindhoven University of Technology, under the supervision of Maurice Heemels and Duarte Antunes. His research project is a collaboration between the Eindhoven University of Technology and Avular, and is part of the Drone Safety Cluster project and is (partly) funded by the European Union through the OPZuid program. The main results of his research are presented in this dissertation.