Inertial stabilization, estimation and visual servoing for aerial surveillance

by

Ing. Martin Řezáč

supervised by Ing. Zdeněk Hurák, Ph.D.

Dissertation

Presented to the Faculty of Electrical Engineering of

Czech Technical University in Prague

in Partial Fulfillment of the Requirements

for the Degree of

Doctor

in Ph.D. programme

Electrical Engineering and Information Technology

in the branch of study

Control Engineering and Robotics

Czech Technical University in Prague

October 2013

To my parents

Acknowledgement

First and the foremost, I would like to thank to Zdeněk Hurák for being a caring and enthusiastic supervisor. He was always ready for discussions, giving me not only a useful technical advise and a hint to a relevant literature but also a wealth of encouragement. His Advanced Algorithms for Control and Communications (AA4CC) group at the Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague was a stimulating place for several years.

I am grateful for the opportunity to work in a broader industrial project consortium formed by a team from Czech Air Force and Air Defense Technological Institute (VTÚLaPVO) lead by Jiří Nohýl, a team from ESSA company lead by Milan Bartoš, a team from Center for Machine Perception (CMP) lead by Václav Hlaváč and our AA4CC team lead by Zdeněk Hurák. This gave my research activities a true multidisciplinary engineering flavor. Negotiating major modifications of the mechanical design with a mechanical engineer, planning the flight tests with the Air Force people, matching the performance of the camera system to that of the inertial stabilization loop with a computer vision expert, all these were very enriching.

In order to keep my own research in algorithms and methods relevant from an engineering point of view, I relied a lot on both indoor and outdoor experiments with various hardware platforms. I am happy to acknowledge numerous assistance from my colleagues at the A44CC group. Above all, the successful accomplishment of my work was totally conditioned by the excellent work of the key developer of electronics — Jaroslav Žoha. I also feel privileged that I have witnessed a few of our graduate students growing into experienced control systems developers, namely Jan Salášek, Ondřej Mikulín and Jaromír Dvořák. Collaboration with these young engineers was a pleasure.

Although not immediately involved in the related research, Jiří Zemánek — a doctoral student at the AA4CC group — brought a lot of inspiration to me in diverse areas of engineering and science. I really enjoyed sharing a part of PhD student's life with him.

Finally, I would like to express the gratitude to my parents and my girlfriend Lenka for their love and patience not only during the time I was finishing this thesis.

MARTIN ŘEZÁČ

Czech Technical University in Prague
October 2013

Inertial stabilization, estimation and visual servoing for aerial surveillance

Ing. Martin Řezáč
Czech Technical University in Prague, 2013

Supervisor: Ing. Zdeněk Hurák, Ph.D.

This thesis addresses a few technical problems that are all related to the central topic of inertial stabilization of the optical axis of an airborne camera system (so-called line-of-sight stabilization). First, the task of augmentation of the classical inertial stabilization loop with an automatic visual tracking loop is solved in a systematic way invoking the concepts from the young discipline of visual servoing. Second, the issue of computer-vision-induced delays was tackled and an intuitive engineering solution was formulated as a special instance of a reset control, a modified Smith predictor and a multirate control, which allowed to propose more efficient solutions. Third, the advanced mechanical configuration with two motors actuating the rotation of the optical payload around a common axis (so-called dual-stage configuration) was studied and recent numerical optimization tools for structured $\mathcal{H}_\infty$-optimal controller design were used to provide controllers with a performance superior to the one obtained with classically tuned PID loops. Definition and selection of these research topics were motivated by a collaboration with an industrial partner within a series of projects. Therefore the thesis also documents a few routine engineering results for completeness. Namely, mathematical modeling of the kinematics and dynamics of common mechanical configurations such as double-gimbal and dual-stage configurations, inertial estimation, design of a feedforward disturbance rejection controller and input command shaping filters. All the algorithms were always implemented in one of the available benchmark systems and tested either in a laboratory or even during helicopter flights.

Contents

Chapter 1

Introduction

This thesis addresses a few technical problems that are all related to the central topic of inertial stabilization of the optical axis of an airborne camera system (the so-called line-of-sight stabilization). They span several sub-disciplines, namely inertial stabilization, inertial estimation and visual servoing.

Definition and selection of these research topics were motivated by a collaboration with an industrial partner within a series of projects. Therefore the thesis also documents few routine engineering results for completeness. All the algorithms were always implemented in one of the available benchmark systems and tested either in a laboratory or even during helicopter flights.

1.1 PROBLEM STATEMENT

The typical scenario that is studied in this thesis from various control engineering viewpoints can be described using Fig. 1.1 as follows. Having an optical payload (day vision camera, night vision camera, laser rangefinder, ...) mounted onto a mobile carrier such as an unmanned aircraft, helicopter or truck, the task for the control system is to ensure that the optical axis of the payload is stationary in the inertial space even when the carrier is a subject to an unwanted and usually unpredictable movement. In addition, some other disturbing phenomena like wind-induced torque are acting on the payload. The requirement of stationarity of the commanded optical axis is commonly denoted as an *inertial line-of-sight stabilization*. The device that is performing the line-of-sight stabilization is called the *inertially stabilized platform* in this thesis.

Although the particular mechanical configuration in Fig. 1.1 allows rotation of the camera with respect to its carrier around two axes only, there are many other configurations available. A rigorous analysis of a few of the most relevant mechanical configurations including their mathematical models is given in this thesis together with their capabilities to attenuate the influence of the (disturbing) carrier motion on the stationarity of the line of sight.

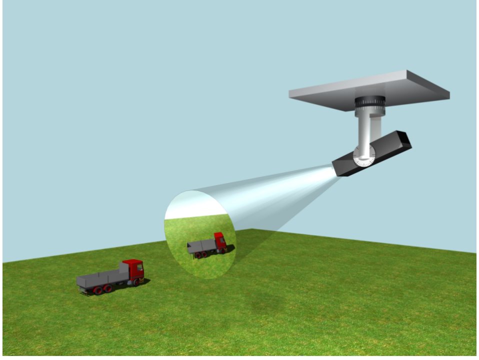


Figure 1.1: Typical scenario for the inertial stabilization system. An optical device (camera) onboard the plane is used in a surveillance mission. As a part of the mission, it is requested to keep a selected ground target in the center of the field of view in spite of the (disturbing) motion of the carrier (the aircraft) and the motion of the object on the ground.

When the payload to be inertially stabilized contains a camera, some more advanced features such as an image-based tracking may be implemented. The incorporation of the camera in the feedback loop introduces a few technical issues and it turns out that some benefit can be obtained by applying the results from the modern discipline of *visual servoing*. However, some clever modifications of the classical (robotics related) visual servoing theory are needed since it typically assumes that the robot sits on the ground and the image coordinate frame can be easily related with the coordinate frame of the individual joints. In aerial applications the angles or angular velocities of the joints are irrelevant from the viewpoint of positioning the image of the object in the image plane because the carrier itself can move (rotate and translate).

Another challenge from a control-systems viewpoint is the issue of a one-sampling-period delay introduced into the pointing loop by the resource-intensive image-tracking routine. Although there are numerous techniques for compensation of a delay in the feedback loop, here the situation is special in that undelayed measurements from some other available sensors can be used to compensate at least for the rotation of the carrier during the time of processing the image information.

Last but not least, there is a task that can be viewed as the *inverse* to the line-of-sight stabilization, that is, the task of determining the coordinates of the (ground) target tracked by the optical system. The terminology actually coincides with the *inverse kinematic task* in robotics. The crucial step is the *inertial estimation of the attitude* of the carrier relying on sensors such as gyros, accelerometers and magnetometers. Supplementing the estimated carrier attitude with its GPS coordinates, the joint/gimbal angles, and the measurement of the distance to the target, one may easily calculate the GPS coordinates of the target on the ground.

1.2 SHORT HISTORY OF THE RELATED PROJECTS SOLVED AT CTU

On the ground of Faculty of Electrical Engineering, Czech Technical University in Prague, the research (and partially development too) in the domain of inertial stabilization was initiated in 2007 when the Czech Air Force and Air Defence Technological Institute (in Czech: Vojenský technický ústav letectva a PVO, VTÚLaPVO) approached the university with an offer of a contract. The goal was to develop a new aerial inertially stabilized camera platform tailored to a new UAV developed by VTÚLaPVO at that time. The team comprising control specialists and computer vision experts was established. Namely, the Center of Machine Perception (CMP) lead by Prof. Václav Hlaváč accepted the computer vision part of the task and the group of Dr. Zdeněk Hurák from Department of Control Engineering accepted the responsibility for the inertial stabilization part. A small mechanical engineering company ESSA located in Prague was invited to the project consortium with the motivation to guarantee a professional-grade mechanical design. As a result, a functional prototype of a platform was developed from the very scratch and provided to VTÚLaPVO in about a two-year time. It was at this time that I joined the group as a first-year graduate student.

Encouraged by the successful accomplishment of this first project, the whole consortium searched for other opportunities and these were identified in both larger and smaller scales of inertially stabilized camera platforms. Two more projects have been started then, both financially supported by the Ministry of Industry and Trade of the Czech Republic with VTÚLaPVO becoming the key project partner. The more detailed description of the outcomes of these projects will be given in later sections, suffice to say now that at the time of submitting this thesis (fall 2013) heavy efforts are invested by VTÚLaPVO into commercialization.

I was actively involved in these projects as a doctoral student till the end of summer 2012, even though I was free of the standard hardware and software development duties and instead I could focus on research in algorithms and control design methods.

1.3 STATE OF THE ART IN INERTIAL STABILIZATION FOR AERIAL APPLICATIONS

1.3.1 Academic publications

The design of control systems for stabilization of the line of sight of the optical payload was extensively studied by the engineering community during past decades but it is not always easy to document the history of this defense-related engineering discipline purely from academic publications. An up-to-date survey can be found in the February 2008 issue of the *IEEE Control Systems Magazine*, where the whole issue was dedicated to the topic of "Inertially stabilized platform technology".

The most focused regular forum for discussion of technical issues related to inertial stabilization for aerial applications seems to be *SPIE Defense, Security, and Sensing* conference held every year. Some interesting papers can be found in their proceedings or in one of the journals published by SPIE, in particular *Optical Engineering*.

Since this thesis includes a collection of research results that are only loosely related by the common goal, it is convenient to place the particular state-of-the-art sections in appropriate chapters.

1.3.2 Commercial products

There are several companies engaged in the production of inertially stabilized platforms. Among the most well-known belong companies like FLIR, L-3 WESCAM, General Dynamics, DST control, HOOD Tech Vision or Cloud Cap Technology. Parameters and the equipment of their products are varying depending on the model. Photos of some stabilized platforms are shown in Fig. 1.2. What is common for all of them is the brevity in technology principles description and obviously no information at all about the algorithms inside since these belong to the proprietary know-how.

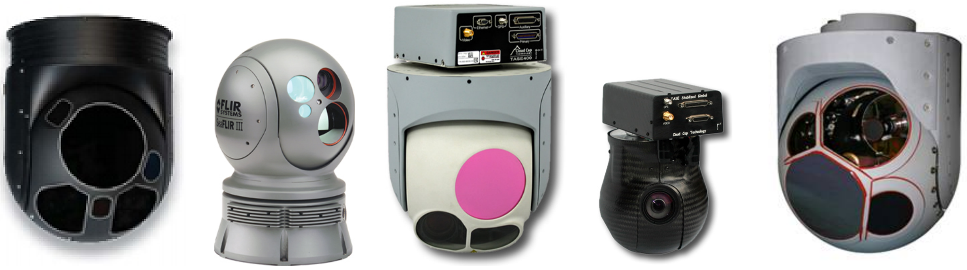


Figure 1.2: Products of competitors. Stabilized platforms developed by FLIR, Cloud Cap Technology and WESCAM companies. Photos are from companies' websites (`http://www.flir.com`, `http://www.cloudcaptech.com` and `http://www.wescam.com/` respectively). Names of products from the left: FLIR *Star SAFIRE 380-HD*, FLIR *SeaFLIR III*, Cloud Cap Technology *TASE400*, Cloud Cap Technology *TASE150* and L-3 WESCAM *MX-20*.

The conference *SPIE Defense, Security, and Sensing* that was mentioned in the previous section is held usually jointly with an exhibition of companies that are engaged in the field of UAVs, Pan-Tilt systems, stabilized platforms and optical and infrared imaging and sensing. Basically all of the mentioned companies present their products at this event. I had an opportunity to attend this event in the year 2010, my colleague Jan Salášek in the year 2013. Devices that were presented at this event may be divided into three groups:

Pan-Tilt systems are designed to allow precise pointing and tracking, but they are designed to be mounted on fixed not moving carrier (ground). They achieve a major precision of the pointing but only with respect to their carrier — that is not with respect to the inertial system. The equipment of these systems is usually arbitrary.

Low-cost stabilized systems are typically designed as a low-size and low-cost devices. They contain usually only a single camera. Due to space, weight and low-cost requirements they are typically equipped with MEMs gyros and allow only a limited gimbal rotation. They use only the two gimbals configuration. Typical representatives are *OTUS-U135* from DST control or *TASE150* from Cloud Cap Technology.

Superior stabilized systems are devices with the superior precision of the inertial stabilization. They usually use the multi-gimbal configuration (four and more gimbals) with inner gimbals equipped with direct drives or other gearless torque actuators. In order to achieve the best precision they are typically equipped with fibre optic gyros. Due to significant size they are capable to carry more optical devices — typically one or two optical cameras, infrared camera, laser rangefinder and pointer. Typical representatives are products from the *Star SAFIRE Family* from FLIR, *TASE400* from Cloud Cap Technology or *MX-20* from L-3 WESCAM.

What makes the comparison of most of presented devices complicated is the absence of any standard for the measurement of the precision. The most of the leaders declare the precision down to units or tens of microradians. But in the end none of the company representatives was willing to clarify how the parameter is measured. After a discussion with few of lecturers at SPIE Defense, Security, and Sensing conference it turned out, that since the conditions under which these parameters are measured are not specified, it usually means that the parameter is just measured angular variance when the device is fixed to the ground. Some examples of stabilization descriptions:

TASE400 pointing resolution 14μrad, gimbal stabilization $< 75\mu$rad

Vector-50 gyro stabilization $<$ 1milirad RMS

MX-20 $< 4\mu$rad

The only exception is the company HOOD Tech Vision that provides the stability precision as the attenuation at frequencies 1 and 2 Hz. *Hood Tech 09MWIR2* system offers the attenuation 39 dB at 1 Hz and 38 dB at 2 Hz.

1.4 CONTRIBUTION OF THE THESIS

The main contribution of this thesis resides in chapters 4, 5 and 6. The first of them deals with a design of the inertial stabilization of the optical axis of the camera — the so-called line-of-sight inertial stabilization. The two remaining deal with the design of the image-based feedback control system tailored to inertially stabilized platforms.

1. Although a number of technical issues related to design of control systems for inertial stabilization have been already solved and reported in the literature as documented in the survey paper [1], little is reported on the augmentation of line-of-sight stabilization system with computer vision systems that provide automatic visual pointing

and tracking. Combination of the two loops — the inertial angular rate loop and the computer vision loop — does not seem straightforward, in particular if the inertial angular rate loops are implemented in the decoupled way. In the paper [2] published in *IEEE Transactions on Control Systems Technology* we (I and the supervisor) propose a systematic procedure based on techniques from visual servoing and give details of the solution for the popular double gimbal (Az-El) platform. In the thesis the topic of the image based tracking for inertially stabilized platforms is studied in chapter 5.

2. Another problem where a contribution is claimed in this thesis is a (partial) compensation of the influence of the time delay introduced by real-time video processing in the outer feedback loop. An intuitive solution is proposed first which relies on integration of the fast sampled measurements of the inertial angular rate provided by the inner feedback loop. This simple compensation scheme is then rigorously shown to be just a special instance of a few more systematic approaches based on reset control/estimation, modified Smith's predictor and multirate estimation. It turned out that investment into this rigorous analysis paid off in the end because a non-intuitive solution was devised that surpassed the performance of the initial intuitive one. The topic is rigorously studied in chapter 6 of this thesis.

List of the author's publications related to the thesis

- Preliminary results on combination of inertial stabilization and computer vision were given at the *2009 IEEE CDC* conference [3] and the *2010 SPIE Defense, Security and Sensing* conference [4]. The journal version [2] took advantage of the feedback obtained from the reviewers and the audience at these two events. In addition, it also documents laboratory experiments.

- Preliminary results obtained on delay compensation in visual servoing with an inner inertial rate loop invoking the concepts of the modified Smith's predictor and reset observer were presented at *2010 IEEE CDC* conference [5]. Significantly extended version of this conference paper was submitted into *IFAC Control Engineering Practice* journal [6] in January 2013. This extended paper includes a rigorous proof of stability of some compensation schemes as well as a documentation of laboratory experiments with a real inertially stabilized airborne camera platform.

- Computational design of a feedback controller for one specific configuration with multiple gimbals — the so-called dual-stage configuration to be presented in section 2.4 — calls for a design of MIMO controller with a specific structure. Namely, some terms of the transfer function matrix are requested to be zero. The paper [7] presented at *2011 IEEE World Congress* offers a case study for the popular HIFOO computational package for Matlab. The extended version of this paper has been accepted for publication in *IFAC Mechatronics* journal [8]. The major improvement of the paper consisted in bringing in another solver for comparison — Hinfstruct solver from Robust control toolbox for Matlab.

- Although the primary task for the inertial stabilization loop is to attenuate the influence of the rotational motion of the carrier, the flight tests with developed platforms mounted underneath a helicopter revealed that unless the camera gimbal is perfectly statically balanced, it is also the vibrations of the carrier (a helicopter) that can create a significant disturbing torque that is disrupting the line of sight. A powerful, albeit somewhat underappreciated engineering solution is to employ a disturbance feedforward. Acceleration of the carrier, which is proportional to the disturbing torque, is measured and a compensating signal is forwarded to the motor. Major improvements in the control performance were reported at *2011 IEEE MSC* conference [9].

- The last publication relevant for this thesis is yet another *2010 SPIE Defense, Security and Sensing* conference paper [10] that describes an implementation of the attitude estimation based on extended Kalman filtering. Although the methodology behind the paper was standard, the inclusion of practical experiments certainly adds some value to this report.

1.5 SOME OTHER MEANS OF PRESENTATION

The results achieved by the whole project consortium in the domain of the inertial stabilization were presented not only to the academic community by means of research papers and to the industrial community by means of participation at fair trades (International Engineering Fair (MSV) in Brno, International Fair of Defence and Security Technology (IDET) in Brno), but it was also presented to a wide public in one episode of the popular Czech TV show called PORT broadcasted by Czech television (ČT1 channel). The episode was focused on inertial stabilization of cameras. The URL link is listed in the appendix.

1.6 OUTLINE OF THE THESIS

Before the key contribution of this thesis is presented, some introductory material is included. Namely, in the immediately following Chapter 2, an overview of a few most common mechanical configurations suitable for line-of-sight stabilization is presented. The chapter contains a description of the classical double-gimbal configuration followed by a description of more complex configurations such as the dual-stage configuration. Mathematical models of presented configurations are always introduced including values of physical parameters. The chapter 3 then enumerates some basic concepts from the general area of motion control, including a discussion of the popular cascade control structure. The main contribution of this thesis can then be found in chapters 4 – 7.

Chapter 4 presents the analysis and design of the line-of-sight stabilization for various mechanical configurations. The emphasis is put on analysis of capability of a control scheme to attenuate the disturbing motion of the carrier.

Chapter 5 augments the standard line-of-sight stabilization scheme based on decoupled inertial angular rate loops by incorporating the image-based pointing and tracking capabilities. The content of this chapter is mainly based on the journal paper [2] but the presentation is generalized so that the ideas are valid for a general mechanical configuration. The chapter also contains experimental laboratory results and some links to videos from real helicopter flight tests.

Chapter 6 deals with control design issues arising due to a one-step delay in slow-sampled image tracking loop. Several delay compensation schemes are proposed and compared in this chapter.

Finally the chapter 7 presents an implementation of the attitude estimating unit. It is based on the conference paper [10] but here another estimation concept is described. Furthermore, a noncausal second derivative of the GPS signal is used to compensate for disturbing accelerations.

Chapter 2

Mechanical configurations

This chapter brings an overview of various mechanical configurations for inertial stabilization including the corresponding mathematical models of kinematics and dynamics and values of physical parameters. The chapter starts with an introduction to the two main approaches to inertial stabilization — the so-called *mass stabilization* of the camera and the *steering mirrors stabilization*. This thesis subsequently deals only with the mass stabilization concept, though.

2.1 MAIN PRINCIPLES

2.1.1 Mass stabilization of the camera

The term *mass stabilization* refers to the easily observed behavior of rigid bodies that tend to keep their orientation in the 3D (inertial) space even if they are exposed to external torques. In particular, the line of sight of a camera supported by a few gimbals tends to be stationary when the base (the carrier) is subject to rotational motion even when no actuators are used to drive the gimbals. This assumes the friction in gimbals is negligible. In fact, if the only task for the control system is the task of inertial stabilization, no control system will be needed, just a sufficient number of frictionless gimbals and a good static balance of the payload. Surprisingly, the heavier the payload (in fact, the moment of inertia needs to be considered), the better. However, mere inertial stabilization is hardly all that is required in practical applications. The line of sight of the payload is typically commanded to (re)point towards a new object and/or as the object on the ground moves, the line of sight is expected to track the object. That is why the motorized gimbals are needed. From a practical viewpoint they are also needed to accomplish the pure task of inertial stabilization because some extra torque needs to be applied to overcome the residual friction in gimbals. The commands for these motors are based on the measurements of the inertial angular rates (using MEMS or laser gyros). Needles to emphasize that for these (re)pointing and tracking tasks, the heavy payload (actually, the large moment of inertia) is a disadvantage.

In the following text, several mechanical configurations will be introduced that are based on the mass stabilization principle. These are

- Classical *double gimbal* with two degrees of freedom (the azimuth and elevation gimbals),
- *Tip-tilt* configuration with two degrees of freedom,
- More degrees of freedom configuration.

Double gimbal (Elevation-Azimuth)

Double gimbal configuration represents the most common implementation of the mass stabilization concept. All platforms mentioned in Fig. 1.2 are of the double-gimbal type (at least from the outside). The sketch of the double gimbal configuration is in Fig. 2.1. Notice that, by rotating the elevation gimbal such a way that the camera is pointing towards ground (or upwards), double gimbal reaches a configuration that is in robotics usually denoted as *singularity point*. At this point, the azimuth gimbal is no longer able to manipulate by the line-of-sight and double-gimbal operates at this point only as a manipulator with one DOF (degree of freedom).

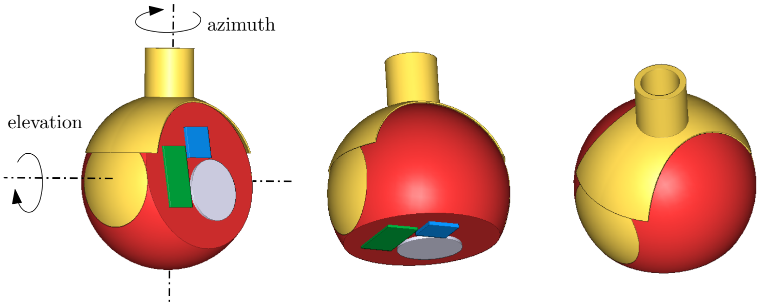


Figure 2.1: Sketch of the Elevation-Azimuth configuration. Platform allows a rotation around the vertical axis by the azimuth gimbal, and around the horizontal axis by the elevation gimbal. Blue, green and grey element represent the optical devices – a payload.

Presence of the singularity when pointing the camera towards the ground disadvantages double gimbal from using in the aircraft when there is a need to observe ground objects directly under the aircraft. This serious issue may be solved be placing double-gimbal platform vertically, so that the singularity point moves towards the horizon as demonstrated in Fig. 2.2.

As a conclusion the main advantage of the double-gimbal configuration is its universality, mechanical simplicity and major field of view that is available for optical devices.

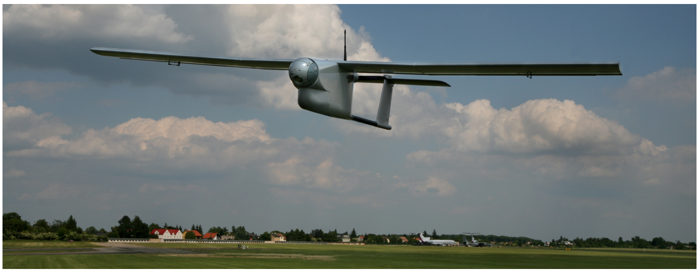


Figure 2.2: A double-gimbal platform mounted vertically (in the UAV nose) to allow stabilized observing of the target that is moving on the ground under the UAV. Singular point is thus moved to other location — horizon in this case.

Tip-Tilt

The main disadvantage of double-gimbal configuration — presence of the singularity point when observing target under the aircraft, may be eliminated by introducing the so-called Tip-tilt configuration. This configuration contains no singularity when observing target under the aircraft. Instead it is shifted to the situation when the horizon is being observed. The Tip-tilt configuration provides a lower field of view that is available for optical devices. This is due to another supporting arm in the outer gimbal (represented by the green color in Fig. 2.3), that is obstructing the view.

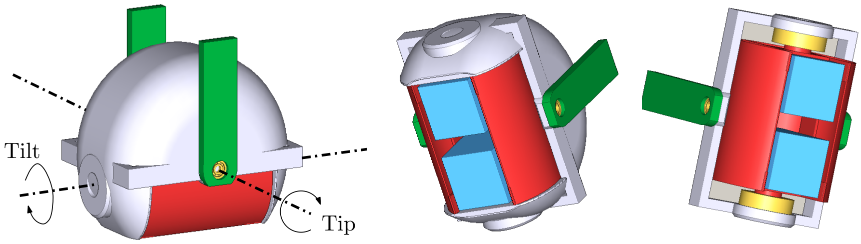


Figure 2.3: Scheme of the Tip-Tilt configuration. Two cameras denoted by blue color may be tilted around two axes — Tip and Tilt.

More degrees of freedom

Combining advantages of the elevation-azimuth (major field of view) and tip-tilt (missing singularity when tilting the camera towards the ground) together, a configuration with more degrees of freedom may be invented. From the outside it appears to be exactly the same as the double-gimbal and in the inside the camera may be rotated by another (one or two) tip-tilt type gimbals to prevent the singularity. The sketch of such system with only a one additional gimbal is in Fig. 2.4. While the two outer gimbals are usually built as self-locking, the inner gimbals are gearless in order to provide the mass stabilization.

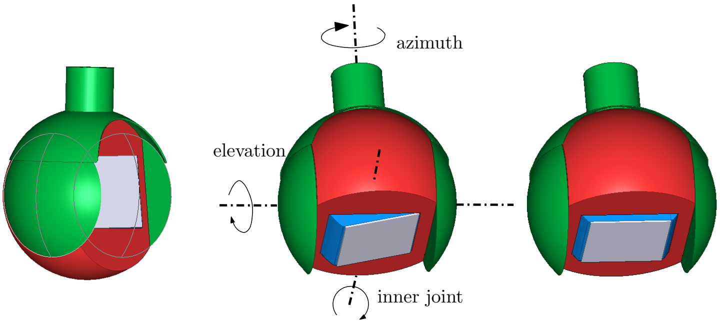


Figure 2.4: More degrees of freedom configuration — three in this case. The classical elevation-azimuth is supplemented by another (inner) gimbal allowing the fine limited angle movement.

The most of superior stabilized platforms are developed with five gimbals. The two outer in the azimuth-elevation configuration are providing the pointing feature; the two inner in the tip-tilt configuration are providing the stabilization feature. The last inner gimbal is providing a rotation around the optical axis. This fifth gimbal allows to prevent the image rotation around the screen center (see the example video of the image rotation around the image center in attachments section in the end of this thesis — video `H240-LOS.avi`).

The main reason to implement the mechanically very demanding fifth gimbal is though not an elimination of the image rotation itself (cheaper platforms can reduce the image rotation by the video post processing), instead it is the fact that any moving or rotating image is blurred. Stabilizing the rotation around the optical axis specially at higher frequencies helps to reduce image blur.

2.1.2 Fast steering mirrors

When the payload to be stabilized is large and heavy, a gimballed mirror may be placed into the optical path to stabilize the line-of-sight instead of mass stabilizing the entire payload. Using this approach leads to reducing the size and weight of the system. However, the line-of-sight kinematics of mirrored gimbal can be quite complex.

A typical two-axis mirror-stabilization configuration is in Fig. 2.5 on the left. The description of this configuration is done here using [1]. The motion of the line of sight about the Z axis is one to one with the motion of the outer gimbal, and thus is mass stabilized. But the motion about the Y or inner axis is more complex. Because of the reflective properties of mirrors, line-of-sight angle θ_{LOS} around the Y axis is related to the mirror rotation θ_M

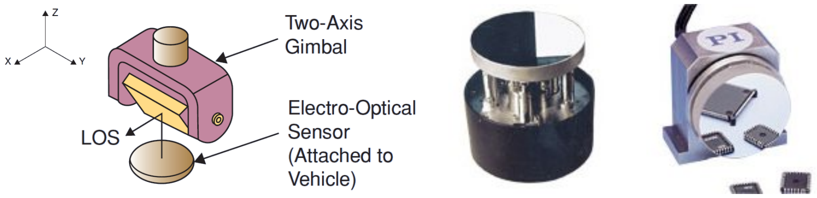


Figure 2.5: Left: The sketch of the fast steering mirror based stabilization principle. Image is adopted from [1].
Middle and right: Fast steering mirrors by Physik Instrumente (PI). Pictures are adopted from PI's website `http://www.physikinstrumente.com`.

and base rotation θ_B by

$$\theta_{LOS} = 2\theta_M - \theta_B. \tag{2.1}$$

Equation (2.1) has several implications that greatly complicate the design. The mirror must move with respect to inertial space to hold the line-of-sight stationary and thus is not mass stabilized. A feedback gyro cannot be attached directly to the mirror since the it has to move two times faster to reject the base motion.

In spite of these facts, the advantage of this approach is that the bandwidth of the pointing is generally much higher/faster compared to the mass stabilization approach. Typically this can be made higher than hundreds of Hertz (see e.g. the systems in Fig. 2.5).

The main weakness is attenuating the influence of the disturbing motion of the base that is beyond the pointing bandwidth. Whereas the mass stabilization based configurations include this capability inherently, in mirror stabilization it turns out that there is no way to attenuate such disturbances at all.

2.2 DOUBLE GIMBAL PLATFORM (AZ-EL)

A successful design of the high performance inertial stabilization and pointing and tracking system must be based on a mathematical model of dynamics of the gimbal system. A complete analysis of the double gimbal was given as early as in 1969 by A.K. Rue [11]. The model was very general, hence simplification to the more practical case of a payload rotating about its principal axis was given by the same author in [12].

In order to develop such a model a two common methodologies can be used: Lagrangian energy-based methodology and Newton-Euler vector-based methodology. Description of the two is given in every other robotics or multibody dynamic systems textbook such as [13].

Lagrangian approach is very versatile and is used to find the dynamics of a double gimbal system in [14] under the assumption that the base (the carrier) remains still in the inertial space. Following the Lagrangian approach in the practically more appealing case of a moving base seems to be approachable by introduction of three artificial/virtual joints. Unfortunately, the resulting equations contain these artificial Euler angles describing the orientation of the base.

The Newton-Euler approach seems more suitable for this situation. In the forward recursion, the inertial angular rates of the relevant frames are computed, started with the base. Having the angular rates, differentiation of these (respecting the rules for differentiation in rotating frames) gives inertial accelerations. In the backward recursion, torques applied at the joints are computed using the knowledge of the just computed velocities and accelerations.

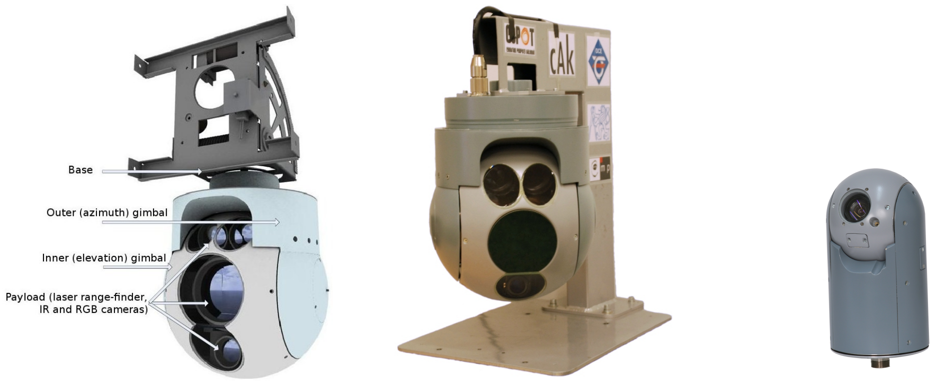
A work [15] provides a development of equations following the both approaches. In this thesis Newton-Euler approach is used to develop the model.

2.2.1 Benchmark systems

The *double gimbal system* is the classical configuration with two degrees of freedom. The inner gimbal allows elevation of the payload, the outer gimbal allows a change in heading (or azimuth) angle. With this type of configuration in total two platforms were developed within a project coordinated by Czech Air Force and Air Defence Technological Institute (Vojenský ústav letectva a PVO) in collaboration with Czech Technical University in Prague and ESSA company, see Fig. 2.6.

The first, shown on the left and middle in Fig. 2.6, is the prototype with the acronym "H240" (240 is the diameter of the platform in millimeters). The payload consists of a regular RGB camera, infrared camera and laser range-finder. Direct drive motors are used for the two axes and MEMS based gyros (inertial rate sensors) are attached to the payload.

The second, shown on the right in Fig. 2.6, represents the lighter version of the platform designated by the acronym "S120" (with the diameter 120 mm) which is employed with only single RGB camera. Instead of expensive direct drive motors, here only classical brush type motors with planetary gearbox were used.



(a) 3D visualization of the platform H240: both the azimuth and the elevation angles can rotate $n \times 360°$.

(b) Photo of the benchmark system H240. The supporting structure is only used in a lab.

(c) Photo of the light version of the platform (S120) equipped by classical motors instead of direct drives.

Figure 2.6: Both platforms were developed by Czech Air Force and Air Defense Technological Institute in collaboration with Czech Technical University in Prague and Essa company.

2.2.2 Assumptions

1. Although big efforts were invested into designing and building the gimbal system so that it is statically balanced and the motors do not have to counteract the gravity, in reality a perfect static balance can hardly be obtained. Nonetheless, the perfect balancing is assumed in the subsequent analysis, which makes the principle of mass stabilization usable here. The impact of static unbalance can be handled separately using acceleration feedforward as described in a later chapter.

2. Another threat for validity of a purely rotational model is the nonzero distance between the inner gimbal elevation axis and the longitudinal axis of the aircraft. Consequently, the roll motion of the aircraft will induce some translational motion of the cameras. However, in the intended applications, observed objects are located at a distance of at least a few hundred meters, which makes this translational motion negligible.

3. Another assumption is that of a symmetry of the system: it is assumed that the payload rotates around its principal inertial axis, that is, one of them in aligned with the optical axis, the other with the elevation axis. The matrix of inertia is then assumed in diagonal form only. Absence of cross-products of inertia makes the model simpler too. Validity of these assumptions is confirmed from the automatic calculation of moments of inertia in the CAD system used by the mechanical engineer.

2.2.3 Notation for coordinate frames and their rotations

Mathematical model relies on expressing rotations of coordinate frames with respect to some other coordinate frames. The right-handed orthogonal coordinate frames are represented by triads of vectors $\{x, y, z\}$ and for simplicity they all assume a common origin; this is certainly justifiable when far-away objects are tracked. The coordinate frames and their symbols used in subscripts and superscripts are: the reference coordinate frame [R] aligned with the ground but translated to the center of gravity of the carrier, its z_R axis oriented towards the ground as is common in aerospace applications; the base coordinate frame fixed to the body of the carrier [B] with its x_B axis heading forward and y_B to the starboard; the coordinate frame attached to the outer (azimuth) gimbal [A], which can rotate with respect to the carrier around the $z_B = z_A$ axis; the coordinate frame attached to the inner (elevation) gimbal [E], which can rotate with respect to the azimuth gimbal around the $y_A = y_E$ axis; and finally the coordinate frame attached to the camera [C]. Rotation of [C] with respect to [E] is fixed and is used just for the "esthetic" purpose of (re)denoting the camera optical axis as the z_C axis.

$$\begin{tikzcd}[B] \arrow[r, "z_B", "\psi"'] & [A] \arrow[r, "y_A", "\theta"'] & [E]\end{tikzcd}$$

Figure 2.7: Schematic diagram of joints rotation composition.

The sequence of the two key rotations expressing the pose of the inner gimbal (fixed to camera) with respect to the base (carrier) is visualized in Fig. 2.8, schematically in Fig. 2.7 and for completeness it is given by

$$R_A^B = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

and

$$R_E^A = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}, \tag{2.3}$$

where the lower and upper indices are used here as "rotation matrix expressing the coordinate triad of the A frame within the B frame". Applying the right-hand rule, the (outer) azimuth gimbal rotates to right for the positive angle ψ and the (inner) elevation gimbal rotates up for a positive increment in the θ angle. Using the common shorthand notation like $c_\psi = \cos\psi$, the composition of the two rotations is given by the matrix product

$$R_E^B = \begin{bmatrix} c_\psi c_\theta & -s_\psi & -c_\psi s_\theta \\ s_\psi c_\theta & c_\psi & -s_\psi s_\theta \\ -s_\theta & 0 & -c_\theta \end{bmatrix}. \tag{2.4}$$

When specifying angular rates, the subscript/superscript scheme used here follows the common style, defined for instance in [13]: one needs to tell which coordinate frame is rotating with respect to which other coordinate frame, and in which coordinate frame is such a vector expressed. For example, $\omega_{A,E}^R$ stands for the angular rate of the $\underline{\text{E}}$levation

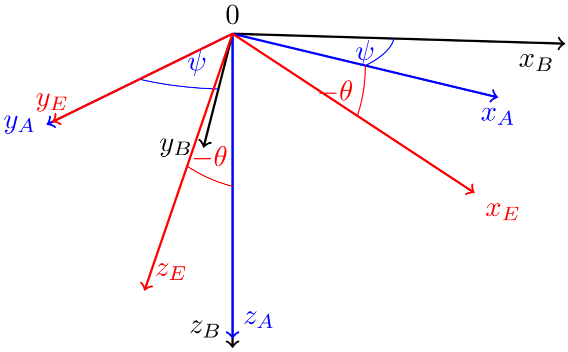


Figure 2.8: Composition of rotation of coordinate frames attached to the base, outer (azimuth) gimbal and inner (elevation) gimbal.

gimbal with respect to the Azimuth gimbal, expressed in the Reference frame. Oftentimes, the notation is relaxed in order to avoid cluttering formulas with indices. For instance, ω_A is a short notation for $\omega_{R,A}^A$, that is, the inertial angular rate of the A gimbal. Its z component is then ω_{Az}.

The transposition of the above matrices give the transformation in the opposite directions, that is, $R_B^A = (R_A^B)^T$ can be used to transform the vector from the base frame to the outer gimbal frame (elevation) and so on.

2.2.4 Forward recursion of Newton-Euler method, determining velocities and accelerations

Inertial angular rates

For the purpose of inertial stabilization and pointing, only rotation of frames is considered, not translation. Coordinate frame can rotate with respect to another frame, but in this application it is of interest to study inertial rotations mostly, that is, rotations of frames with respect to the inertial coordinate frame (represented here by the reference frame subscripted with the letter R). As every other vector, the vector of angular rate can be expressed in an arbitrary frame. In this application three inertial angular rates (vectors) are necessary to be considered:

1. The inertial angular velocity of the base frame is described by the vector ω_B^B (or $\omega_{R,B}^B$ to emphasize that the rotation is with respect to the inertial frame $o_R x_r y_R z_R$, or ω_B for short), where the symbols used for the components are standard in aerospace (p, q and r for roll, pitch and yaw rates, respectively):

$$\omega_B^B = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.5}$$

2. Inertial angular velocity of the outer frame is described by the vector ω_A^A

$$\omega_A^A = \begin{bmatrix} \omega_{Ax} \\ \omega_{Ay} \\ \omega_{Az} \end{bmatrix} \tag{2.6}$$

3. Inertial angular velocity of the inner frame, to which the camera is attached, expressed in the same frame, is described by the vector ω_E^E

$$\omega_E^E = \begin{bmatrix} \omega_{Ex} \\ \omega_{Ey} \\ \omega_{Ez} \end{bmatrix} \tag{2.7}$$

As a part of the forward step in Newton-Euler method for finding model of dynamics, these angular velocities must be determined. A crucial point here is that because direct drives motor essentially decouple the two frames in their rotation along one axis (but not about the two other).

In general, there can be several contributions to the vector of inertial angular rate. These can be added but need always be expressed in the same coordinate systems. The aim is to express all the angular rates in the body frames. To start with, the inertial angular rate of the outer gimbal whose relative rotation to the base is described by the angle ψ is

$$\omega_A^B = z_B\dot{\psi} + \omega_B^B = \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} p \\ q \\ \dot{\psi} + r \end{bmatrix}. \tag{2.8}$$

But note that a proper expression for the inertial angular rate of the azimuth gimbal (as a function of the inertial angular rate of the base) in the case of direct drive joints should read as

$$\omega_A^B = \omega_{Az} z_B + B_z \omega_B^B = \begin{bmatrix} 0 \\ 0 \\ \omega_{Az} \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} p \\ q \\ \omega_{Az} \end{bmatrix}, \tag{2.9}$$

where the scalar variable ω_{Az} describes inertial rotation around the common z axis of the base and the outer gimbal and the role of the matrix B_z is to block the influence of the inertial rotation rate of the outer gimbal ($z_B = z_A$) by the rotation rate of the base. That this is indeed the case can be seen by considering rotation of the base around its vertical axis, that is, $r \neq 0$. The equation (2.8) would suggest that $\omega_{Az} \neq 0$, but unless friction of the brushes or ball bearings or external disturbing torques are considered, the second law of Newton states that $\omega_{Az} = 0$. This is the very principle of mass stabilization. The value of ω_{Az} is totally unrelated to r.

Expressing this inertial angular rate of the angular gimbal in its own frame gives

$$\omega_A^A = R_B^A z_B \omega_{Az} + R_B^A B_z \omega_B^B = \begin{bmatrix} pc_\psi + qs_\psi \\ qc_\psi - ps_\psi \\ \omega_{Az} \end{bmatrix}. \tag{2.10}$$

The relative angular rate $\dot{\psi}$ is then found as a difference between the z components of the vectors of inertial angular rate of the outer gimbal and the base. Even though computationally it makes no difference, at which coordinate frame these are being subtracted, to be formally correct, let's choose one, say the outer gimbal frame:

$$\dot{\psi} = [\omega_A^A - \omega_B^A]_z, \tag{2.11}$$

which can be easily seen (also from 2.8) to be

$$\dot{\psi} = \omega_{Az} - r. \tag{2.12}$$

Proceeding one gimbal further, the inertial angular rate of the inner (elevation) frame, which is only partially being stirred by the outer gimbal, is

$$\omega_E^A = B_y \omega_A^A + y_A \omega_{Ey} = \begin{bmatrix} pc_\psi + qs_\psi \\ \omega_{Ey} \\ \omega_{Az} \end{bmatrix}, \tag{2.13}$$

where again the matrix B_y blocks the rotation about the y_A axis while this is described by the independent variable ω_{Ey}. Expressing this vector in the coordinate frame attached to the inner gimbal we get

$$\omega_E^E = R_A^E \omega_E^A = \begin{bmatrix} c_\theta(pc_\psi + qs_\psi) - \omega_{Az}s_\theta \\ \omega_{Ey} \\ s_\theta(pc_\psi + qs_\psi) + \omega_{Az}c_\theta \end{bmatrix}. \tag{2.14}$$

The relative elevation angle θ is given as the y component of a difference between the rates of the inner and outer gimbals

$$\dot{\theta} = [\omega_E^E - \omega_A^E]_y, \tag{2.15}$$

where the inertial angular rate of the outer gimbal can be expressed in the inner gimbal frame as

$$\omega_A^E = R_A^E \omega_A^A = \begin{bmatrix} c_\theta(pc_\psi + qs_\psi) - \omega_{Az}s_\theta \\ qc_\psi - ps_\psi \\ s_\theta(pc_\psi + qs_\psi) + \omega_{Az}c_\theta \end{bmatrix}. \tag{2.16}$$

Hence

$$\dot{\theta} = \omega_{Ey} + ps_\psi - qc_\psi. \tag{2.17}$$

Angular accelerations

Determining angular accelerations of the two moving gimbals consists in differentiating the expressions (2.10) and (2.14) in their respective rotating coordinate frames. A standard result on differentiation of vectors expressed within rotating coordinate frames must be invoked. Determining the angular acceleration of the base frame is trivial

$$\alpha_B^B = \dot{\omega}_B^B + \omega_B^B \times \omega_B^B = \dot{\omega}_B^B. \tag{2.18}$$

The inertial angular velocity of the outer gimbal (2.10) is differentiated in its own coordinate frame using the same rule, however, here the cross product yields a nonzero result

$$\begin{aligned}
\alpha_A^A &= R_B^A \alpha_A^B = R_B^A B_z \dot{\omega}_B^B + R_B^A z_B \dot{\omega}_{Az} + R_B^A \left(\omega_B^B \times (B_z \omega_B^B + z_B \omega_{Az})\right) \\
&= R_B^A B_z \dot{\omega}_B^B + z_A \dot{\omega}_{Az} + \left(\omega_B^A \times (R_B^A B_z \omega_B^B + z_A \omega_{Az})\right) \\
&= \begin{bmatrix} c_\psi(\dot{p} + \omega_{Az} q - qr) + s_\psi(\dot{q} - \omega_{Az} p + pr) \\ c_\psi(\dot{q} + \omega_{Az} p - pr) - s_\psi(\dot{p} - \omega_{Az} q + qr) \\ \dot{\omega}_{Az} \end{bmatrix}.
\end{aligned} \tag{2.19}$$

Following the same principle, the angular acceleration of the inner gimbal (expressed in the inner gimbal frame) can be obtained

$$\begin{aligned}
\alpha_E^E &= R_A^E \alpha_E^A = R_A^E B_y \dot{\omega}_A^A + R_A^E y_E \dot{\omega}_{Ey} + R_A^E \left(\omega_A^A \times (B_y \omega_A^A + y_A \omega_{Ey})\right) \\
&= R_A^E B_y \dot{\omega}_A^A + y_E \dot{\omega}_{Ey} + \omega_A^E \times (R_A^E B_y \omega_A^A + y_E \omega_{Ey}).
\end{aligned} \tag{2.20}$$

The full resulting vector is not included here; the vector-matrix computation can easily been reproduced using a symbolic computation package of choice.

2.2.5 Backward recursion of Newton-Euler method, determining torques

The total torque exerted on the inner elevation gimbal is equal to the derivative of the angular momentum of the gimbal. Following once again the rules for differentiation of vectors in rotating frames

$$T_E^E = I_E \alpha_E^E + \omega_E^E \times I_E \omega_E^E. \tag{2.21}$$

The calculation can be easily done with some symbolic computation package, but some work can be saved by focusing on the behavior around the y_E only, because it is around that axis that the torque is induced exclusively by the motor and the friction in the bearings and brushes. The y component of the torque is

$$T_{Ey} = I_{Ey} \dot{\omega}_{Ey} - (I_{Ex} - I_{Ez})\Big(\omega_{Az} c_\theta + s_\theta(p c_\psi + q s_\psi)\Big)\Big(\omega_{Az} s_\theta - c_\theta(p c_\psi + q s_\psi)\Big). \tag{2.22}$$

The torque exerted on the outer azimuth gimbal is computed similarly but includes an extra term representing the reaction torque from the elevation gimbal (of course expressed correctly in the outer gimbal frame)

$$T_A^A = R_E^A T_E^E + I_A \alpha_A^A + \omega_A^A \times I_A \omega_A^A. \tag{2.23}$$

Again, it is only the z component T_{Az} that is driven by the motor. Full equation for T_{Az} is not listed here due to the length of the formula.

2.2.6 Nonlinear differential equations for dynamics of Az-El system

Structuring the total torque around the y_E (elevation) axis as the motor induced torque and the friction induced torque, the full differential equation describing the motion of the inner gimbal around the elevation axis is

$$I_{Ey}\dot{\omega}_{Ey} = T_{Em} - T_{Ef}(\dot{\theta}) - T_{Ed} + (I_{Ex} - I_{Ez})\Big(\omega_{Az}c_\theta + s_\theta(pc_\psi + qs_\psi)\Big)\Big(\omega_{Az}s_\theta - c_\theta(pc_\psi + qs_\psi)\Big). \tag{2.24}$$

Performing the same step for the outer gimbal, the ultimate differential equation describing the motion of the outer gimbal around the azimuth axis is

$$I_{Az}\dot{\omega}_{Az} = T_{Am} - T_{Af}(\dot{\psi}) - T_{Ad} + ...^1 \tag{2.25}$$

To get an essence of the dynamic equations just derived, assume the base rests still in inertial space, that is, $p = 0$, $q = 0$ and $r = 0$. The nonlinear differential equations describing the dynamics of the two-gimbal platform are

$$\left(I_{Az} + I_{Ex}s_\theta^2 + I_{Ez}c_\theta^2\right)\dot{\omega}_{Az} = T_{Am} - T_{Af} - T_{Ad} - 2(I_{Ex} - I_{Ez})c_\theta s_\theta \omega_{Az}\omega_{Ey} \tag{2.26}$$

and

$$I_{Ey}\dot{\omega}_{Ey} = T_{Em} - T_{Ef} - T_{Ed} + (I_{Ex} - I_{Ez})s_\theta c_\theta \omega_{Az}^2. \tag{2.27}$$

This agrees with the available results for this standard configuration, for instance those presented in [14]. The structure of these equations has a physical interpretation: the equation for the outer gimbal includes a Coriolis term (product of the two inertial angular rates), which couples the rotation around the elevation axis to the rotation around the azimuth axis. The equation for the inner gimbal then includes a centripetal term that makes the elevation easier when approaching the horizontal orientation.

In the case when $I_{Ex} = I_{Ez}$, the equations simplify considerably to

$$(I_{Az} + I_{Ex})\dot{\omega}_{Az} = T_A \tag{2.28}$$

and

$$I_{Ey}\dot{\omega}_{Ey} = T_E. \tag{2.29}$$

Unfortunately, this simplification is not applicable to platforms from Fig. 2.6. The condition $I_{Ex} = I_{Ez}$ requires that the payload is axially symmetric with respect to the elevation axis. Therefore, coupling between the two equations through the Coriolis and the centrifugal terms will have to be considered in the model.

The two equations (2.26) and (2.27) need to be accompanied by the two equations for the positional (motor) angles ψ and θ, stated already in (2.12) an (2.17).

[1] Too lengthy to be given here. The full result may be calculated using an arbitrary software for symbolic calculations. See the simplified version for p, q, r=0 to get an insight.

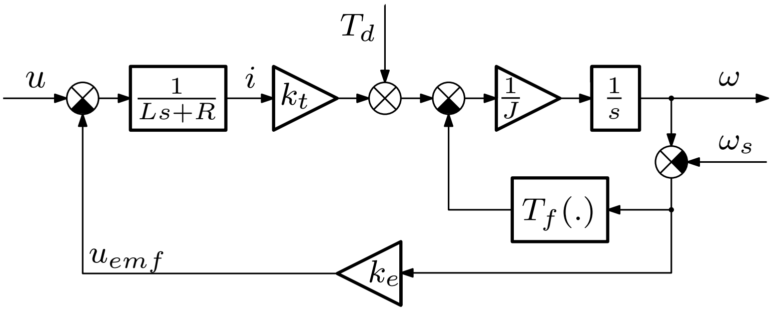


Figure 2.9: Model of the DC direct drive motor considering the rotation of the stator. The friction term T_f accounts not only for the motor friction but also for ball bearing friction.

2.2.7 Rigid body dynamics modeling

Both the elevation and the azimuth gimbals are modeled as a classical DC motor with a payload. The total inertia J in the case of elevation joint reads $J = I_{Ey}$ while in the case of azimuth joint $J = I_{Az} + I_{Ex}s_\theta^2 + I_{Ez}c_\theta^2$. The load for the azimuth axis clearly depends on the elevation θ of the camera. The standard *mechanical equation* of the DC motor attached via rigid link to load is

$$J\dot{\omega} = k_t i - T_f \underbrace{(\omega - \omega_s)}_{\text{relative rate}} - T_d, \tag{2.30}$$

where ω [rad/s] is the angular rate, i [A] is the motor current, T_f [Nm] is the disturbing friction torque and T_d is some other unmodeled disturbing torque. The remaining parameters are listed in Table 2.1. The *electrical equation* is

$$L\frac{di(t)}{dt} + Ri(t) + k_e\omega(t) = u(t), \tag{2.31}$$

where $u(t)$ is the voltage produced by the controller. Equations (2.30) and (2.31) are rather common with the exception of a slight detail that both the friction torque and the back emf voltage are induced by the relative motion of the rotor with respect to the stator, see the block diagram Fig. 2.9. As soon as stator rotates in the inertial space, the motor “feels“ this and as a result an extra torque is exerted on the rotor in the direction of the stator rotation. This presents an unwanted coupling and corrupts the mass stabilization. Even in a friction free system, the rotor would rotate eventually with the same angular velocity as the stator. This can be compensated by implementing a current feedback loop in later sections.

2.2.8 Modeling the friction

It appears in the experiments that friction in joints plays a major role in determining the dynamic response of the platform. The need of a more advanced friction modeling and compensation is pronounced here because in a standard mode of use the system works in close neighborhood of a zero velocity operating point. Simple viscous or Coulomb friction models are not much useful. The popular LuGre dynamic friction model [16] was used here because it is capable of describing the frictional phenomena around the zero velocity (the so-called Stribeck effect). A short introduction to LuGre model is made here just for convenience of readers. The dissertation theses [17, 18] and [19] may be used to get the insight to friction modeling, identification and estimation. Some more recent survey papers are [20], [21] and [22].

The LuGre model contains a single friction state $z(t)$ governed by

$$\dot{z}(t) = \omega(t) - \frac{|\omega(t)|}{g(\omega(t))} z(t), \tag{2.32}$$

where $g(\omega)$ is defined as

$$g(\omega) = \frac{1}{\sigma_0} \cdot \Big(T_c + (T_s - T_c) e^{-(\omega(t)/\omega_s)^2} \Big). \tag{2.33}$$

The friction torque $T_f(t)$ is then written as

$$T_f(t) = \sigma_0 z(t) + \sigma_1 \dot{z}(t) + \sigma_2 \omega(t), \tag{2.34}$$

where σ_0, σ_1, σ_2, ω_s, T_s and T_c are the parameters of the friction according to Table 2.1.

Steady state characteristic of the friction model is

$$T_f(\omega) = T_c \mathrm{sgn}(\omega) + (T_s - T_c) \mathrm{sgn}(\omega) e^{-\left(\frac{\omega}{\omega_s}\right)^2} + \sigma_2 \omega. \tag{2.35}$$

2.2.9 Experimental identification results

Experimental identification of model parameters for platform H240

For simulation purposes, all parameters of the model of elevation gimbal described by (2.30) – (2.34) were identified and are listed in Table 2.1. Some parameters were known from the components' documentation, the remaining were identified using gray-box model fitting methods that are available in System Identification Toolbox for Matlab. The static part of the identified friction model is plotted in Fig. 2.10.

Experimental identification of closed loop frequency responses for platform H240

The bandwidth of the inertial rate loop can be experimentally demonstrated to be at least 1 Hz, see Fig. 2.11.

Experimental identification of inner gimbal inertia tensor

Under a reasonable validity of the assumptions stated in section 2.2.2, there are only four "mass parameters" that need to be determined: I_{Az}, I_{Ex}, I_{Ey} and I_{Ez}. I_{Ey} was already identified during friction identification experiment. The remaining moments may be identified using (2.26) and the following identification experiment:

Table 2.1: Parameters of the elevation gimbal (including friction)

Parameter	Value	Unit
Peak torque	$T_p = 0.42$	[N-m]
Supply voltage	$V_p = 26.0$	[V]
Motor torque constant	$k_t = 0.284$	[N-m/A]
Back e.m.f. constant	$k_e = 0.284$	[V/rad/s]
Armature resistance	$R = 16.5$	[Ohm]
Armature inductance	$L = 6.3$	[mH]
Moment of inertia	$I_{Ey} = 0.0455$	[kg-m/s^2]
Coulomb friction torque	$T_c = 0.03$	[N-m]
Stribeck friction torque	$T_s = 0.0664$	[N-m]
Stribeck velocity	$\omega_s = 0.05$	[rad/s]
Sigma 0	$\sigma_0 = 328$	[-]
Sigma 1	$\sigma_1 = 2.97$	[-]
Sigma 2 (viscous friction)	$\sigma_2 = 0.0644$	[N-m/rad/s]

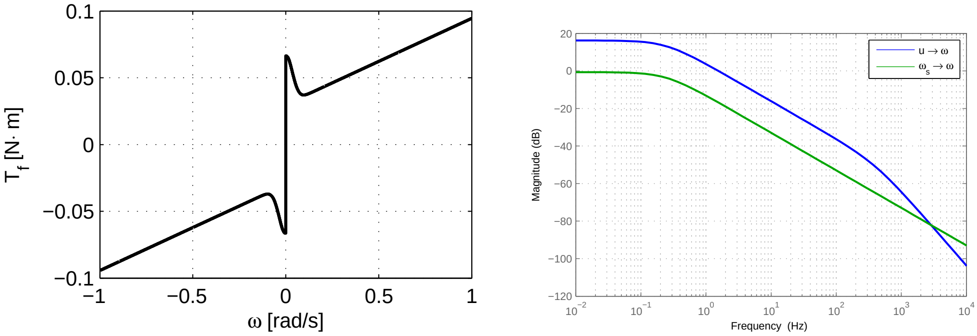
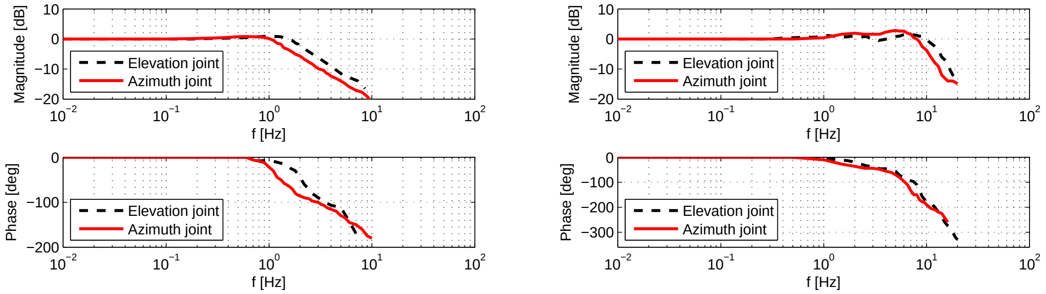


Figure 2.10: Left: Static friction characteristic — the friction torque as a function of the angular velocity. Generated using (2.35) with the parameters in Table 2.1.
Right: Bode plots of transfer functions generated from the model (2.30) – (2.34) for the elevation gimbal. Only the linear viscous part of the LuGre friction model was used (parameter σ_2).

- platform is standing still on the ground, thus $p, q, r = 0$ and $\dot{\psi} = \omega_{Az}$
- elevation angular rate $\dot{\theta} = \omega_{Ey}$ is set to zero, this way the last term in (2.26) is zero too
- for at least three different settings of θ the identification experiment is started resulting in at least three values of $J(\theta)$, with $J(\theta) = I_{Az} + I_{Ex}s_\theta^2 + I_{Ez}c_\theta^2$
- unknown parameters I_{Az}, I_{Ex} and I_{Ez} are then calculated by solving three equations with three unknowns, or in case of more θ settings by least squares method.



(a) The amplitude of the reference inertial rate was set up to 73°/s.

(b) The amplitude of the reference inertial rate was set up to 7.3°/s

Figure 2.11: Measured frequency responses for both inertial rate loops. The required inertial rate as an input and the true (measured) inertial rate as an output.

2.3 INTRODUCING DUAL STAGES

The key concept in the inertial stabilization — the so-called mass stabilization, which was mentioned in chapter 2.1.1, guarantees the significant performance of the inertial stabilization and inertial movement disturbance rejection by introducing gearless motors such as direct drives. However, gearless motors have, in comparison to the classical DC motors with gearbox, much lower maximum torque. Low peak torque disadvantages them in situations where there is a presence of disturbing effects like friction forces due to sealing rings or the wind disturbance torque in airborne applications. On the contrary, gearbox motors have thanks to gears a major torque. But they do not offer the mass stabilization concept so that the performance in inertial movement disturbance rejection is very poor. The idea behind the dual-stage stabilization is to combine advantages of both motors in one device. A powerful motor with a gearbox performs the task of the outer gimbal that is exposed to all disturbance effects, while the inner gimbal with the limited angle range takes care of the precise and high bandwidth inertial stabilization by taking the advantage of the mass stabilization.

Following sections are focused on presenting principles of dual-stage stabilization that are already known in the literature. These principles are then extended to the particular case of the inertial dual-stage stabilization. In order to demonstrate dual-stage control strategies, two benchmark systems are used (see the Fig. 2.12). Both were developed within a joint project among Czech Air Force and Air Defense Technical Institute, Czech Technical University in Prague and ESSA company.

2.3.1 Short survey of dual-stage stabilization literature

Related research results can be found under the keyword of *dual-stage control* in the area of positioning of read/write (R/W) heads in some modern hard disk drives. It is reported, for instance, by [23, 24, 25] and [26]. In order to see how much inspiration can be taken

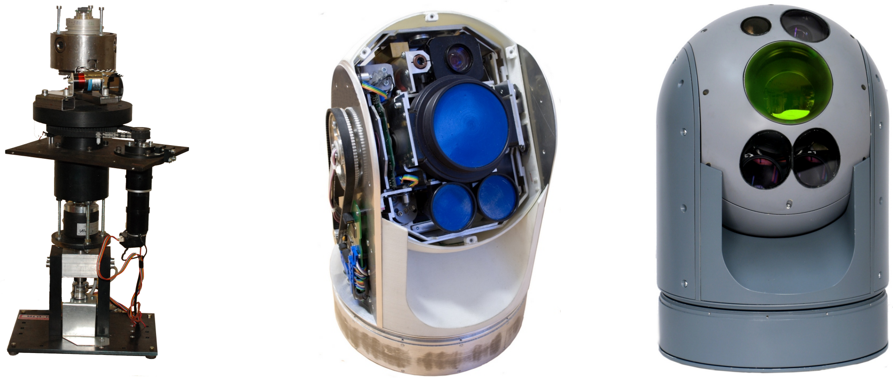


Figure 2.12: Left: Model of a single-axis dual-stage platform. Middle and Right: Four-gimbal dual-stage platform. Outer gimbals are equipped by classical motor, inner motors are linear voice coil motors.

from this area, let's describe the basic principle. A rotating arm with an R/W head at one end rotates around a joint at the other end due to a torque from a rotary voice-coil motor. This positions the R/W head onto the required track. Standard hard disks live with just this configuration. The upgrade consists in fabrication a tiny flexible joint with a very limited angle at the tip of the arm. Deflection of the tip (carrying the R/W head) is driven by contracting/extracting piezoelectric microactuators. This creates the second stage delivering the fine and fast pointing. Another application of the same principle but applied to positioning of a tip-tilt table is reported by [27] and for a rotational table by [28].

The key conceptual difference between the dual-stage control in hard-disk drives and inertial platforms is in the kinematic coupling between the stages. In hard-disk drives the second stage is *fully coupled* with the first stage — the voice-coil motor that is responsible for large angular deflection is carrying the second stage actuated by piezos that are used for a tiny but fast refinement. In inertially stabilized platforms the inner gimbal is *kinematically decoupled* from the motion of the outer gimbal, at least within a limited angular range (and disregarding the negligible friction). This decoupling is a most desirable property in inertially stabilized systems since it enables the *mass stabilization.* [1] offers a discussion of the concepts of kinematic coupling and mass stabilization.

To the best of the authors' knowledge, the use of dual-stage concept in the area of inertial line-of-sight stabilization is not documented in the literature. Perhaps the only exception is [29], which is focused more on the image-processing related issues rather than the inertial sensing based stabilization.

There is yet another application area where the issue of controlling a single variable using two actuators has been studied in a context almost identical to the one presented here. It is denoted *mid-range control* and is explained in some process control texts, for instance, by [30].

2.4 SINGLE-AXIS DUAL-STAGE PLATFORM

In this section the electromechanical benchmark system from Fig. 2.12 on the left that uses two aligned motorized gimbals to rotate a payload around a **single** axis is described. Besides orienting the payload upon command, the other key role of the control system is to isolate the inertial orientation of the payload from an unwanted disturbing rotational motion of the platform base, thus emulating the realistic scenario when the platform is carried on some mobile vehicle (car, ship or unmanned aircraft). This benchmark system was built in order to investigate the control issues arising in inertially stabilized platforms with dual joints. The sketch of the system is shown in Fig. 2.14). It consists of two electrical motors rotating the payload around a single (vertical) axis. The outer motor is the brushed rotary DC motor RE36 produced by Maxon (type 118800 with a planetary gear-head GP42C type 203123). It carries the inner stage motorized by a linear voice-coil motor (type NCC05-11-011-1X by H2W Technologies), which in turn actuates the dummy payload (in a real application this would be a camera, laser marker or range finder etc.). The inertial angular rate of the payload is measured by MEMS gyro ADIS 16255 by Analog Devices. The device can be seen in action in the short video included in attachments section in the end of the thesis (the name of the video is `dual-stage.avi`).

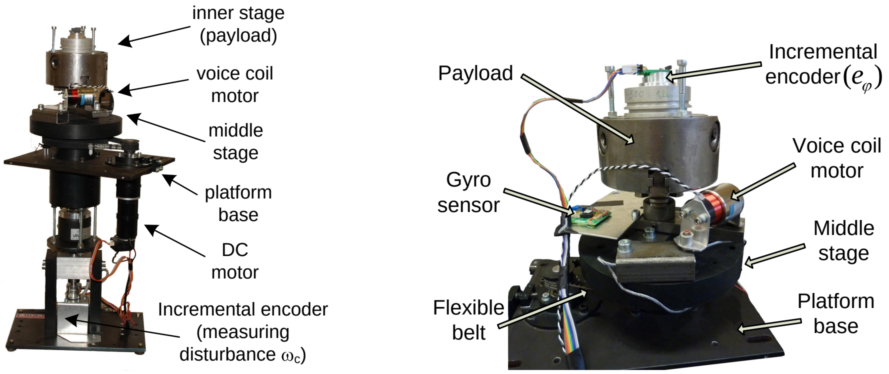


Figure 2.13: Left: Experimental system for benchmarking dual-stage inertial stabilization algorithms. The inner gimbal is actuated by a linear voice coil motor, which imposed stringent limits on the angular range ($\pm 5^o$). Right: The detail of the upper dual-stage part.

2.4.1 Notation and coordinate frames

Building a mathematical model, several coordinate frames are needed as seen in Fig. 2.14. These are: the reference frame [R] attached to the ground, the platform base frame [B] (green), the middle stage frame [M] (orange) and the inner stage frame [I] (blue). The frames [M] and [B] are connected via the DC rotary motor and a (flexible) belt gear. The

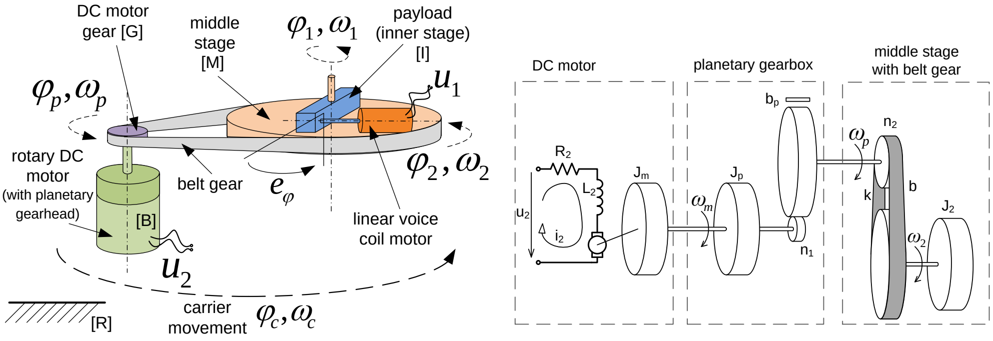


Figure 2.14: Left: Mechanical scheme of dual-stage system. Angles and rates denoted by dashed arrows stand for inertial quantities (with respect to reference frame [R]).
Right: Detail of the DC motor with a planetary gearbox and a belt gear.

frame [G] is attached to the output shaft of the DC motor. The frames [M] and [I] are then connected through the linear voice coil motor, which exerts a torque. The inertial angular rate of the shaft of the DC motor is denoted ω_p, the inertial angular rate of the middle stage, which is dynamically connected to the DC motor via a belt gear and which hosts the "stator" of the voice coil motor, is denoted ω_2. The inertial angular rate of the payload as measured by the attached MEMS gyro and actuated by the VCM is denoted ω_1. The remaining variables are the inertial angular rate of the carrier denoted ω_c, which in this particular application is measured through a measurement of the angle φ_c using an incremental encoder. Another angular measurement is the misalignment e_φ between the inner stage and the middle stage. Whereas the rotary DC motor can rotate n times 360°, the inner stage misalignment angle e_φ is limited to $\pm 5° \approx \pm 0.087\,\mathrm{rad}$.

2.4.2 Model of dynamics

This section develops a model of dynamics of the system. The list of all components and their parameter values is in Table 2.2.

The structure of the DC motor with a gearbox and an external flexible belt gear is depicted in Fig. 2.14. The structure is modeled as a classical DC motor with an electrical and mechanical parts and with a viscous friction b_p in the bearings. The belt gear is modeled as a double-mass-spring system with the spring constant k and damping b. Most of the parameters listed in Table 2.2 are known from the motor's datasheet, the remaining — k, b, b_p and J_2 — were identified using gray-box model fitting methods available in System Identification Toolbox for Matlab. The model is represented by (2.38) through (2.42).

The voice coil motor, described by (2.36) and (2.37), is modeled as a classical DC motor as well. But it is a linear motor, thus the applied force needs to be transformed into a torque. The projection of the force into a torque depends nonlinearly on the angle e_φ.

Table 2.2: Technical parameters of components.

Par.	Description	Value	Unit
x_s	VCM stroke length	12.7	mm
F_c	VCM continuous force	4.9	N
F_p	VCM peak force	14.7	N
k_m	VCM motor constant	2.95	N/$\sqrt{\mathrm{W}}$
R_1	VCM resistance	2.9	Ω
L_1	VCM inductance	0.99	mH
k_{t1}	VCM force constant	5.11	N/A
k_{e1}	VCM back emf constant	5.11	V/(m/s)
m_{t1}	VCM total mass	144	g
m_{m1}	VCM moving part mass	27	g
b_1	VCM viscous friction	0.125	Nm/(rad/s)
$U_{1_{max}}$	VCM supply voltage	7	V
r	radius of action of VCM	65	mm
J_1	payload moment of inertia	10	gm^2
k_{t2}	rot. motor torque constant	56.6	mNm/A
k_{e2}	rot. motor back emf	169	rpm/V
R_2	rot. motor resistance	2.74	Ω
L_2	rot. motor inductance	0.487	mH
$U_{2_{max}}$	rot. motor supply voltage	32	V
J_m	rot. motor mmnt. of inertia	67.8	gcm^2
b_p	rot.m. viscous friction	0.48	Nm/(rad/s)
n_1	planetary gear-head ratio	74:1	–
J_p	planetary gear-head inertia	15	gcm^2
T_m	max torque at gear output	15	Nm
b_m	avg. backlash at gear output	0.5	^{o}
J_2	mid. stage mmnt. of inertia	25	gm^2
n_2	belt gear ratio	4:1	-
k	spring constant for the belt	250	Nm/rad
b	damping of the belt	0.9	Nm/(rad/s)
f_{bw}	ADIS 16255 analog bandwidth	50	Hz
ω_{max}	ADIS 16255 maximum angular rate	5.59	rad/s

Nonetheless, this effect is negligible for the prescribed bounds on the misalignment e_φ.

On the other hand, the friction in the inner stage bearings cannot be neglected since it is significant enough to disturb the inertial rate of the payload when the outer gimbal moves. As the first step, a linear (viscous) model represented by the constant b_1 was used. Of course, in order to obtain more realistic model that predicts the annoying stick-slip behavior at slow speeds, more advanced models should be used, for example the popular LuGre model proposed in [16] or [31]. One way or another, the presence of friction causes undesired coupling between the payload and the carrier. Hence the transfer function from the disturbance ω_c to ω_1 measured at the payload is certainly nowhere close to zero as one might desire. At very low frequencies it is even exactly equal to 1 (see the Fig. 2.16).

Equations (2.36) through (2.45) constitute the full state-space model of the dual-stage benchmark system, i_1 and i_2 are the motor currents in [A], the inputs are the normalized voltages u_1 and u_2. Moment of inertia J_{mp} is defined as $J_{mp} = n_1^2(J_m + J_p)$. Remaining variables were introduced in Fig. 2.14 or are listed in table 2.2. Equations are also visualized the block diagram form in Fig. 2.15.

State equations :

$$U_{1_{\max}} u_1(t) = R_1 i_1(t) + L_1 \frac{di_1(t)}{dt} + r k_{e1}\Big(\omega_1(t) - \omega_2(t)\Big), \tag{2.36}$$

$$J_1 \dot{\omega}_1(t) = r k_{t1} i_1(t) - b_1\Big(\omega_1(t) - \omega_2(t)\Big), \tag{2.37}$$

$$U_{2_{max}} u_2(t) = R_2 i_2(t) + L_2 \frac{di_2(t)}{dt} + n_1 k_{e2}\Big(\omega_p(t) - n_2\omega_c(t)\Big), \tag{2.38}$$

$$\begin{aligned} J_{mp}\dot{\omega}_p(t) = n_1 k_{t2} i_2(t) &- k\Big(\varphi_p(t) - n_2\varphi_2(t)\Big) \\ &- b\Big(\omega_p(t) - n_2\omega_2(t)\Big) - b_p\Big(\omega_p(t) - n_2\omega_c(t)\Big), \end{aligned} \tag{2.39}$$

$$J_2\dot{\omega}_2(t) = k\Big(\varphi_p(t) - n_2\varphi_2(t)\Big) + b\Big(\omega_p(t) - n_2\omega_2(t)\Big) - r k_{t1} i_1(t), \tag{2.40}$$

$$\dot{\varphi}_p(t) = \omega_p(t), \tag{2.41}$$

$$\dot{\varphi}_2(t) = \omega_2(t), \tag{2.42}$$

$$\dot{\varphi}_1(t) = \omega_1(t). \tag{2.43}$$

Output equations :

$$y_1(t) = \omega_1(t), \tag{2.44}$$

$$y_2(t) = \varphi_1(t) - \varphi_2(t) = e_\varphi(t). \tag{2.45}$$

The equations (2.36) through (2.45) may be written in terms of the system transfer function matrix G

$$\begin{bmatrix} \hat{\omega}_1 \\ \hat{e}_\varphi \end{bmatrix} = G \begin{bmatrix} \hat{u}_1 \\ \hat{u}_2 \\ \hat{\omega}_c \end{bmatrix}. \tag{2.46}$$

Fig. 2.16 on the left shows the open-loop frequency responses of the two loaded motors independently. The magnitude response of the stage with the rotary DC motor is

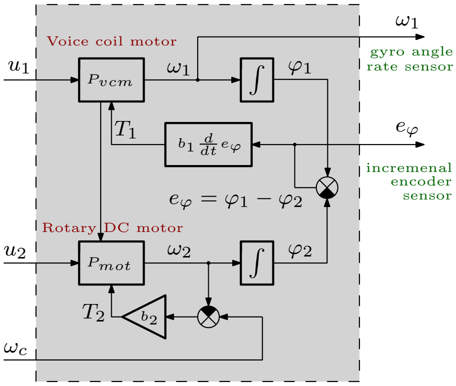


Figure 2.15: Block diagram of the benchmark model. Block diagram represents equations (2.36) through (2.45). It is only for the sake of simplicity of the figure that the reaction torque generated by the VCM and applied to the rotary motor is not depicted here and neither are the back emf voltages, which are derived from the relative and not the inertial rates. Furthermore a friction b_2 represents together both frictions b_p and b. Diagram clearly demonstrates the only way how disturbance ω_c can affect ω_1 that is via friction T_1 (or via the missing back emf voltage $rk_{e1}(\omega_1 - \omega_2)$).

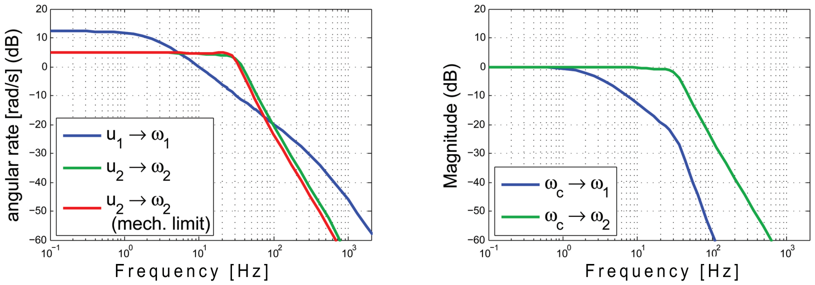
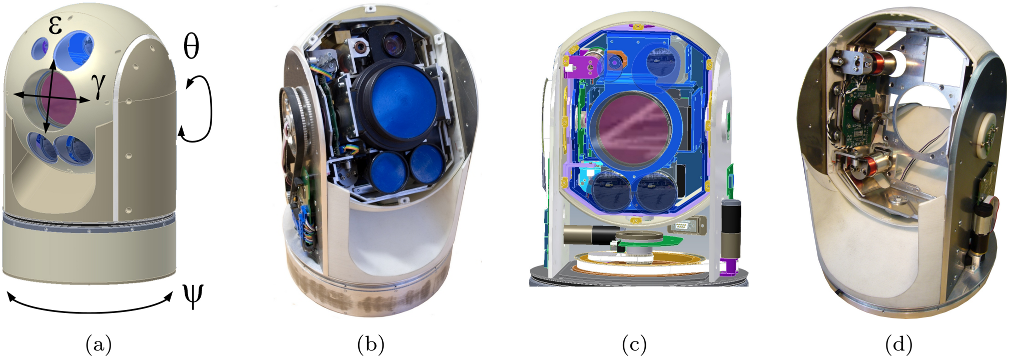


Figure 2.16: Left: Open-loop magnitude Bode plots for both stages. The inputs are the normalized voltages to the two motors, the outputs are the inertial angular rates [rad/s]. Right: Open-loop magnitude Bode plot for the disturbance attenuation.

represented by two plots, one for the situation when angle e_φ is below the physical limits, and the other for the situation when it hits the mechanical limits and the rotary DC motor is then pulling the payload along. No bouncing back was modeled. The moment of inertia J_2 of the outer stage then has to be increased by the moment of inertia of the payload J_1, hence the bandwidth is lowered. The same Fig. 2.16 on the right shows the open-loop magnitude bode plots for the disturbance attenuation. The inertial rate ω_1 is only affected by disturbance rate ω_c at very low frequencies. The middle and high frequency ranges exhibit natural attenuation. In contrast, the inertial rate ω_2 is affected by ω_c up to 40 Hz, at higher frequencies it is attenuated as well due to the elasticity of the belt gear. Both plots were generated using the mathematical model.

2.5 FOUR-GIMBAL DUAL-STAGE PLATFORM

The concept of the dual-stage configuration around the single axis promises major improvement in stabilization performance. This section introduces a mechanical configuration that is taking advantage of the dual-stage principle and at the same time allows for orienting the camera arbitrarily just as the classical double-gimbal configuration. The exterior look on the configuration appears identical to the double-gimbal, but it introduces another two gimbals inside the elevation gimbal that allow a fine movement of the payload around the two more axes. These two gimbals may move only by a limited angle (similarly as in the single-axis dual-stage benchmark model). The picture of the four-gimbal dual-stage platform is shown in Fig. 2.17. The platform was developed within a joint project coordinated by Czech Air Force and Air Defence Technological Institute (Vojenský technický ústav letectva a PVO) in collaboration with Czech Technical University in Prague and ESSA company. This platform is designated in this thesis by an acronym "S250" (the number 250 stands for its diameter in millimeters).



(a)

(b)

(c)

(d)

Figure 2.17: Photos and CAD models of the four-gimbal dual-stage platform S250.
(a) Platform is equipped with four motorized gimbals — the two outer (ψ, θ) and the two inner (ϵ, γ). Inner gimbals allow a fine rotation of the camera up-down and left-right within only a limited range.
(b) Uncovered. A dummy payload is used instead of real cameras. The photo shows both inner gimbals as well as the belt gear on the side through which the elevation θ is driven.
(c) The detail of the inner two gimbals assembly. The light pink and blue colors represent the inner gimbal [I], and inner elevation gimbal [U] respectively. The cover of the cameras (gray color) is the elevation gimbal [E]. Further description is given in the section 2.5.1.
(d) The two inner gimbals without any payload. The two actuators (voice coil motors) are visible (red coils).

The following section reports on development of a mathematical model of the four-gimbal dual-stage platform. Similarly as in the section 2.2, the Newton-Euler approach is used.

2.5.1 Coordinate frames and their rotations

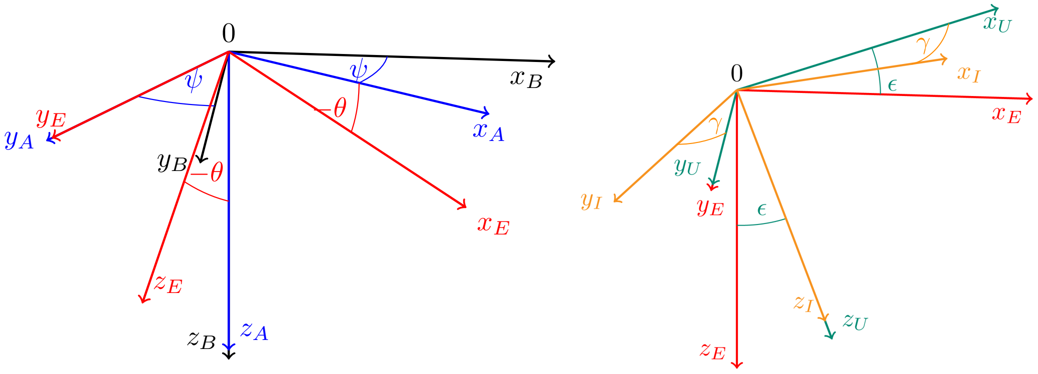


Figure 2.18: Composition of rotations of coordinate frames attached to the Base, Azimuth and Elevation gimbals on the left and Elevation, inner elevation U and Inner gimbals on the right.

Since the configuration of the four-gimbal platform is such that the two outer gimbals can be viewed as equivalent to the double gimbal configuration, the notation for the coordinate frames inherits the notation already used in section 2.2. That is

- **reference** coordinate frame [R]
- **base** coordinate frame fixed to the body of the carrier [B]
- coordinate frame attached to the outer (**azimuth**) gimbal [A]
- coordinate frame attached to the **elevation** gimbal [E].

To describe the four-gimbal platform, another two coordinate frames are necessary:

- The coordinate frame connected to the **inner elevation** gimbal [U] which can rotate with respect to the elevation frame [E] around the axis $y_U = y_E$ with the corresponding angle variable ϵ.
- The **inner** coordinate frame [I] which allows the rotation around $z_U = z_I$ using the corresponding inner cross-elevation gimbal with the associated angle γ[2].

$$[B] \xrightarrow[\psi]{z_B} [A] \xrightarrow[\theta]{y_A} [E] \xrightarrow[\epsilon]{y_U} [U] \xrightarrow[\gamma]{z_I} [I]$$

Figure 2.19: Diagram of gimbal rotation composition.

[2] The term cross-elevation was already introduced in section describing classical double gimbal and it denotes the axis perpendicular to the elevation axis.

The sequence of the key rotations expressing the pose of the inner gimbal (fixed to camera) with respect to the base (carrier) is schematically depicted in Fig. 2.18 and Fig. 2.19. Corresponding rotation matrices are

$$R_A^B = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.47}$$

$$R_E^A = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}, \tag{2.48}$$

$$R_U^E = \begin{bmatrix} \cos\epsilon & 0 & \sin\epsilon \\ 0 & 1 & 0 \\ -\sin\epsilon & 0 & \cos\epsilon \end{bmatrix}, \tag{2.49}$$

and finally

$$R_I^U = \begin{bmatrix} \cos\gamma & 0 & \sin\gamma \\ 0 & 1 & 0 \\ -\sin\gamma & 0 & \cos\gamma \end{bmatrix}. \tag{2.50}$$

The first two rotation matrices were already introduced in section 2.2. It is clear that the configuration of azimuth and elevation gimbals is identical to the classical double gimbal. Thus, the model for elevation and azimuth gimbals of the four-gimbal platform will be identical too. The only difference is that of adding another two gimbals (and coordinate frames) to the end of the kinematic chain. In order to not repeat these equations (namely equations (2.5) – (2.17)) in following sections it is only the *inner elevation frame* and *inner frame* angular rates and accelerations that are studied.

2.5.2 Forward recursion of Newton-Euler method

Inertial angular rates

Inertial angular velocity of the inner elevation frame is described by the vector ω_U^U

$$\omega_U^U = \begin{bmatrix} \omega_{Ux} \\ \omega_{Uy} \\ \omega_{Uz} \end{bmatrix}. \tag{2.51}$$

Inertial angular velocity of the inner frame, to which the camera is attached, expressed in the same frame, is described by the vector ω_I^I

$$\omega_I^I = \begin{bmatrix} \omega_{Ix} \\ \omega_{Iy} \\ \omega_{Iz} \end{bmatrix}. \tag{2.52}$$

Following the procedure from section 2.2, the inertial angular rate of the inner elevation frame [U] expressed in elevation frame [E] is

$$\omega_U^E = B_y\omega_E^E + y_U\omega_{Uy} = \begin{bmatrix} c_\theta(pc_\psi + qs_\psi) - \omega_{Az}s_\theta \\ \omega_{Uy} \\ s_\theta(pc_\psi + qs_\psi) + \omega_{Az}c_\theta \end{bmatrix}. \tag{2.53}$$

Expressing this inertial angular rate of inner elevation gimbal in its own frame gives

$$\omega_U^U = R_E^U\omega_U^E = R_E^U B_y\omega_E^E + R_E^U y_U\omega_{Uy}. \tag{2.54}$$

The relative angular rate $\dot{\epsilon}$ is then found as a difference between the y components of the vectors of inertial angular rate of the elevation and inner elevation gimbals

$$\begin{aligned}\dot{\epsilon} &= [\omega_U^U - \omega_E^U]_y = [\omega_U^U - R_E^U\omega_E^E]_y = \omega_{Uy} - \omega_{Ey} \\ &= \omega_{Uy} - \dot{\theta} + ps_\psi - qc_\psi.\end{aligned} \tag{2.55}$$

Proceeding finally to the inner gimbal [I], the angular rate of the inner gimbal expressed in the inner elevation frame is

$$\omega_I^U = B_z\omega_U^U + z_I\omega_{Iz}, \tag{2.56}$$

where again the matrix B_z blocks the rotation about the z_U axis since it is described by the independent variable ω_{Iz}. To express this vector in the coordinate frame attached to the inner gimbal the rotation R_U^I is required

$$\omega_I^I = R_U^I\omega_I^U. \tag{2.57}$$

The relative inner cross-elevation angle γ is given as the z component of a difference between the rates of the inner gimbal and inner elevation gimbal

$$\dot{\gamma} = [\omega_I^I - \omega_U^I]_z, \tag{2.58}$$

where the inertial angular rate of the outer gimbal can be expressed in the inner gimbal frame as

$$\omega_U^I = R_U^I\omega_U^U. \tag{2.59}$$

Even though both ω_U^I and ω_I^I were not enumerated due to their complexity, the resulting $\dot{\gamma}$ may be calculated and reads

$$\begin{aligned}\dot{\gamma} &= \omega_{Iz} - c_\epsilon\big(\omega_{Az}c_\theta + s_\theta(pc_\psi + qs_\psi)\big) + s_\epsilon\big(\omega_{Az}s_\theta - c_\theta(pc_\psi + qs_\psi)\big) \\ &= \omega_{Iz} - (pc_\psi + qs_\psi)(c_\epsilon s_\theta + s_\epsilon c_\theta) - (\dot{\psi} + r)(c_\epsilon c_\theta - s_\epsilon s_\theta).\end{aligned} \tag{2.60}$$

Angular accelerations

Determining angular accelerations of the two remaining moving gimbals ([U], [I]) consists in differentiating the expressions (2.54) and (2.57) in their respective rotating coordinate frames.

$$\alpha_U^E = \dot{\omega}_U^E + \omega_E^E \times \omega_U^E \tag{2.61}$$

Expressing this inertial angular acceleration of inner elevation gimbal in its own frame gives

$$\alpha_U^U = R_E^U(\dot{\omega}_U^E + \omega_E^E \times \omega_U^E) = R_E^U B_y \dot{\omega}_E^E + R_E^U y_U \omega_{Uy} + R_E^U\big(\omega_E^E \times (B_y \omega_E^E + y_U \omega_{Uy})\big). \tag{2.62}$$

The same principle used for the inner frame [I] gives

$$\alpha_I^U = \dot{\omega}_I^U + \omega_I^I \times \omega_I^U, \tag{2.63}$$

and

$$\alpha_I^I = R_U^I(\dot{\omega}_I^U + \omega_U^U \times \omega_I^U) = R_U^I B_z \dot{\omega}_I^I + R_U^I z_I \omega_{Iz} + R_U^I\big(\omega_U^U \times (B_z \omega_U^U + z_I \omega_{Iz})\big). \tag{2.64}$$

2.5.3 Backward recursion of Newton-Euler method, determining the torques

In Forward recursion of the Newton-Euler method it was possible to (simply) extend the model of the double gimbal for inner elevation gimbal [U] and the inner gimbal [I] and keep the equations for the azimuth [A] and elevation [E] gimbals from the double-gimbal equations. In the case of the backward recursion the situation is not that simple. The total torque exerted on the elevation gimbal must be extended for an extra term representing the reaction torque from the inner elevation gimbal. The torques must be calculated backwards along the kinematic chain. That is from the inner gimbal towards the azimuth gimbal (opposite the arrows in Fig. 2.19).

In the following text I_X is represents the inertia matrix of the each gimbal, where the X subscript represents one of the gimbals A, E, U, I. The structure is assumed to be in diagonal form as discussed in assumptions in section 2.2.2.

$$I_X = \begin{bmatrix} I_{Xx} & 0 & 0 \\ 0 & I_{Xy} & 0 \\ 0 & 0 & I_{Xz} \end{bmatrix} \tag{2.65}$$

The difference when compared to double gimbal is that now the payload which usually has the major moment of inertia is placed inside the inner gimbal I_I instead of the elevation gimbal I_E.

The total torque exerted on the inner gimbal [I] is equal to the derivative of the angular momentum of the gimbal. Following the rules for differentiation of vectors in rotating frames

$$T_I^I = I_I \alpha_I^I + \omega_I^I \times I_I \omega_I^I. \tag{2.66}$$

Proceeding one frame backwards to the inner elevation gimbal [U] gives

$$T_U^U = I_U \alpha_U^U + \omega_U^U \times I_U \omega_U^U + R_I^U T_I^I, \tag{2.67}$$

where the total torque exerted on the inner elevation gimbal must be extended for an extra term representing the reaction torque from the inner elevation gimbal (it must be though rotated first to appropriate coordinate frame).

The same procedure must be then repeated for the elevation gimbal T_E^E and the azimuth gimbal T_A^A

$$T_E^E = I_E \alpha_E^E + \omega_E^E \times I_E \omega_E^E + R_U^E T_U^U \tag{2.68}$$

and

$$T_A^A = I_A \alpha_A^A + \omega_A^A \times I_A \omega_A^A + R_E^A T_E^E. \tag{2.69}$$

The calculation of all the above equations can be easily done with arbitrary symbolic computation tool.

Nonlinear differential equations for dynamics

To derive nonlinear differential equations for the dynamics of four-gimbal platform it is desired to structure the total torque around the each gimbal axis (z_I, y_U, y_E, z_A) as the motor induced torque, the friction induced torque, and the remaining part of the dynamics (gyroscopic and coriolis terms, ...). The obtained equations are though very extensive. To get at least an essence of the dynamic equations, in the following assume the base rests still in inertial space, that is, $p = 0$, $q = 0$ and $r = 0$.

The nonlinear differential equation describing the motion of the inner gimbal around the z_I axis is

$$\begin{aligned} I_{Iz}\dot{\omega}_{Iz} \;=\; & +T_{Im} - T_{If}(\dot{\gamma}) \\ & +I_{Iz}\omega_{Uy}\Big(\omega_{Uy}\sin\gamma - \omega_{Az}\sin(\epsilon+\theta)\cos(\gamma)\Big) \\ & +I_{Iz}\omega_{Az}\sin(\epsilon+\theta)\Big(\omega_{Uy}\cos\gamma + \omega_{Az}\sin(\epsilon+\theta)\sin\gamma\Big) \\ & -(I_{Ix}-I_{Iy})\Big(\omega_{Uy}\cos\gamma + \omega_{Az}\sin(\epsilon+\theta)\sin\gamma\Big)\Big(\omega_{Uy}\sin\gamma - \omega_{Az}\sin(\epsilon+\theta)\cos\gamma\Big), \end{aligned} \tag{2.70}$$

where $T_{If}(\dot{\gamma})$ is cross-elevation joint friction and T_{Im} is the torque applied by inner cross-elevation voice coil motor.

Following the same principle, the differential equation describing the motion of the inner elevation gimbal around the elevation axis y_U is

$$\dot{\omega}_{Uy}(I_{Iy}\cos^2\gamma + I_{Ix}\sin^2\gamma + I_{Uy}) = +T_{Um} - T_{Uf}(\dot{\epsilon}) + \ldots^3, \tag{2.71}$$

where $T_{Uf}(\dot{\epsilon})$ is inner-elevation joint friction and T_{Um} is the torque applied by inner elevation voice coil motor. Notice that moment of inertia is not in this case constant as in (2.70),

[3] The equation is very extensive. To get at least an essence of the dynamics the simplified version with $\sin\epsilon = \sin\gamma = 0$ and $\cos\epsilon = \cos\gamma = 0$ is in (2.74).

but it is composed of moment of inertia of both inner and inner-elevation gimbals being the function of γ.

The differential equation describing the motion of the elevation gimbal [E] around the elevation axis y_E is

$$I_{Ey}\dot{\omega}_{Ey} + \dot{\omega}_{Uy}(I_{Iy}\cos^2\gamma + I_{Ix}\sin^2\gamma + I_{Uy}) = +T_{Em} - T_{Ef}(\dot{\theta}) + \ldots, \qquad (2.72)$$

where again $T_{Ef}(\dot{\theta})$ is elevation joint friction and T_{Em} is the torque applied by elevation DC motor. The last remaining differential equation, describing the azimuth gimbal dynamics is not enumerated here due to its extensiveness.

Since inner gimbals angles γ and ϵ are mechanically limited to be ±5° at maximum and moreover the aim of the control design is to ensure that both angles are close to the zero angle most of the time, equations (2.70), (2.71) and (2.72) may be simplified[4] by assuming $\sin\epsilon = \sin\gamma = 0$ and $\cos\epsilon = \cos\gamma = 1$. The error caused by this assumption may be taken as yet another disturbing torque acting in gimbals. The equations (2.73), (2.74) and (2.75) are the simplified versions of (2.70), (2.71) and (2.72).

$$I_{Iz}\dot{\omega}_{Iz} = +T_{Im} - T_{If}(\dot{\gamma}) + (I_{Ix} - I_{Iy})\Big(\omega_{Uy}\omega_{Az}\sin(\epsilon+\theta)\Big) \qquad (2.73)$$

$$\begin{aligned} \dot{\omega}_{Uy}(I_{Iy} + I_{Uy}) &= +T_{Um} - T_{Uf}(\dot{\epsilon}) \\ &\quad +(I_{Ux} - I_{Uz})\omega_{Az}^2\cos\theta\sin\theta \\ &\quad +(I_{Ix} - I_{Iz})\omega_{Az}\omega_{Iz}\sin\theta \end{aligned} \qquad (2.74)$$

$$\begin{aligned} I_{Ey}\dot{\omega}_{Ey} &= +T_{Em} - T_{Ef}(\dot{\theta}) \\ &\quad -\dot{\omega}_{Uy}(I_{Iy} + I_{Uy}) \\ &\quad +(I_{Ex} - I_{Ez} + I_{Ux} - I_{Uz})\omega_{Az}^2\cos\theta\sin\theta \\ &\quad +(I_{Ix} - I_{Iz})\omega_{Az}\omega_{Iz}\sin\theta \end{aligned} \qquad (2.75)$$

2.6 CONCLUSION

This chapter provided mathematical models of the kinematics and dynamics for various mechanical configurations using the mass stabilization concept. The most important are models for double-gimbal, single-axis dual-stage model and full four-gimbal dual-stage model. Mathematical models of these configurations will be used in subsequent chapters to develop control structure for the line-of-sight stabilization. Before proceeding to development of the line-of-sight stabilization the next chapter will at first introduce some fundamentals of the single-axis control structure.

[4] It must be emphasized at this place that this simplification does not represent any linearization but the simple neglecting as $\sin 5° = 0.01 \cong -40\text{dB}$.

Chapter 3

Single-axis control structure

This chapter recapitulates one fundamental in automatic motion control — *cascade* control structure. Introduction of this concept is crucial because the subsequent chapters always deal only with a specific part of the cascade. In order to present the main principles without any added complexity, it is the single axis configuration only (classical DC motor with an attached payload) that is studied in this chapter.

3.1 CASCADE CONTROL STRUCTURE

Standard feedback control configuration in motion control systems is that of a cascade. The cascade consists of three levels, or three feedback loops, one inside the other. The most inner feedback loop is responsible for following the reference current (or applied torque which is proportional to the current via constant k_t). It controls the applied voltage. The middle loop — angular velocity feedback loop — is responsible for following the reference velocity. It sets the current (or torque) to achieve it. In case of no velocity commands, the task is to keep the system still irrespectively of exogenous disturbing forces and/or torques. The outer loop is responsible for following the reference position (or angle in the rotational systems). It sets the reference value for the inner (velocity) feedback loop. The configuration is visualized in Fig. 3.1.

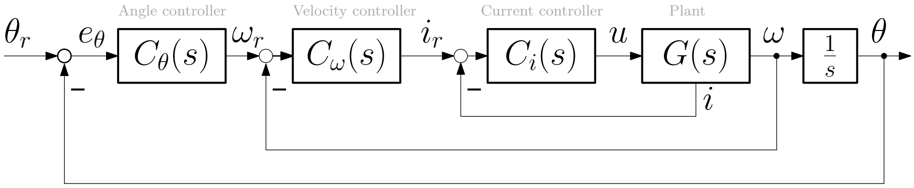


Figure 3.1: Cascade feedback control structure. $G(s)$ is a model of a motor with an attached payload, a the transfer function with a single input u and two outputs ω[rad/s] and i[A].

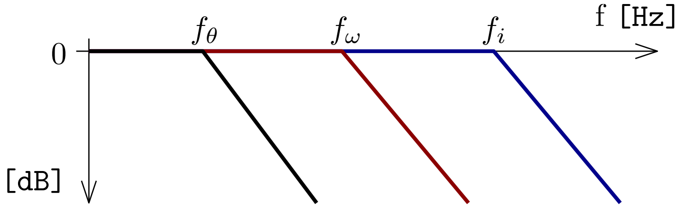


Figure 3.2: Sketch of the bandwidth of all three closed loops within the cascade.

In the case of inertial stabilization systems the middle velocity loop is usually implemented using angular rate gyros as sensors. The outer position loop is though usually implemented in various configurations depending on the required functionality at the moment. The most obvious configurations are

- *Image tracker* configuration, where an image tracker is used as a position sensor. Image tracker is an algorithm based on an advanced computer vision techniques that allows to detect the object location in the camera image by real-time processing the video signal.

- *GPS lock* configuration, where GPS receiver together with laser rangefinder are used as sensors to calculate the target coordinates. The aim of this configuration is to keep (lock) specified GPS coordinates in the camera view.

- *Manual* configuration, where the incremental encoders that are attached to platform joints are used as a sensors. The aim of this configuration is to provide a possibility to setup point of view with respect to the carrier and not to the inertial space.

The possibility to keep the two inner loops unchanged while switching the outer position loop from various configurations represents the major benefit of introducing the cascade control structure.

A crucial assumption for the independence of the design of all the three controllers in Fig. 3.1 is that the bandwidth of inner loops is always much higher (usually at least ten times or more) than the bandwidth of outer loops. Designing the cascade then proceeds by designing the controllers for inner loops first and then for the outer. The inner loop is then viewed as closed — that is — having the (closed-loop) transfer function approximately equal to one. This simplifies the design and implementation of the control system dramatically compared to a full MIMO controller.

3.2 CURRENT LOOP

The innermost loop in the cascade is the current loop with a current controller. The detailed scheme of the current controller together with the motor structure is in Fig. 3.3. Although it is certainly possible to skip designing and implementing a current controller (some cheaper motion control applications do skip it), there are several reasons for implementing it:

1. The motor torque is proportional to the current through its winding (via the torque constant k_t). The (outer) rate controller can then specify the required reference torque to be applied to the payload.
2. The effective value of the current ($i_{\mathrm{rms}} = \sqrt{\frac{1}{T}\int i^2(t)\mathrm{d}t}$) is proportional to the heat produced in the windings. Thus the current controller may be complemented by a set of heuristics that are providing a thermal protection by limiting a required torque (reference current).
3. The ohmic resistance of the windings is always temperature-depended, hence the amplitude of the current or torque or heat generated by the constant applied voltage is depended on the temperature too.
4. A well tuned current controller is capable to cancel out the electrical dynamics of the motor. The response of the torque step response may be speeded up this way.
5. The current controller makes the back emf voltage transparent for outer loops.

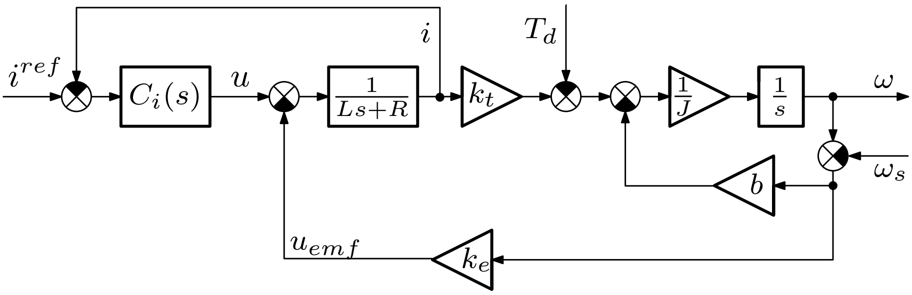


Figure 3.3: Location of the current controller within the motor model scheme.

To see things in an analytical form, the transfer functions for ω and i are given bellow

$$\omega(s) = \frac{k_t}{(Ls+R)(Js+b)+k_tk_e}u(s) + \frac{(Ls+R)b+k_tk_e}{(Ls+R)(Js+b)+k_tk_e}\omega_s(s) + \frac{-(Ls+R)}{(Ls+R)(Js+b)+k_tk_e}T_d(s) \tag{3.1}$$

$$i(s) = \frac{Js+b}{(Ls+R)(Js+b)+k_tk_e}u(s) + \frac{-k_eJs}{(Ls+R)(Js+b)+k_tk_e}\omega_s(s) + \frac{k_e}{(Ls+R)(Js+b)+k_tk_e}T_d(s). \tag{3.2}$$

To keep the most relevant part of the dynamics to get an easier insight, let's assume zero friction and inductance

$$\omega(s) = \frac{\frac{1}{k_e}}{\frac{JR}{k_tk_e}s+1}u(s) + \frac{1}{\frac{JR}{k_tk_e}s+1}\omega_s(s) + \frac{-\frac{R}{k_tk_e}}{\frac{JR}{k_tk_e}s+1}T_d(s) \tag{3.3}$$

$$i(s) = \frac{\frac{J}{k_t k_e}s}{\frac{JR}{k_t k_e}s+1}u(s) + \frac{-\frac{J}{k_t}s}{\frac{JR}{k_t k_e}s+1}\omega_s(s) + \frac{\frac{1}{k_t}}{\frac{JR}{k_t k_e}s+1}T_d(s) \tag{3.4}$$

Comparing the terms with ω_s in the last two equations, the well-known advantage of the current feedback can be seen: the current predicts a rotation of the stator (it includes a derivative term), so if this variable is measured and used in a feedback controller, much faster attenuation of this disturbance can be obtained and the system may behave as if no kinematic coupling was present.

3.2.1 Current controller design

Current controller is usually designed as a PI controller. To see necessity of the integrator, see first the behavior of the simple P controller $C_i = k_p$. With this controller the transfer function from i^{ref} to i is

$$T_i^P(s) = \frac{bk_p + Jk_p s}{JLs^2 + (Lb + Jk_p + JR)s + Rb + bk_p + k_e k_t}, \tag{3.5}$$

with the steady state value of the step response equal to

$$\lim_{s\to 0} T_i^P(s) = \frac{bk_p}{Rb + bk_p + k_e k_t}. \tag{3.6}$$

Unless the constant k_p is much higher than term $Rb + k_e k_t$ there is always steady state error. To improve steady state behavior the PI controller $C_i = k_p + k_i/s$ is usually implemented and the transfer function from i^{ref} to i is

$$T_i^{PI}(s) = \frac{Jk_p s^2 + (Jk_i + bk_p)s + bk_i}{JLs^3 + (Lb + Jk_p + JR)s^2 + (Rb + Jk_i + bk_p + k_e k_t)s + bk_i}, \tag{3.7}$$

with the steady state value of the step response equal to

$$\lim_{s\to 0} T_i^{PI}(s) = 1. \tag{3.8}$$

Adding an integrator into current controller may be though a little bit tricky. Some clarification is needed first.

In the case of the zero friction ($b = 0$) it was seen in (3.4) that the transfer function from current voltage u to current i contained derivative. Thus, when the integrator is to be implemented, this zero is canceled by the integrator. This may be proved by calculation of the steady state value of the transfer function from reference current i^{ref} to applied voltage u. This value is equal to

$$\lim_{s\to 0} T_u^{PI}(s) = \frac{Rb + k_e k_t}{b}. \tag{3.9}$$

Now it is clear that even when the friction is not zero but only small enough the output voltage of the current controller is reaching the saturation, since in real applications a power supply voltage is always limited. This leads to the inability to track the constant

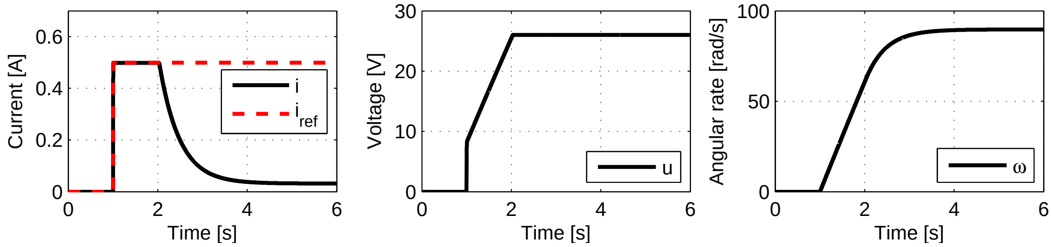


Figure 3.4: Simulation of tracking the reference step in the current. Reference current i^{ref} is tracked only until the voltage u reaches its maximum (power supply) value.

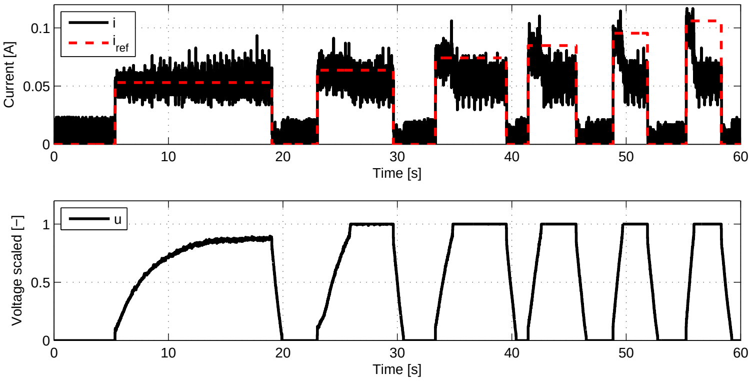


Figure 3.5: Experiment of tracking the reference step in the current. The same configuration as in Fig. 3.4 but here the graph contains data from the real experiment provided by the smallest double-gimbal platform developed within projects — Fig. 2.6 (c)

reference current. Such behavior is demonstrated by simulation and also by experiment in Fig. 3.4 and Fig. 3.5.

Since the current loop serves as the most inner loop within the cascade, its bandwidth should be the highest. Usually it is recommended to be ten times higher than the bandwidth of the velocity loop. If the purpose of the current controller is to cancel out the electric dynamics of the motor, the sampling rate must be sufficiently higher than the electrical time frequency L/R. For the most permanent-magnet motors, voice-coil motors or direct drives assumed in this thesis these values are higher than several hundreds of Hz, thus the appropriate current measurement must always be implemented. Sometimes, this loop may be already implemented as analog within the power-driving circuit — this approach overcomes the troubles with insufficient limited sampling rates achievable by the microcontroller.

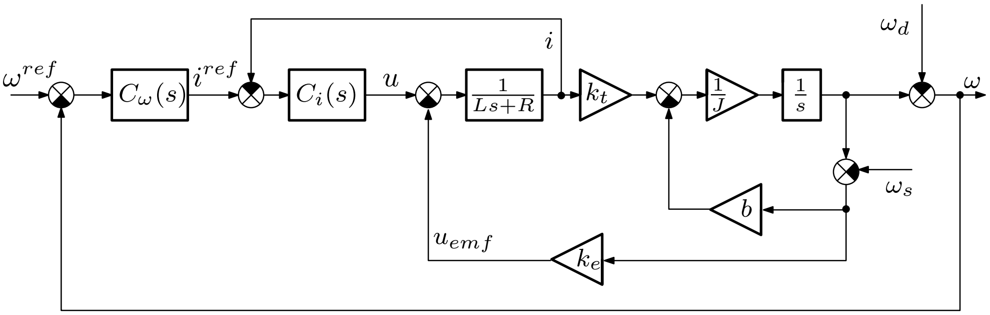


Figure 3.6: Velocity control scheme with depicted disturbances ω_s and ω_d.

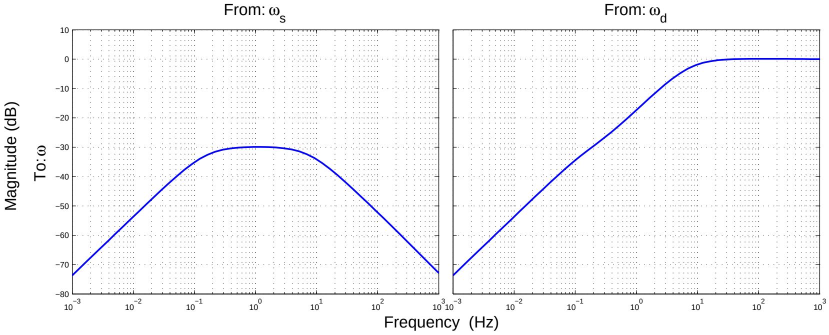


Figure 3.7: Bode graphs of transfer functions from two considered disturbances to the output. Graphs were generated using the model (2.30) – (2.34) and PI angular rate controller. Left: From ω_s to ω. Right: From ω_d to ω. Graphs clearly show that because ω_s enters the loop via the friction b, it is well rejected at all frequencies. On the contrary, the disturbance ω_d enters the loop directly, so that it can be well rejected only at frequencies at which the motor has the sufficiently high gain — that is at low frequencies.

3.3 ANGULAR VELOCITY (RATE) LOOP

Angular velocity loop (or the angular rate loop) is closed around the already designed current loop. Angular rate controller $C_\omega(s)$ takes the difference between the required and the measured angular rates as an input and calculates the required motor current as an output. The scheme is in Fig. 3.6.

The velocity (the angular rate) controller $C_\omega(s)$ is often designed as a PI controller with a saturation. The proper implementation of the anti-windup scheme is always essential. The topic is discussed further in section 3.5. The main advantage of the cascade structure is that it is usually possible to design PI velocity controller by manual tuning the parameters. Nevertheless, when the model and its parameters are already known, as described for example by (2.30) – (2.34), model based design tools such as $\mathcal{H}_\infty$ optimal control may be employed.

The aim is to design velocity controller such that the two disturbances ω_s and ω_d are sufficiently rejected (see the disturbances in Fig. 3.6). Both of them are the most common in line-of-sight stabilization applications. The first one — ω_s enters the loop basically only via the friction b[1]. Disturbance ω_s is in fact the part of the carrier's angular rate vector which acts around the joint axis, so that the mass stabilization principle itself helps to reject this disturbance. On the contrary ω_d enters the loop directly. This disturbance corresponds to the carrier motion that is projected into the measured vector (which is to be stabilized) but it is not paralel with the joint axis. Such disturbances are due to their direct entering the loop not very well rejected. Two magnitude bode plots shown in Fig. 3.7 demonstrate achievable disturbances rejection with using a simple PI control.

3.3.1 Inertial angular rate sensors (gyroscopes)

The angular rate sensors are denoted in the domain of inertial stabilization as *gyroscopes* or *gyros*. In low-end and low-cost applications usually MEMS gyros[2] are employed. High-end and hi-cost applications typically require the use of Fibre optic gyroscopes (FOG)[3].

This section presents a brief comparison of typical representative sensors from MEMS and FOG groups that were employed in series of projects devoted to inertially stabilized platforms at Czech Technical University. Studied sensors are the inertial measurement module ADIS 16400 that is offering 3-axis accelerometer, gyroscope and magnetometer in a single package, 3-axis MEMS gyroscope IGT 3200 and optical fibre gyroscope DSP 1750. Parameters of sensors are listed in the table 3.1.

Sensor	Manufacturer	Type	Axes	Bandwidth [Hz]	Noise meas. [°/s] rms	Noise doc. [°/s] rms
ADIS 16400	Analog Devices	MEMS	3	330	0.34	0.9
IGT 3200	InvenSense	MEMS	3	256	0.12	0.38
DSP 1750	KVH	FOG	2	440	$8.82 \cdot 10^{-5}$	NA

Table 3.1: List of studied sensors with their parameters. "Bandwidth" was obtained from from sensors' datasheets so as the "Noise doc." parameter which is the total rms noise. Parameter "Noise meas." is the experimentally measured rms value of the noise.

One of the most important parameters that influences the precision of angular rate reference tracking is the noise present in the angular rate signal. In order to compare noise levels of the three mentioned sensors a simple experiment was conducted. Sensors were placed on an anti-vibrational table to be isolated from a vibration of an environment. Since all sensors were isolated from the other movement, their measured output contained indeed only the noise. The recorded signal is shown in Fig. 3.8 on the left. It is obvious that Fibre optic gyro signal provides approximately two orders lower noise comparing to other two MEMS sensors. This fact is also demonstrated using histograms in Fig. 3.9 which also prove

[1] It enters the loop indeed also via the so-called *electrical friction* term k_e. This term is though negligible comparing to the friction b.

[2] http://en.wikipedia.org/wiki/Vibrating_structure_gyroscope#MEMS_Gyroscopes

[3] http://en.wikipedia.org/wiki/Fiber_optic_gyroscope

that the noise may be regarded as gaussian (at least in the FOG case). To see also frequency domain properties of the noise, FFT analysis was conducted and results are presented in Fig. 3.8 on the right. In case of FOG is the amplitude of the noise mostly constant in the entire frequency spectrum, which is the property of the *white noise*. Fibre optic gyro evinces again the best results since both MEMS sensors provide comparing to FOG worse results.

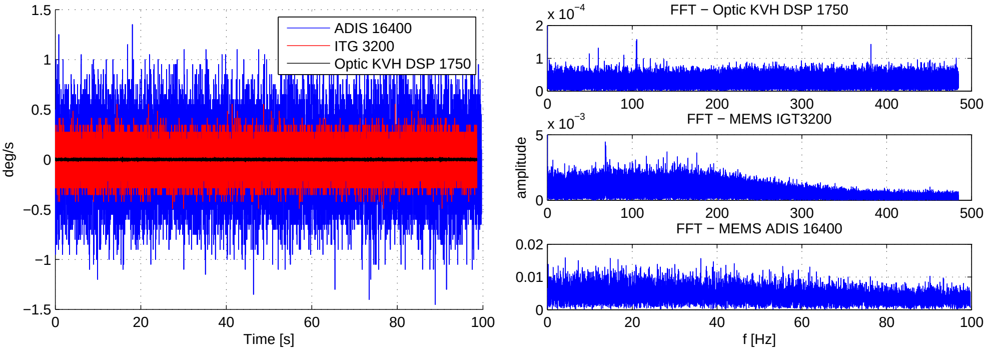


Figure 3.8: Left: Measured angular-rate signals recorded during the experiment. Sensors were placed onto an anti-vibrational table in order to avoid unwanted disturbing motion. All sensors in fact provide only their noise as the output.
Right: FFT analysis of recorded data. Amplitude of the noise present in DSP 1750 is constant at all frequencies — this is the property of the white noise. Gaussian white noise in the signal is an assumption in a lot of results in the state observers theory.

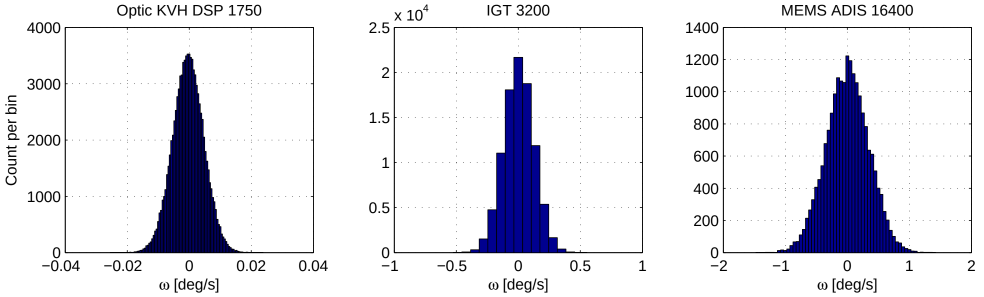


Figure 3.9: Histograms of recorded data from Fig. 3.8 left.

3.4 (ANGULAR) POSITION LOOP

(Angular) position loop as the outermost loop of the cascade is closed around the already designed velocity loop. As discussed at the beginning of this chapter, in the field of inertial stabilization there are several modes of operation of the position loop depending on the currently required functionality. An operator of the inertially-stabilized system can always switch the mode operation of the device according to his or her wish. In order to not entangle in technical details of whatever device's functionalities, let's assume a purely general case with just some angular or position sensor. The following text is valid for all of the modes or configurations.

A general structure of the position loop with the feedback control configuration is demonstrated in Fig. 3.10. The control loop contains two source signals

- the reference θ_r corresponding to operator clicking to another object to be locked and tracked (in the image or in the map), or by another way specifying the reference;

- the disturbance θ_t corresponding to the movement of the target object itself (within the camera screen or within the world frame represented by GPS coordinates).

The reference θ_r is known to the controller and is limited in the amplitude so that it is usually a constant or a step. The disturbance θ_t is not not known to the controller, moreover the character of this disturbance is assumed to be in general a ramp. Such requirement is obvious since the target usually travels with some velocity thus it is changing its coordinates with some speed. The type of both disturbances θ_r and θ_t is very important for the design of the position controller C_θ.

According to the *internal model principle* [32] to reject the disturbance of some type, the model of the disturbance must be included in the open loop. In order to follow the constant reference signal θ_r, the model of the constant must be included in the open loop — that is an integrator. By inspecting the scheme in Fig. 3.10 it is clear that the integrator is already present in the open loop[4] even if the position controller C_θ includes only a P controller.

In the case of the disturbance θ_t the situation is more complicated. Since this disturbance was assumed to be of a ramp type, this means that the open loop needs to contain a double integrator. The controller C_θ must include an integrator. The PI controller should suffice the requirement.

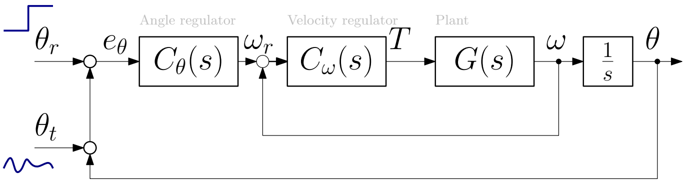


Figure 3.10: Cascade configuration of a feedback motion control system. The feedback control system needs to track both large steps in the reference position (corresponding to manual selection of another target to be tracked) and slow evolutions of the reference position (corresponding to the motion of the target).

[4] the integrator between ω and θ

3.4.1 Role of the feedforward and trajectory shaping

In spite of the classical feedback controller scheme provides functioning results, there is present no feedforward controller as it is known in robotics applications [13]. The missing feedforward reference shaper brings to the forefront a presence of the peak in the motor current when the step in the reference signal θ_r is required (the peak is also often called as Jerk). The simulation that is introducing two Jerks during steps in references θ_r and θ_t is shown in Fig. 3.11 on the left.

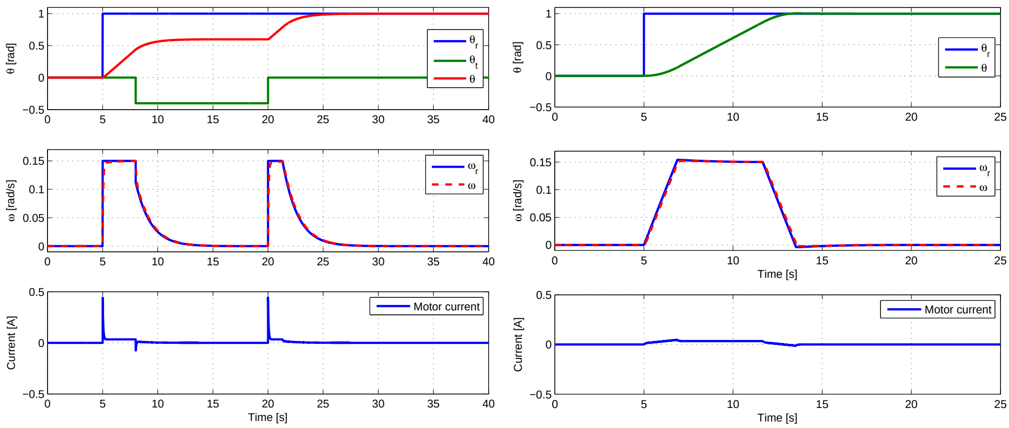


Figure 3.11: Left: Demonstration of the control configuration with proportional feedback gains (no shaping of the reference signal, no feedforward). Undesirable peaks in the required torque, hence the current are observed.
Right: Simulation of the jerk free step response. The simulation was generated using the scheme from Fig. 3.12 with applying the reference shaper for the reference signal θ_r.

Apparently, the feedback-only scheme is not very plausible because it responds to the step in the reference variable by requesting the highest possible torque, hence current. The peaks in the current are certainly not welcome in real applications.

3.4.2 Reference shaping (or trajectory planning) while the target moves

Neglected a bit in standard control systems textbooks, the topic of shaping the reference signal is well-studied in basic textbooks on robotics such as [13] or more recent research papers such as [33]. Usually it is studied under the name of trajectory planning. The abrupt step change in the reference signal is replaced by a smoother trajectory, be it a cubic polynomial or composition of several segments (cubic, linear) yielding the popular S-curves. Whether the reference signal is produced by a generator or a filter is discussed in [34]. The principle is sketched in Figure 3.12.

The simulation outcomes are in Fig. 3.11 on the right. They show that when the target does not move, the task of (re)pointing is perfectly helped by preshaping the reference

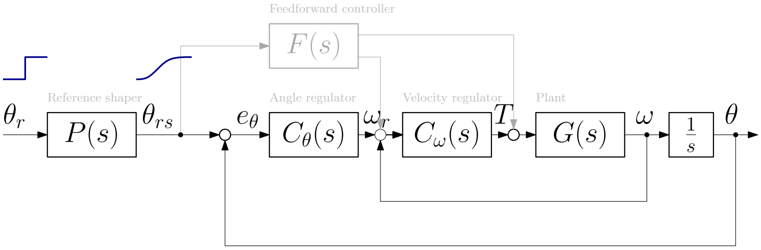


Figure 3.12: Cascade configuration of a feedback motion control system with a shaped reference and feedforward.

trajectory.

However, as soon as the target to be tracked moves, the tracking performance of this scheme deteriorates. The reason is that during the repointing phase the system works effectively open-loop. This is highlighted by adding a new reference input in Fig. 3.13.

3.4.3 Trajectory shaping vs. tracking

It would be nice, if the trajectory-shaped scheme from Fig. 3.12 could be exploited such to be used also in the situation with the visual feedback. Unfortunately, as it is demonstrated in Fig. 3.13 and Fig. 3.14, this scheme is not quite useful since it does not take into account additional input in terms of the target movement θ_t.

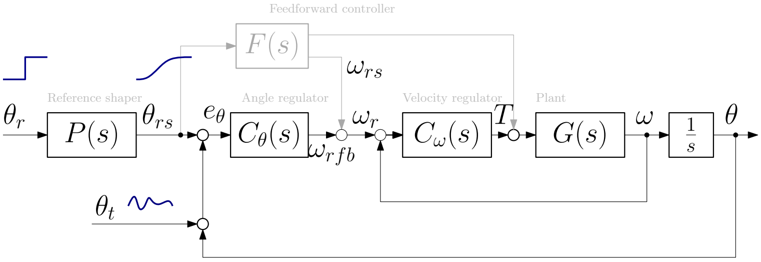


Figure 3.13: Cascade configuration of a feedback motion control system with a shaped reference and feedforward.

The simulation outcomes supporting the claim about deterioration of the tracking with preshaped reference and a moving target are in Fig. 3.14 on the left.

3.4.4 Trajectory shaping by using saturation and rate limiter

In the case when it does not matter that both references θ_r and also θ_t are shaped by the reference shaper, the role of the shaper then may be adopted by the limiting the output of the feedback controller C_θ. The saturation of the controller output corresponds to the slope of the step response and the rate-limiting the controller output may be used to limit the jerks. This scheme is visualized in Fig. 3.15 with simulations in Fig. 3.14 on the right.

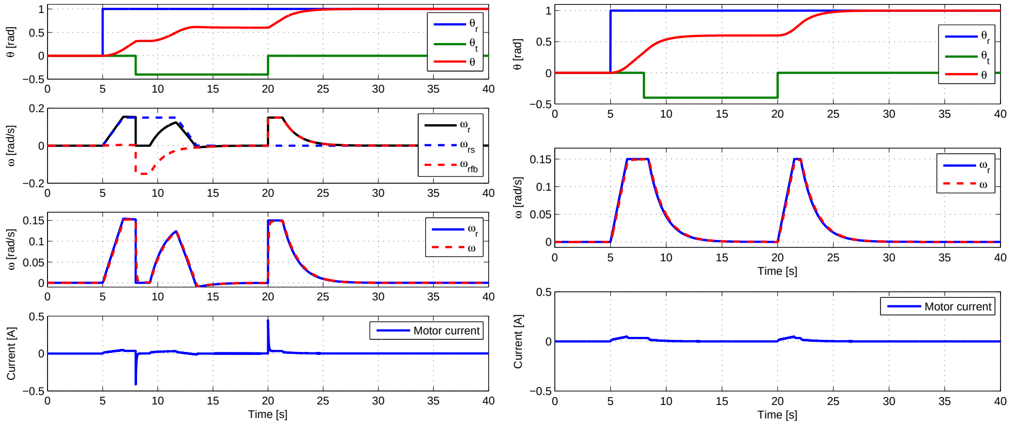


Figure 3.14: Simulation results of the cascade scheme with a trajectory generator. The target moves during pointing which leads to problems.

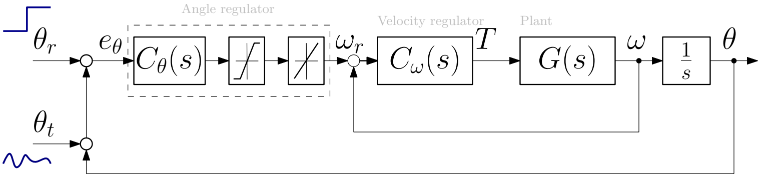


Figure 3.15: Cascade configuration with trajectory shaping done using the saturation and the rate limiter in the position controller.

3.5 SATURATING THE CONTROLLER SIGNAL

A controller with the integral action combined with an actuator that becomes saturated can produce some very undesirable behavior. This behavior appears when the error signal is so large that integrator saturates the actuator. When finally the error signal is crossing the zero value it may take a long time until the state of the integrator is reduced back to zero. This effect is called *integrator windup*.

In order to avoid windup issues, controllers with an integrator should be always implemented properly with the *antiwindup*. See the one of the possible implementation of the antiwindup in Fig. 3.16. The scheme was adopted from [35].

3.6 BIAS PRESENT IN GYRO RATE SIGNAL

In reality, it is inevitable that the measurement of the rate (velocity) signal is distorted by some constant or slowly varying offset (bias). This is even more serious for inertial measurement performed by cheap MEMS devices with a limited possibility of real-time recalibration. The influence of a bias in the rate signal can be investigated in terms of a

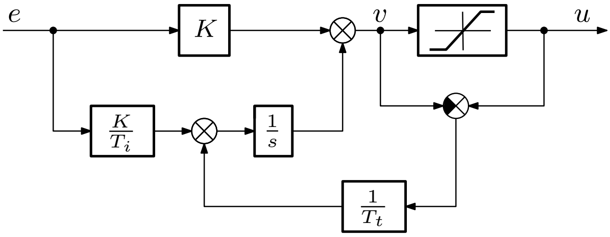


Figure 3.16: PI controller with an antiwindup. When the saturation of the actuator occurs, the integrator is being un-integrated via the time constant T_t. The state of the integrator remains always such to avoid the windup (avoid undesirable growth in v).

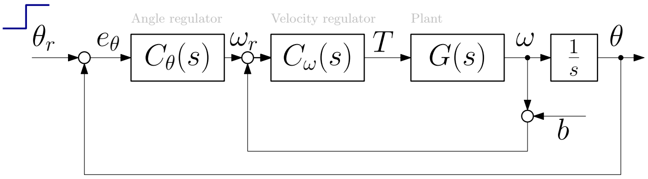


Figure 3.17: Bias signal entering the feedback loop. At certain conditions bias may cause steady-state tracking error in position θ.

steady-state tracking error. Consider the closed-loop system from Fig. 3.17 with the bias input b as depicted.

In the following use a PI control for the rate loop, that is, $C_\omega(s) = K_i/s + K_p$ and a proportional control for the outer position loop, that is, $C_\theta(s) = P_p$. The transfer function from the gyro bias $b(t)$ to the position output $\theta(t)$ is

$$S(s) = \frac{-k_t(K_i + K_p s)}{JL_a s^4 + JR_a s^3 + (k_e + K_p)k_t s^2 + (K_p P_p + K_i)k_t s + K_i P_p k_t} \tag{3.10}$$

and its steady-state response seen to be

$$\lim_{s\to 0} S(s) = -\frac{1}{P_p} \tag{3.11}$$

is nonzero. Hence, not only the disturbance in the outer position loop but even the disturbance (bias) in the middle velocity loop causes nonzero steady-state tracking error. Replacing the proportional controller C_θ by a PI controller $C_\theta(s) = P_p + P_i/s$ brings the steady-state limit

$$\lim_{s\to 0} S_{PI}(s) = 0. \tag{3.12}$$

Implementing the PI position controller thus helps reducing the bias disturbance. In practice it is though more convenient to implement some kind of ”bias observer” to reject bias disturbance completely. It will be shown later in chapter 6 that implementing PI position control does not ensure complete bias rejection when some delay compensation algorithms are introduced.

3.7 CONCLUSION

In this chapter the fundamentals of the single-axis motion control were introduced. It was explained that the approach of using the cascade structure is highly suitable for the implementation of control algorithms for the inertial stabilization. The major advantage resides in the ability to design controllers at each level of the cascade separately.

Understanding the concept of cascade control structure is crucial for the rest of this thesis, because immediately following chapter 4 actually deals with using inertial rate gyros in the velocity loop to implement the line-of-sight stabilization and the chapter 5 shows an extension of the gyroscope-based velocity loop by an image-based pointing loop (that plays the role of the position loop).

Finally this chapter introduced some often issues encountered in designing control algorithms such as consequences of the bias presented in velocity sensors.

Chapter 4

Line-of-sight inertial Stabilization

The very basic control task for steerable cameras or antennas mounted on mobile carriers such as trucks, unmanned aircrafts or ships, is to keep the commanded line of sight (optical axis) still even in the presence of various disturbing phenomena like the mass imbalance, the aerodynamic (or wind-induced) torque and possible the kinematic coupling between gimbal axes. As several mechanical configurations were introduced in chapter 2 this chapter presents the line-of-sight stabilization control design for all of them. The chapter is concluded by introducing feedforward control design to overcome issues with balancing the payload around gimbal axes. Algorithms are accompanied by numerical simulations and real experimental results.

Inertial line-of-sight stabilization on mobile carriers

Motivated also by defence technological needs, the topic of inertial stabilization was studied extensively in the past few decades. Several relevant papers from 1970s through 1990s were archived in the selection [36]. Dedication of a full issue of *IEEE Control System Magazine* (February 2008) featuring nice survey papers [37], [1] and [38] confirms that the topic is still relevant for the engineering community. Another recent issue of the same journal brings a rigorous analysis of control problems related to a standard double gimbal system [14], though it is not directly applicable to the inertial stabilization.

4.1 DOUBLE GIMBAL INERTIAL STABILIZATION

The mathematical model of the double gimbal was thoroughly studied in section 2.2. This section uses the formalism introduced therein. The basic scenario is sketched in Fig. 4.1. In order to make the line of sight insensitive to external disturbances, a simple controller structure can be used. Two decoupled SISO inertial rate controllers suffice, one for each measured (component of the) inertial angular rate. Namely,

- the inertial angular rate ω_{Ey} (also denoted with the mnemotechnic ω_{EL}) of the payload about the axis of the elevation motor (camera elevation rate),
- and the inertial angular rate ω_{Ez} of the payload around its own vertical axis, also nicknamed camera cross-elevation rate (and denoted ω_{CEL}) since its axis is always orthogonal to the ω_{EL} axis.

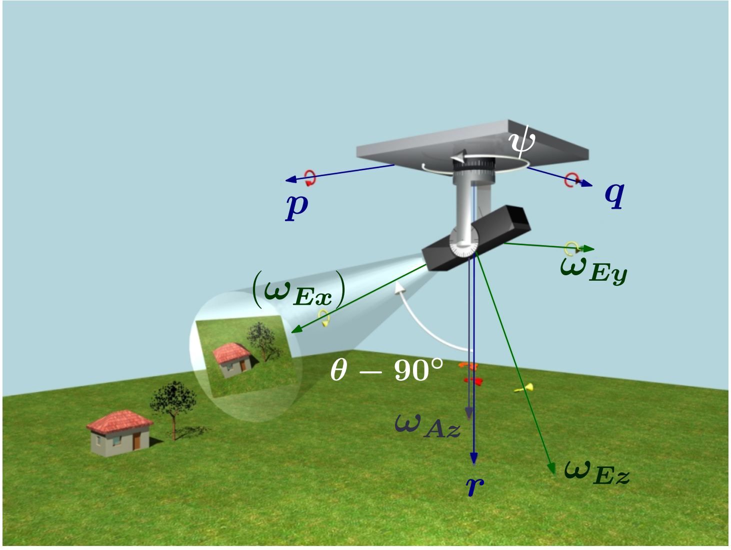


Figure 4.1: Basic scenario for the inertial line-of-sight stabilization. Depicted in green are the components $\omega_{Ex}, \omega_{Ey}, \omega_{Ez}$ of the vector of inertial angular rate of the elevation frame (as measured by MEMS gyros attached to the camera), blue vectors p, q, r denote the rate components of the base (UAV here). The ω_{Az} component is attached to the outer gimbal (the other two components are not shown). Two white arcs denote the two relative angles. The origins of the coordinate frames are assumed to coincide in the computations.

This is visualized in Fig. 4.1. The resulting decoupled controller configuration is in Fig. 4.2. It is clear from Fig. 4.1 that the cross-elevation controller must include a secant gain correction $1/cos(\theta)$, because the motor in the azimuth gimbal cannot directly affect $\omega_{\mathrm{CEL}}(=\omega_{Ez})$. It can only do so indirectly through ω_{Az}. It is only when the camera is pointing to the horizon, that is, when $\theta = 0$, that $\omega_{Az} = \omega_{\mathrm{CEL}}(=\omega_{Ez})$. This fact

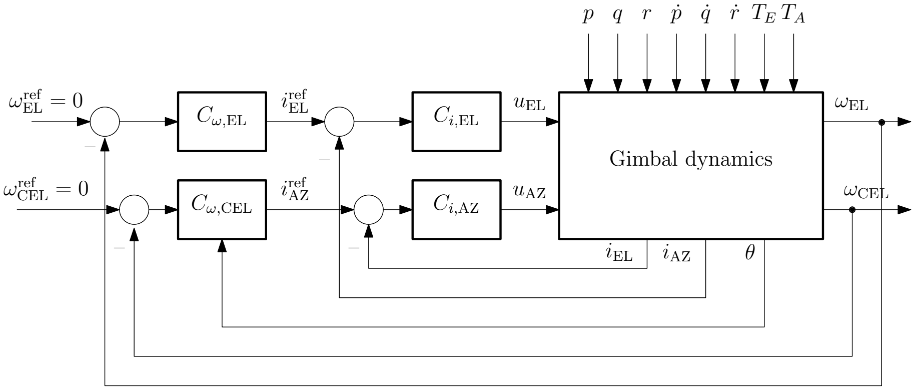

p q r $\dot{p}$ $\dot{q}$ $\dot{r}$ T_E T_A
$\omega_{\mathrm{EL}}^{\mathrm{ref}} = 0$
$\omega_{\mathrm{CEL}}^{\mathrm{ref}} = 0$
$C_{\omega,\mathrm{EL}}$
$C_{\omega,\mathrm{CEL}}$
$i_{\mathrm{EL}}^{\mathrm{ref}}$
$i_{\mathrm{AZ}}^{\mathrm{ref}}$
$C_{i,\mathrm{EL}}$
$C_{i,\mathrm{AZ}}$
u_{EL}
u_{AZ}
Gimbal dynamics
ω_{EL}
ω_{CEL}
i_{EL}
i_{AZ}
θ
$-$

Figure 4.2: Inertial stabilization. Two independent (decoupled) SISO feedback loops, one for each rate gyro attached to the body of camera, $\omega_{\mathrm{EL}}(=\omega_{Ey})$ and $\omega_{\mathrm{CEL}}(=\omega_{Ez})$. The cross-elevation stabilizing controller must contain secant gain correction $1/cos(\theta)$. The disturbing variables are the carrier roll, pitch and yaw rates, p, q, r, respectively, their derivatives and external torques around the two motor axes. The innermost current loops are also depicted.

(thoroughly studied in [1]), may be explained also by taking (2.14), (2.12) and (2.17) to obtain

$$\omega_E^E = R_A^E \omega_E^A = \begin{bmatrix} c_\theta(pc_\psi + qs_\psi) - s_\theta(r+\dot{\psi}) \\ -ps_\psi + qc_\psi + \dot{\theta} \\ s_\theta(pc_\psi + qs_\psi) + c_\theta(r+\dot{\psi}) \end{bmatrix} = \begin{bmatrix} \omega_{Ex} \\ \omega_{Ey} \\ \omega_{Ez} \end{bmatrix}. \tag{4.1}$$

It is clear that to achieve the line-of-sight stabilization, both ω_{Ey} and ω_{Ez} must be zero. For this purpose $\dot{\theta}$ and $\dot{\psi}$ are available. While the impact of $\dot{\theta}$ on ω_{Ey} is direct, the impact of $\dot{\psi}$ on ω_{Ez} comes through the term $\cos(\theta)$. Block diagram of the control structure for cross-elevation axis is shown in Fig. 4.3.

Fig. 4.3 also represents an important fact how disturbing rotational movement of the carrier $[p, q, r]$ enters the cross-elevation loop. It is only the r component that enters the loop exclusively via friction in the azimuth joint (and via back emf, which is not shown in figure). The r component is thus very well rejected by the mass stabilization principle, it corresponds to disturbance ω_s depicted in Fig. 3.6 and Fig. 3.7.

The remaining components p, q enter the loop directly via the corresponding projection. When the elevation angle θ is not zero, these signals always enter the loop directly, and may be rejected only by applying feedback control action with always limited bandwidth. This corresponds to disturbance ω_d depicted in Fig. 3.6 and Fig. 3.7.

In case of the elevation joint there are only two disturbing angular rates p and q that enter the loop, and they do so indirectly via friction in elevation joint. This fact guarantees that the line of sight is vertically (means the component ω_{Ey}) very well stabilized. The block diagram for elevation axis is in Fig. 4.4.

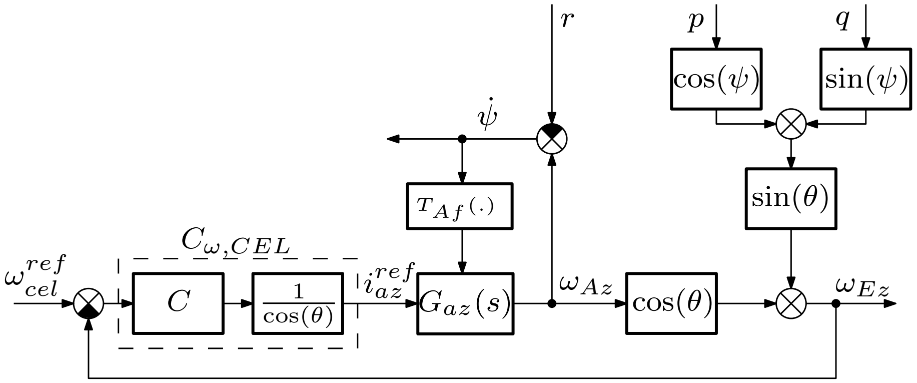


Figure 4.3: Inertial stabilization for the cross-elevation. The image is redrawn from [1]. $G_{az}(s)$ corresponds to azimuth part od gimbal dynamics with the current loop already closed.

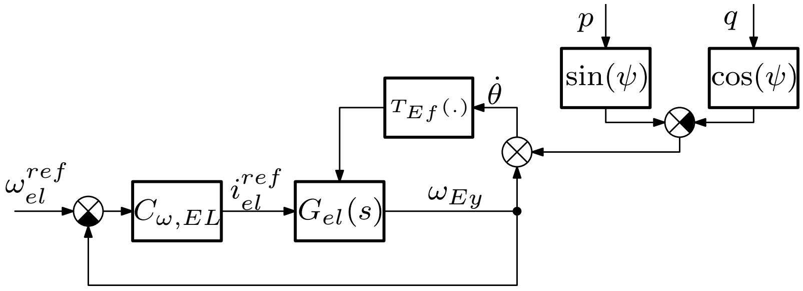


Figure 4.4: Inertial stabilization for the elevation. $G_{el}(s)$ corresponds to the elevation part of the gimbal dynamics with the current loop already closed.

Even though there is some gyroscopic coupling between the two axes[1], its influence is not worth designing a MIMO rate controller. The neglected gyroscopic effect can be cast as yet another external disturbing torque and as such left to the rate controller to suppress.

4.1.1 LOS stabilization experiment with H240 platform

In order to demonstrate the functionality of the line-of-sight stabilization and to experimentally prove the suitability of the proposed decoupled control structure, the two experiments were performed.

Fig. 4.5 shows results of the experiment of stabilizing the disturbing base (rotational) motion p in the elevation axis (see Fig. 4.4 for the schematic diagram of the setup). Notice that the camera movement was very well stabilized (less than -20dB) and the integral of the signal from gyroscope placed onto the camera ($\omega_{Ey} = \omega_{EL}$) did not move more than fractions of a degree.

[1] See the chapter 2.2, [11] and [12] for full models or [14] for the simplified version when the base is still.

The other experiment (see Fig. 4.6) for the cross elevation axis does not show so gratifying results. As expected, the rejection of disturbing motion of the base that enters the loop directly is well rejected only at very low frequencies. Disturbance q enters the loop via term $\sin 55° \cdot \cos 0° = 0.82$, so that at the time $25 - 30\,\mathrm{s}$ it is almost not rejected at all. The schematic diagram for the setup of this experiment is visualized in Fig. 4.4.

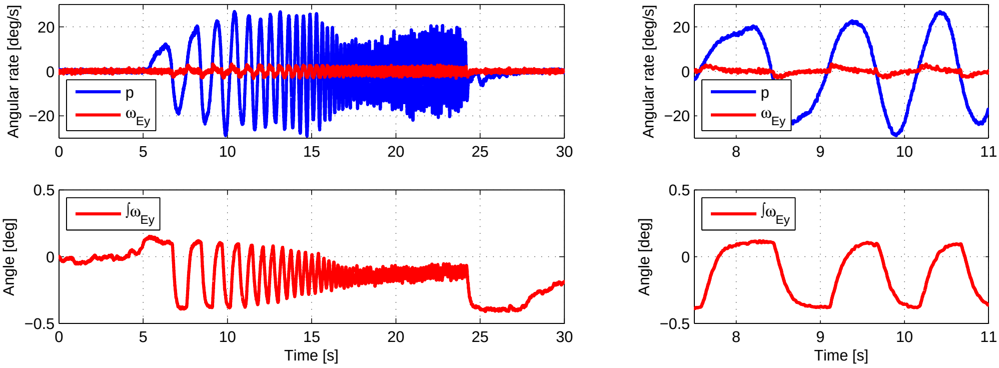


Figure 4.5: Line-of-sight stabilization experiment for elevation axis of H240 platform. The input disturbance for the experiment is a rotation of the platform base in p axis. The disturbance is very well rejected.

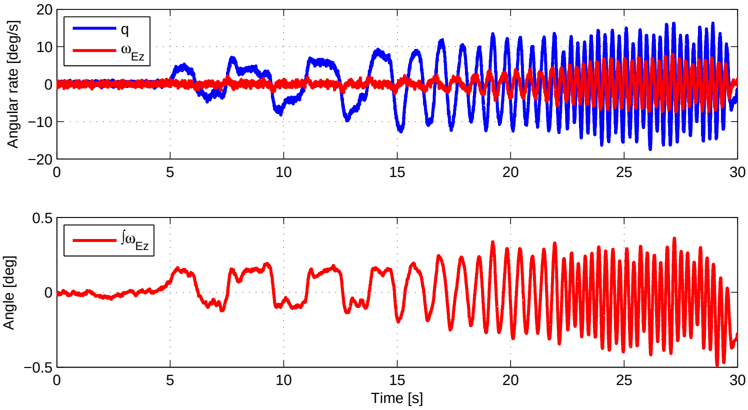


Figure 4.6: Line-of-sight stabilization experiment for cross-elevation axis of H240 platform. The azimuth of the platform was set up to $\psi = 0°$, the elevation to $\theta \approx 55°$. The input disturbance for the experiment is a rotation of the platform base around q axis. Because the disturbance enters the loop directly, it is well rejected only at very low frequencies.

4.2 SINGLE-AXIS DUAL-STAGE INERTIAL STABILIZATION

In this section, control algorithms for already introduced single-axis dual-stage benchmark model that consists of two coaxial motorized gimbals stabilizing the angular motion of an optical payload around a single axis (introduced in the section 2.4) will be presented. After a brief description of the feedback control configuration, the already classical $\mathcal{H}_\infty$ control design methodology is followed.

It will be shown, however, that the particular control problem naturally invokes a requirement on a structure of the MIMO controller. By the term *structure* it is meant that some of the transfer functions in the controller transfer function matrix are fixed to zero. Moreover, the order of the controller can also be fixed before the optimization starts. Such design constraints are tremendously difficult to take into account in an optimization-based control design because they render the set of admissible controllers non-convex. One of the contributions, apart from the discussion of the control configuration as such, is a detailed documentation of a practical application of the two concrete numerical solvers for fixed-structure and fixed-order $\mathcal{H}_\infty$ control design. These solvers constitute a most practical exploitation of the research in numerical techniques for non-convex non-smooth optimization. Namely,

HIFOO solver for Matlab, which was for the first time presented to the community by [39] and then by [40] and [41]. The acronym stands for *H-infinity fixed-order optimization.* The theory behind HIFOO is described in [42]. The software is available for free download and is distributed as open-source within GNU GPL 3.0 license. The openness of the code encourages other researchers in providing their own modifications and extensions; for example, [43] provide an extension to a discrete-time case.

Hinfstruct solver, which was for the first time presented to the community by [44] and is now part of Robust Control Toolbox for Matlab. The theory behind is described in [45]. The solver is distributed commercially (within the toolbox) as a binary code.

Since the 2012b release of Matlab, the Robust Control Toolbox has contained a new tool called SYSTUNE, which saves the user from the necessity to formulate his or her control design problem in the standard LFT framework. For the comparison purposes the pure Hinfstruct solver is used.

The content of this section is mainly based on the conference paper [7] and journal paper [8] by the author.

4.2.1 Feedback control configuration

The key task for the control system is to regulate the inertial angular rate ω_1 of the payload to a desired value and keep it there irrespectively of the disturbing rotational motion of the carrier given by the rate ω_c. When there is no request to reorient (repoint) the payload, the task is to keep the payload still, that is, $\omega_1^{\mathrm{ref}} = 0$, while keeping the angular misalignment e_φ between the inner and the outer gimbals in the middle of the range, that is $e_\varphi = 0$ so as not to hit the mechanical limits.

Now, which motor is responsible for what? As a matter of fact, both motors can participate in both tasks leading to a full MIMO controller, but from the viewpoint of simplicity of implementation and fine-tuning, a decision was made to assign the "roles" to the motors. Regulation of the angular rate ω_1 or tracking of its reference value ω_1^{ref} is achieved by commanding the voltage u_1 by the controller K_1 and applying it to the VCM. This is the *inertial angular velocity control loop.* The *angular deviation loop* measures the angle e_φ and regulates this value to zero by commanding the voltage u_2 by the controller R_2 and applying it to the rotary DC motor. Block diagram is in Fig. 4.7.

Furthermore, a major improvement in regulating the angle e_φ can be achieved by introducing a feedforward term F from the reference rate ω_1^{ref} to the voltage u_2. The presence of this term can be easily justified. A classical feedforward controller for motion control applications differentiates the reference angle φ_1 twice and scales the result appropriately to estimate the needed torque. This is then transformed into the applied voltage u_2. But here φ_1 is not measured. Instead, the dynamics from ω_1^{ref} to φ_1 is known (determined by the already designed controller K_1). The feedforward controller therefore needs to include this model in order to act directly on ω_1. The output of the filter is applied to the outer stage (DC rotary motor) so that the misalignment angle e_φ is kept small. In other words, when there is a nonzero inertial rate to be tracked by the payload, it is not only the VCM motor that starts rotating but the rotary DC motor also responds immediately, it does not wait for the error e_φ to build up.

4.2.2 Design of a structured MIMO low-order controller

The popular control design methodology based on H_∞ optimization is formulated as the minimization of

$$\min_{K\ \mathrm{stabilizing}} \|F_l(P,K))\|_\infty, \tag{4.2}$$

where F_l stands for a lower linear fractional transformation (LFT), K is a stabilizing controller and P is the augmented plant composed of both the original plant dynamics and the artificial weighting filters used to express the control objectives. A wealth of various software packages exist for solving this optimization problem. HIFOO and Hinfstruct stand out by being able to fix the order and the structure of the controller.

Augmented plant structure

The augmented plant model is in Fig. 4.8. There are two exogenous variables: ω_1^{ref}, and ω_c, and four regulated variables: $z_{e_{\omega_1}}$, z_{e_φ}, z_{u_1} and z_{u_2}, which are to be minimized (in the sense of a signal 2-norm) by the feedback controller K. The variable $z_{e_{\omega_1}}$ specifies the weighted error between the required reference rate ω_1^{ref} and the measured rate ω_1 from the

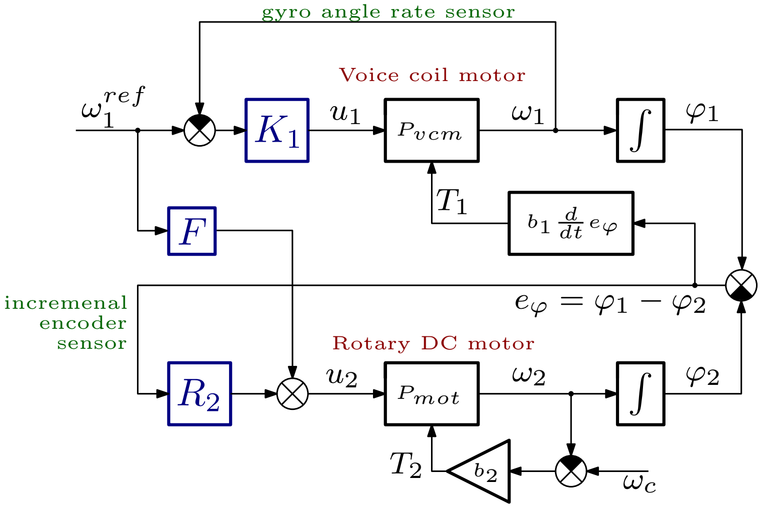


Figure 4.7: Feedback control configuration. It is only for the sake of simplicity of the figure that the reaction torque generated by the VCM and applied to the rotary motor is not depicted here and neither are the back emf voltages, which are derived from the relative and not the inertial rates. Furthermore a friction b_2 represents together both frictions b_p and b. Detailed structure of the model can be understood directly from equations (2.36)–(2.45).

gyro. Similarly, the variable z_{e_φ} penalizes the deviation of the angular misalignment e_φ. The variables z_{u_1} and z_{u_2} express the actuator effort.

The controller K receives the measurements from two sensors (the MEMS gyro giving the inertial angular rate ω_1 and incremental encoder giving the angle e_φ) and it also receives the reference value for the inertial rate ω_1^{ref}. It computes two voltages u_1 and u_2. It should be emphasized that this is just a reformulation of the configuration described in the section 4.2.1. That is, the controllers K_1, R_2 and F form the components $K_{(1,2)}$, $K_{(2,3)}$ and $K_{(2,1)}$, respectively, with the remaining sub-blocks of the compound controller K set to zero. Capability of handling such structural constraints was exactly the reason for adoption of HIFOO and Hinfstruct here.

One possibility to set the components $K_{1,1}$, $K_{2,2}$ and $K_{1,3}$ of the controller's transfer function matrix to zero is to consider the controllers K_1, R_2 and F connected in parallel. The resulting state space model is characterized by matrices A, B, C, D

$$A = \begin{bmatrix} A_{n_1\times n_1} & 0 & 0 \\ 0 & A_{n_2\times n_2} & 0 \\ 0 & 0 & A_{n_3\times n_3} \end{bmatrix} B = \begin{bmatrix} B_{n_1\times 1} & 0 & 0 \\ 0 & B_{n_2\times 1} & 0 \\ 0 & 0 & B_{n_3\times n_1} \end{bmatrix} \tag{4.3}$$

$$C = \begin{bmatrix} 0 & C_{1\times n_2} & 0 \\ C_{1\times n_1} & 0 & C_{1\times n_3} \end{bmatrix} D = \begin{bmatrix} 0 & D_{1\times 1} & 0 \\ D_{1\times 1} & 0 & D_{1\times 1} \end{bmatrix}, \tag{4.4}$$

where n_1, n_2 and n_3 are orders of controllers K_1, R_2 and F respectively.

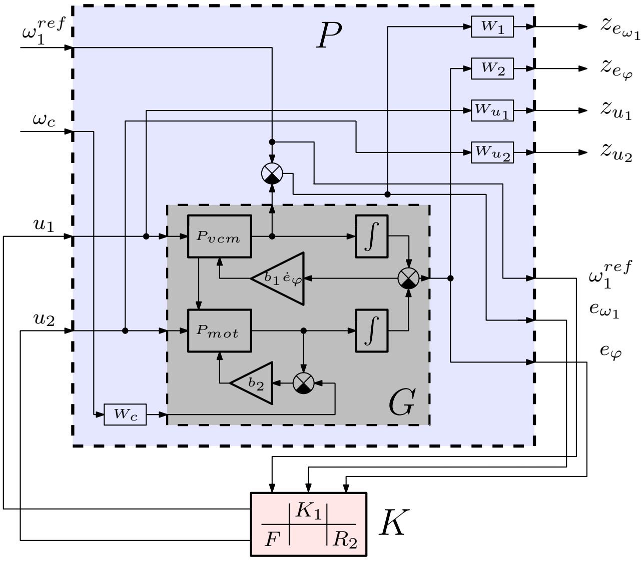


Figure 4.8: LFT of an augmented system and a structured controller. The structuring of the controller K into sub-blocks corresponds to the practical feedback configuration discussed in the section 4.2.1.

Weighting filters selection

The bandwidth of the reference angular rate that is to be tracked should certainly be below the gyro's bandwidth f_{bw}. Adding a requirement of the precise tracking at low frequencies with the error as low as $-60\,\mathrm{dB}$ specifies the filter W_1

$$W_1 = \frac{0.2s + 50}{s + 0.05}.$$

The filter W_2 is selected such that effect of disturbance ω_c on the misalignment e_φ is minimized and kept below 0.09 rad, which is the mechanical limit of the angle e_φ over all frequencies. Moreover the low frequency disturbances and especially the steady state disturbance should by attenuated by more than $-60\,\mathrm{dB}$. These requirements are expressed by the filter

$$W_2 = \frac{10s + 100}{s + 0.025}.$$

To improve the attenuation of the effect of the disturbance ω_c in the inertial velocity loop up to frequency 1 Hz, where the unwanted oscillation of the carrier is expected, a filter

$$W_c = \frac{2s + 3.6}{s + 3.6}$$

is appended at the ω_c's input.

As both input voltages are normalized, a simple choice of the filter penalizing the actuator effort was made

$$W_{u_1} = W_{u_2} = 1.$$

4.2.3 Controllers designed using HIFOO and Hinfstruct

Since both HIFOO and Hinfstruct rely on the algorithms that search only for local minima, it is necessary to specify not only the augmented plant P and the required structure of the controller's state-space matrices A, B, C, D (as in (4.3) and (4.4) with $n_1 = 1$, $n_2 = 2$ and $n_3 = 2$), but also an "initial controller". The task of searching for the structured H_∞ optimal controller may be then interpreted as that of refining the already known controllers K_1 and R_2 (found using some structure ignoring techniques) and designing the controller F. Here two PI controllers $K_1 = (s + 100)/s$ and $R_2 = 10(s + 0.5)/s$ were used to build the initial compound controller (A_0, B_0, C_0, D_0). The feedforward controller F was initially not used at all. In order to provide more freedom to the optimization routine, the orders of designed controllers were increased comparing to initial controllers

$$A_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad B_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 8 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.5}$$

$$C_0 = \begin{bmatrix} 0 & 12.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.5 & 0 \end{bmatrix} \qquad D_0 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 10 \end{bmatrix}. \tag{4.6}$$

The state-space realization of the HIFOO controller is

$$A_1 = \begin{bmatrix} -4.16 & 0 & 0 & 0 & 0 \\ 0 & -14.79 & 5.89 & 0 & 0 \\ 0 & -6.51 & -5.83 & 0 & 0 \\ 0 & 0 & 0 & -32.03 & -322.2 \\ 0 & 0 & 0 & -45.1 & -454.4 \end{bmatrix} \qquad B_1 = \begin{bmatrix} -14.09 & 0 & 0 \\ 0 & 5.49 & 0 \\ 0 & -2.6 & 0 \\ 0 & 0 & 63.54 \\ 0 & 0 & 63.21 \end{bmatrix} \tag{4.7}$$

$$C_1 = \begin{bmatrix} 0 & 7.09 & 1.56 & 0 & 0 \\ -0.03 & 0 & 0 & -30.53 & -323.3 \end{bmatrix} \qquad D_1 = \begin{bmatrix} 0 & 1.13 & 0 \\ 0.31 & 0 & 65.93 \end{bmatrix}. \tag{4.8}$$

The state-space realization of the Hinfstruct controller is

$$A_2 = \begin{bmatrix} -2.32 & 0 & 0 & 0 & 0 \\ 0 & -14.29 & 10.49 & 0 & 0 \\ 0 & -9.36 & 0.87 & 0 & 0 \\ 0 & 0 & 0 & -484.8 & 12.24 \\ 0 & 0 & 0 & -103 & 2.56 \end{bmatrix} \qquad B_2 = \begin{bmatrix} -0.9 & 0 & 0 \\ 0 & -7.31 & 0 \\ 0 & 1.4 & 0 \\ 0 & 0 & -116.5 \\ 0 & 0 & -30.43 \end{bmatrix} \tag{4.9}$$

$$C_2 = \begin{bmatrix} 0 & -3.89 & 0.89 & 0 & 0 \\ -1.07 & 0 & 0 & 651.5 & -20.82 \end{bmatrix} \qquad D_2 = \begin{bmatrix} 0 & 1.129 & 0 \\ 0.1386 & 0 & 179.4 \end{bmatrix}. \tag{4.10}$$

Apparently the structure in the matrices was preserved. Bode plots for the new controllers and the classically tuned PI controllers designed for the two decoupled subsystems are in Fig. 4.9.

Closed-loop $\mathcal{H}_\infty$ norms

A possible performance measure of the designed controllers is their achieved optimization cost (4.2). The $\mathcal{H}_\infty$ norms of the corresponding closed-loop systems calculated for all the designed controllers are in Table 4.1. It is no surprise that the full $\mathcal{H}_\infty$ controller achieves

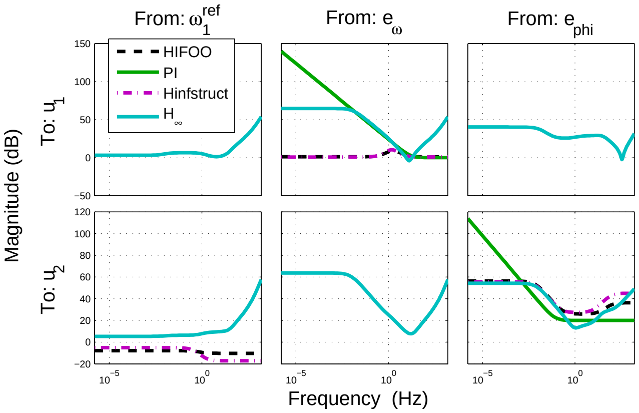


Figure 4.9: Magnitude Bode plots for the structured MIMO controllers designed using HIFOO and Hinfstruct and comparison with the PI decoupled controllers. The structured controllers contains an extra feedforward term $K_{(2,1)} = F$.

Table 4.1: Resulting $\mathcal{H}_\infty$ norms according to (4.2)

Controller	$\mathcal{H}_\infty$ *norm*
Classical $\mathcal{H}_\infty$	1.1929
HIFOO	1.3408
Hinfstruct	1.2990
PI controller (initializing)	7.4378

the lowest norm and the HIFOO and Hinfstruct deliver only suboptimal solutions. Notice also that no controller fulfills completely the requirements specified by the filter $1/W_2$ (see Fig. 4.10), which is evidenced by the fact that the norm is greater than 1.

Since HIFOO uses randomized starting points, the same results are not obtained every time HIFOO is run. By default, HIFOO uses three randomized starting points. For the purpose of comparison, Hinfstruct was set up so that three random starting points are used together with the mentioned initial point

```
options = hinfstructOptions('RandomStart',3) .
```

To provide more accurate and unbiased comparison of the computational burden of HIFOO, Hinfstruct and the classical Hinfsyn routines, the optimizations were run ten times in the loop. The achieved results with corresponding average times and average norms are listed in the Table 4.2. HIFOO turns out approximately 8 times slower than Hinfstruct when using the default setup.

According to [40], the HIFOO algorithm may be speeded up upon installation of `linorm` function of the SLICOT package. However, the (old) `linorm` of Slicot is now slower

Table 4.2: Average results obtained from repeated runs (10x) for all methods

Controller	$\mathcal{H}_\infty$ *norm*	*Time* [s]	*Note*
Classical $\mathcal{H}_\infty$	1.1929	1.24	Hinfsyn using an LMI solver
HIFOO	1.3640	68.68	no SLICOT, no quadprog, 4 points
Hinfstruct	1.2951	8.64	4 points
PI controller (init.)	7.438	-	initial controller

then the (new) `norm(*, inf)` function of Matlab[2]. All the code was designed in Matlab 2012a with HIFOO version 2.0 and without SLICOT and without any `quadprog` (neither from MOSEK nor from Optimization Toolbox). The code has been made downloadable at `http://www.mathworks.com/matlabcentral/fileexchange/42845`.

Closed loop simulations

The performance of the two designed controllers given by (4.7) – (4.8), (4.9) – (4.10) and the initial controller (4.5) – (4.6) will now be evaluated by analyzing the closed-loop transfer functions. Fig. 4.10 shows the closed-loop transfer functions from the exogenous inputs (disturbance and reference) to the regulated (error) signals. The requirements on the reference tracking were successfully accomplished with a 10 Hz bandwidth. The disturbance transfer to the inner stage (from ω_c to ω_1) was attenuated to -20 dB in the worst case. The misalignment angle e_φ is very well regulated too because the worst case value -30 dB suffices to avoid the $\pm 5^\circ$ mechanical limit (assuming the amplitude of the disturbance is bounded by 1 rad/s).

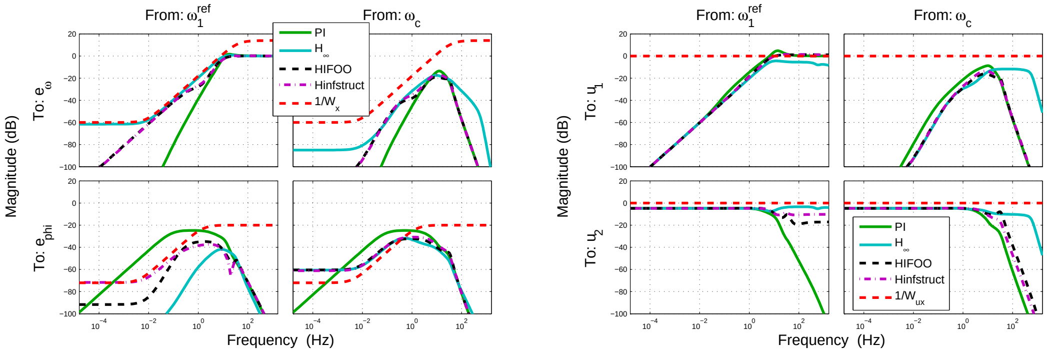
Fig. 4.10 then follows with the closed-loop transfer functions from the disturbance and the reference signal to the control signals (actuator outputs). All of these frequency responses are below the 0 dB value except for the top left figure. This means that the controller signal saturation may occur in case of the PI controller. This will happen only for reference signals changing faster than ≈ 0.5 rad/s since the overshoot is less than 6 dB.

Finally, Fig. 4.11 complements frequency response plots with the responses to step changes in the reference and disturbance.

4.2.4 Experimental verification

The experimental dual-stage benchmark system allows a verification of designed controllers directly from Matlab and Simulink using Real-Time Toolbox and the MFC624 general-purpose DAQ card, both produced by Humusoft. MFC624 card contains 50 MHz counters which may be employed to generate 50 kHz PWM signal with a 10-bit resolution, which is sufficient for this application. Output PWM signals drive the motors via H-bridges: L6234 for the VCM and L6201PS for the rotary DC motor. Both have sufficient switching frequency

[2] This information was obtained from one of the anonymous reviewers during the review process of the paper [8]



(a) From the disturbances and references to the error signals.

(b) From the disturbances and references to controller actions.

Figure 4.10: Magnitude Bode plots for the closed-loop transfer functions.

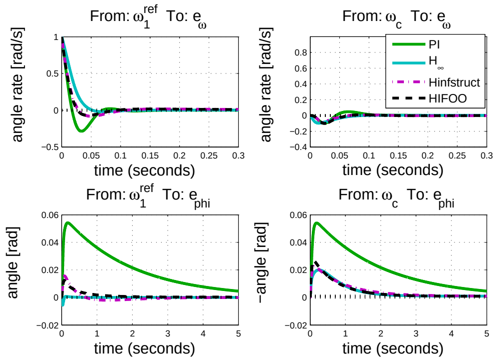
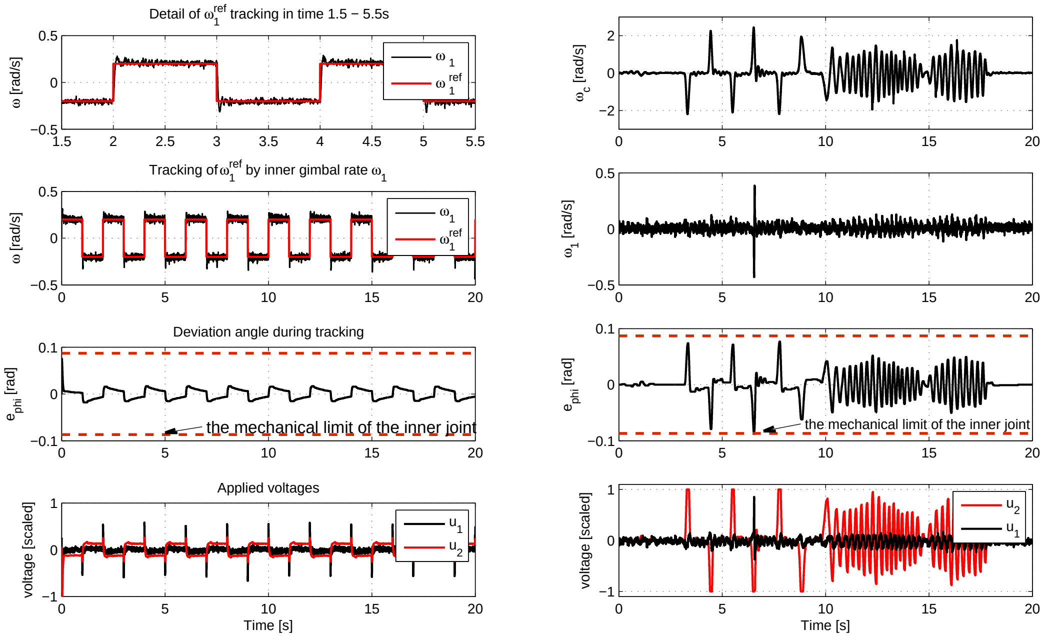


Figure 4.11: Closed-loop step responses. The response of the angle e_φ is improved compared to the initial PI controller. Comparing the bottom graphs shows the usefulness of the feedforward term F — the reference signal ω_1^{ref} viewed as a disturbance is now better rejected by the outer joint. Note the minus sign at the y-label.

50 kHz resp. 100 kHz in case of L6201PS. The sampling rate of the control algorithm was set to 1 kHz. In total two experiments were conducted with a controller designed by HIFOO.

In Fig. 4.12 (a), a reference tracking was demonstrated. During the fast rising edge of the steps, the angle e_φ is increased at first and then slowly reaches zero. It always remains inside the ±5° region as desired.

In Fig. 4.12 (b), a satisfactory disturbance rejection is shown. System was exposed to disturbing rotational motion of the base, with the (recorded) angular rate ω_c of the amplitude up to 2 rad/s. The inner stage (driven by the voice coil motor) has attenuated the influence of this disturbance sufficiently, as the graph of ω_1 shows. An exception is one



(a) The experimental results of reference ω_1^{ref} tracking.

(b) The experimental result of ω_c rejection. At time 6.6 s the misalignment angle e_φ has reached its mechanical limit and the disturbance passed into ω_1 as well. In cases where ω_c was higher than the scaled 1 rad/s, the saturation of u_2 was observed.

Figure 4.12: Real-time experiments performed with dual-stage platform.

short moment is at time 6.6 s when the angle e_φ reached its mechanical limit.

4.2.5 Conclusion

The section described a practical control design based on $\mathcal{H}_\infty$-minimization under constraints on the order and the structure of a MIMO controller. Two popular numerical solvers have been used to accomplish it—HIFOO and Hinfstruct.

In particular, a single-axis electromechanical motion control system with two motorized stages was employed as a benchmark system. The system offers two variables for measurement and two independent actuators, hence, the problem can be tackled either by using classical techniques based on cascading SISO loops, or direct design of a MIMO controller. Combining benefits of both approaches leads to a structured MIMO controller design.

Numerical simulations and laboratory experiments with the computed structured controller support the conclusion that the degradation of the closed-loop performance was negligible compared to the full (unstructured) $\mathcal{H}_\infty$-(sub)optimal MIMO controller; the simplified implementation and finetuning present a major benefit of the proposed methodology. The simultaneous design of the two loops (and one cross-coupling feedforward term) offers a significant convenience for the designer compared to the sequential character of the classical decoupled/cascade design.

The two numerical solvers provided almost identical results with Hinfstruct achieving a bit better $\mathcal{H}_\infty$ performance and significantly shorter computational time. On the other hand, HIFOO as an open-source software offers a possibility to learn from the code and possibly modify it.

One practically important issue that was not addressed is the issue of hitting the mechanical stops when the deviation between the inner and the outer stages reaches its limits of $\pm 5°$. Unless some bound on the magnitude of the disturbance (the motion of the carrier) is guaranteed, this hitting could happen and the controller must be prepared for such event. Without any measures, the inner stage controller will then keep pushing against these mechanical stops (with the current being burnt in the motor windings). Simply switching the actuator off once it touches the mechanical bounds leads to chattering-like behavior. This is neither the situation with a bounded actuator, which could be solved (albeit heuristically) with an antiwindup scheme, nor a constrained-state problem, which is commonly solved with MPC-like techniques. Here the state (the relative angle) is constrained mechanically. No measures were applied in this particular laboratory experiment, although some heuristic techniques is applied a real application (four-gimbal dual-stage platform): the controller action is modulated by a constant that approaches zero as the angle approaches its mechanical bound.

4.3 FOUR-JOINT DUAL-STAGE STABILIZATION

In this section, control algorithms for the line-of-sight stabilization of the full four-gimbal dual-stage platform are presented. Mathematical model of the four-gimbal dual-stage platform was described in section 2.5. This section uses the formalism introduced therein. The control structure for the single-axis dual-stage line-of-sight stabilization was studied in previous section. It was shown, see Fig. 4.7, that line-of-sight stabilization may be achieved using a simple control structure with two decoupled control loops. That is

- the inner angular-rate loop is responsible for the fine stabilization by measuring the inertial angular rate of the payload using a gyro (controlled by K_1 in Fig. 4.7)
- the outer angle loop is responsible for preventing the misalignment angle from reaching its mechanical limit (in Fig. 4.7 controlled by controller R_2).

This section documents the necessary steps that are required to extend the simple single-axis structure to the full four-gimbal dual-stage stabilization.

In order to make the line of sight insensitive to external disturbances the two inertial angular rates that are responsible for the line-of-sight movement must be kept zero (or must follow reference rate obtained from higher level control loop). Namely it is,

- the inertial angular rate ω_{Iy} of the inner frame, corresponding to up-down movement in the camera frame,
- the inertial angular rate ω_{Iz} of the payload around its own vertical axis, also denoted the camera cross-elevation rate, corresponding to left-right movement in the camera frame.

By rebuilding the forward recursion of the Newton-Euler method such a way that inertial angular rates ω_{Az}, ω_{Ey} and ω_{Uy} disappear, the equation (4.12) is obtained. The inertial angular rate of the inner frame (ω_I^I) is expressed only using base inertial angular rates $\omega_B^B = [p, q, r]^T$ and all four joint variables (ψ, θ, ϵ and γ).

$$\omega_I^I = \begin{bmatrix} \omega_{Ix} \\ \omega_{Iy} \\ \omega_{Iz} \end{bmatrix} = R_U^I \left(R_E^U \left(R_A^E \Big(R_B^A (\omega_B^B + z_B \dot{\psi}) + y_A \dot{\theta} \Big) + y_U \dot{\epsilon} \right) + z_I \dot{\gamma} \right) = \tag{4.11}$$

$$= \begin{bmatrix} -c_\gamma (s_\epsilon s_\theta - c_\epsilon c_\theta)(p c_\psi + q s_\psi) - c_\gamma (s_\epsilon c_\theta + c_\epsilon s_\theta)(\dot{\psi} + r) + s_\gamma (\dot{\epsilon} + \dot{\theta} + q c_\psi - p s_\psi) \\ +s_\gamma (s_\epsilon s_\theta - c_\epsilon c_\theta)(p c_\psi + q s_\psi) + s_\gamma (s_\epsilon c_\theta + c_\epsilon s_\theta)(\dot{\psi} + r) + c_\gamma (\dot{\epsilon} + \dot{\theta} + q c_\psi - p s_\psi) \\ \dot{\gamma} + (s_\epsilon c_\theta + c_\epsilon s_\theta)(p c_\psi + q s_\psi) - (s_\epsilon s_\theta - c_\epsilon c_\theta)(\dot{\psi} + r) \end{bmatrix} . \tag{4.12}$$

As it was already stated in this section to achieve perfect line-of-sight stabilization, both ω_{Iy} and ω_{Iz} must be perfectly zero. However, in contrast with double gimbal, here there are in total four gimbals available to achieve it (with joint velocities $\dot{\psi}$, $\dot{\theta}$, $\dot{\epsilon}$ and $\dot{\gamma}$).

Which gimbal is though responsible for what purpose? The simplest solution is to preserve the control structure from the single-axis dual-stage stabilization. Using that approach the two decoupled dual-stage control loops are developed for both rates ω_{Iy} and ω_{Iz}.

To be more concrete, the single-axis dual-stage scheme from Fig. 4.7 should be implemented twice — for ω_{Iy} and for ω_{Iz}. The angular rates $\dot{\epsilon}$ and $\dot{\gamma}$ are used for stabilization using rate controllers K_{Iy} and K_{Iz} respectively, while angular rates $\dot{\psi}$ and $\dot{\theta}$ are used for controlling angles γ and ϵ to remain inside their mechanical limits using controllers R_ϵ and R_γ.

To explore such control structure in detail, the inner inertial-stabilization loops are studied first followed by providing the complete control structure in the two axes.

4.3.1 Stabilization of inertial angular rates

From (4.12) it is clear that the impact of $\dot{\gamma}$ on ω_{Iz} is direct. It is thus convenient to close the control loop so as suggested in Fig. 4.13. The rate controller $K_{Iz} = C_{Iz}$ is taking ω_{Iz} measured by the rate gyroscope (MEMS or laser) and uses this signal together with the required reference inertial angular rate ω_{Iz}^{ref} to control the required torque (current) in voice coil motor in γ joint.

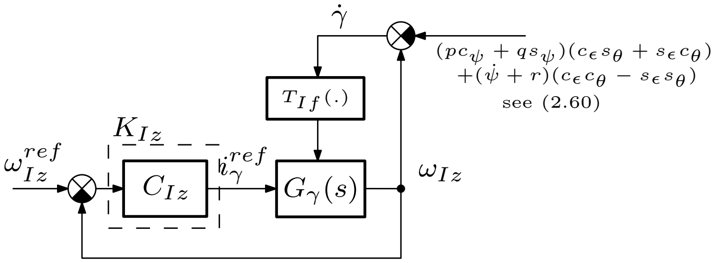

$\dot{\gamma}$
$(pc_\psi + qs_\psi)(c_\epsilon s_\theta + s_\epsilon c_\theta)$
$+(\dot{\psi} + r)(c_\epsilon c_\theta - s_\epsilon s_\theta)$
see (2.60)
$T_{If}(.)$
K_{Iz}
ω_{Iz}^{ref}
C_{Iz}
i_γ^{ref}
$G_\gamma(s)$
ω_{Iz}

Figure 4.13: Schematic diagram of the inner stabilization loop for cross-elevation axis — stabilizing ω_{Iz}. The scheme shows the way how disturbing angular rates are entering the control loop — exclusively via friction $T_{If}(\dot{\gamma})$. The significant disturbance rejection is ensured by this control structure (see the discussion about possible disturbances in the control loop and their impact on the rejecting performance in section 3.3).

It is also instructive to examine what kind of disturbances enter the control loop and how. Disturbing base angular rates enter the control loop exclusively via the friction $T_{If}(\dot{\gamma})$ which ensures the significant disturbance rejection. The expression for the disturbing friction may be easily obtained from (2.60) since it is a function of $\dot{\gamma}$. Notice also that the part of the disturbance is the motion of the azimuth joint $\dot{\psi}$. This means that every movement in azimuth joint introduces some disturbance in the inner gimbal even if the platform is standing at rest. The movement in two remaining joints $\dot{\theta}$ and $\dot{\epsilon}$ does not enter the loop as disturbance because they act in the perpendicular axis.

The impact of $\dot{\epsilon}$ on ω_{Iy} is not direct as in ω_{Iz} case. It is clear from (4.12) that ω_{Iy} may be affected by $\dot{\epsilon}$ only via term $\cos\gamma$. To close the control loop, the inverse of this term should included in the controller $K_{Iy} = \frac{1}{\cos\gamma} C_{Iy}$ as suggested in Fig. 4.14. Since the angle γ is mechanically limited to be less than ±5°, it is feasible to neglect this term in the controller (the gain error is less than 0.004).

Controlling the angular rate ω_{Iy} using the inner elevation joint rate $\dot{\epsilon}$ (which does not

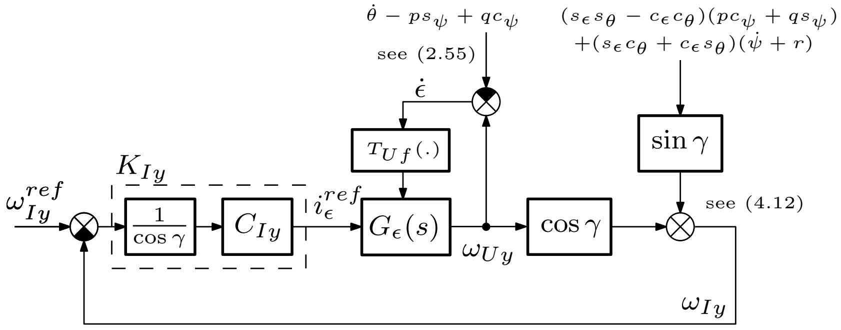


Figure 4.14: Schematic diagram of the inner stabilization loop — stabilizing ω_{Iy}. The scheme clearly shows that ω_{Iy} may be affected only by the projection via $\cos\gamma$ using the inner-elevation joint. Comparing to the previous scheme in Fig. 4.13, here appears an extra disturbing term that is entering the loop directly. The exact enumeration of this disturbance is possible by substituting ω_{Uy} from (2.55) into the third row in (4.12).

act directly in this axis), brings about an issue with disturbances entering the control loop directly and not only via friction. This behavior was already presented in the double-gimbal platform — the disturbance was entering the loop via the $\sin\theta$ term . Here the situation is the same with only exception that disturbance is entering the loop via the term $\sin\gamma$. Since γ is limited to be $\pm 5°$ at most, its sine is limited to be $|\sin(5°)| < 0.09 \approx -20\,\text{dB}$ and thus this disturbance is already well rejected. The remaining part of the disturbance enters the loop only via friction $T_{Uf}(\dot{\epsilon})$ so that it is well rejected too. The exact enumeration of both (direct and via friction) disturbances may be obtained from (4.12). It is schematically depicted in Fig. 4.14.

The behavior described above — the disturbances that enter the loop directly are well attenuated while the remaining disturbances enter the loop via friction — represents a major advantage of the four-gimbal dual-stage stabilization configuration. Compared to the double gimbal, this mechanical configuration promises better disturbance rejection (the insufficient stabilization of the cross-elevation axis for the double gimbal was shown in Fig. 4.6).

4.3.2 Closing the outer loops — the full dual-stage stabilization

In order to provide a complete description of the control structure of the line-of-sight stabilization using four-gimbal dual-stage platform this section focuses on description of both outer control loops. The outer loops are taking care of that angles γ and ϵ are not reaching their mechanical limit. Exactly as in previous section also here the problem is studied separately for the inner-elevation (ω_{Iy}) and cross-elevation (ω_{Iz}) axes.

The control scheme for the inner-elevation axis is shown in Fig. 4.15. The scheme keeps the notation of the rate controller labeled as K and misalignment angle labeled as R from the previous section devoted to the single-axis dual-stage problem. The scheme represents the extended version of the scheme from Fig. 4.14. Notice that since both joints (ϵ and θ) operate around the same axis $y_E = y_U$, the impact of θ on ϵ is linear. This fact is also seen in (2.55). To control ϵ one may use $\dot{\theta}$, while the remaining variables (ω_{Uy}, p and q) are viewed as disturbances. The controller that is taking care of the angle ϵ is labeled as R_ϵ. Notice that by neglecting the disturbance coming through the $\sin\gamma$, the only disturbance that is entering the control loop is $ps_\psi - qc_\psi$. It does so by the same way as disturbance ω_C did in single-axis dual-stage configuration (compare to Fig. 4.7).

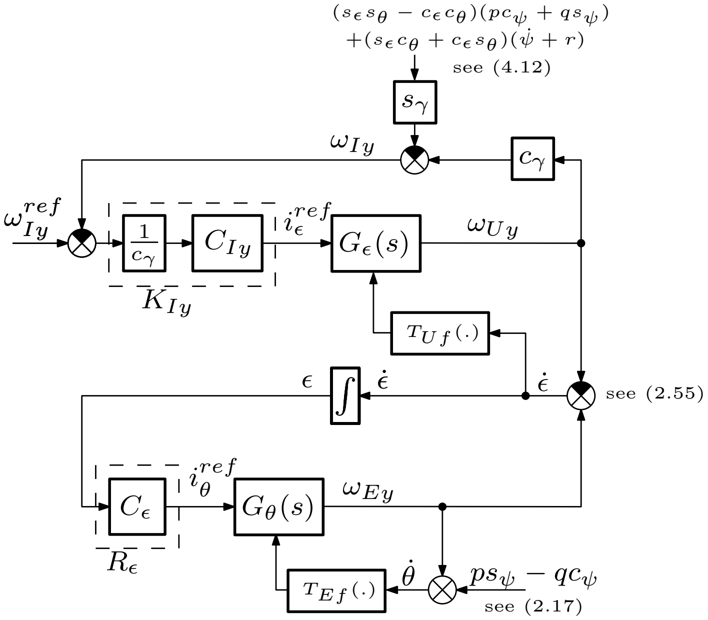


Figure 4.15: Dual-stage line-of-sight stabilization for the inner-elevation axis. The inertial-stabilization part of the scheme is redrawn from Fig. 4.14. To control the angle ϵ the elevation angle θ is used.

The control scheme for the inner-elevation axis is shown in Fig. 4.16. The scheme represents the extended version of the scheme from Fig. 4.13. As already suggested in this section to control misalignment angle γ the outer azimuth joint ψ is used. However, both joints operate around different axes. The impact of azimuth joint rate $\dot{\psi}$ on misalignment angle γ comes according to (2.60) via term $(c_\epsilon c_\theta - s_\epsilon s_\theta)$. The inverse of this term should be thus included in the controller $R_\gamma = \frac{1}{c_\epsilon c_\theta - s_\epsilon s_\theta} C_\gamma$. Since the angle ϵ is comparing to θ very small, a simplification

$$c_\epsilon c_\theta - s_\epsilon s_\theta \approx c_\theta$$

is used in the controller R_γ in Fig. 4.16. Notice that comparing to the inner-elevation axis, there is another disturbance that is entering the outer control loop directly. In fact, the source of this disturbance is the same as for classical double gimbal $pc_\psi + qs_\psi$ (compare with the scheme in Fig. 4.3).

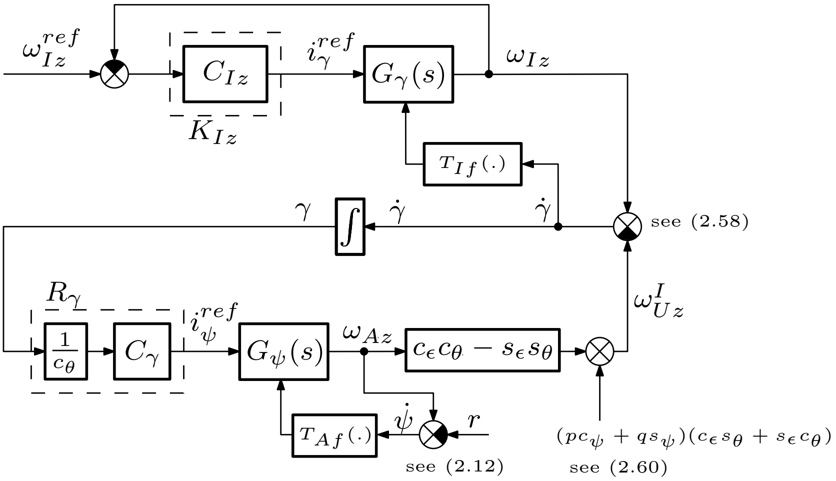


Figure 4.16: Dual-stage line-of-sight stabilization for the inner cross-elevation axis. The inertial stabilization part of the scheme is retaken from Fig. 4.13. To control the angle γ the current applied to azimuth joint motor ψ is used.

4.3.3 Experimental results of the LOS stabilization

In order to demonstrate the functionality of proposed control structure for the line-of-sight stabilization and to experimentally prove the suitability of the proposed decoupled control structure, several experiments were performed with the platform S250 (the platform was presented in Fig. 2.17):

1. Fig. 4.17 shows the result of the line-of-sight stabilization experiment in the elevation axis ω_{Iy}. The experiment corresponds to the scheme in Fig. 4.15 where the disturbing angular rate p was created by manipulating the platform in hands. This disturbing signal was also measured by another MEMS gyroscope mounted onto the platform base frame [B]. Signals ϵ and ω_{Iy} were measured and recorded too. The azimuth angle was set up to $\psi = 90°$.

2. Fig. 4.18 shows the experiment of the line-of-sight stabilization in the cross-elevation axis ω_{Iz}. The experiment corresponds to the scheme in Fig. 4.16. Similarly to the previous experiment the disturbance q was created by manipulating the platform in hands. The disturbing signal was also measured by another MEMS gyroscope mounted onto the platform base frame [B]. Signals γ and ω_{Iz} were measured and recorded. The azimuth angle was set up to $\psi = 90°$, the elevation angle $\theta \approx 55°$. The experiment clearly confirms the advantage of dual-stage concept — the disturbance is well rejected at all tested frequencies. Notice that the same experiment, but performed with double gimbal as seen in Fig. 4.6), showed that such disturbance was not rejected at all in the case of double gimbal.

3. Fig. 4.19 presents the reference-tracking experiment in the elevation axis. The exper-

iment corresponds to the scheme in Fig. 4.15 where ω_{Iy}^{ref} was commanded and ϵ and ω_{Iy} were measured and recorded. Several step commands were set up in the reference angular rate ω_{Iy}^{ref}. The graph shows the step response of the combination of two joints — the inner elevation joint ϵ and the elevation joint θ. Achieved results are comparable to those obtained with the single-axis dual-stage stabilization system with the only difference that the step response in ω_{Iy} is a little bit slower comparing to ω_1 in the experiment in Fig. 4.12. This is caused by the two reasons — PI rate controller K_{Iy} was tuned only empirically so that was not optimized at all and the second — the payload to be repointed has obviously much higher moment of inertia.

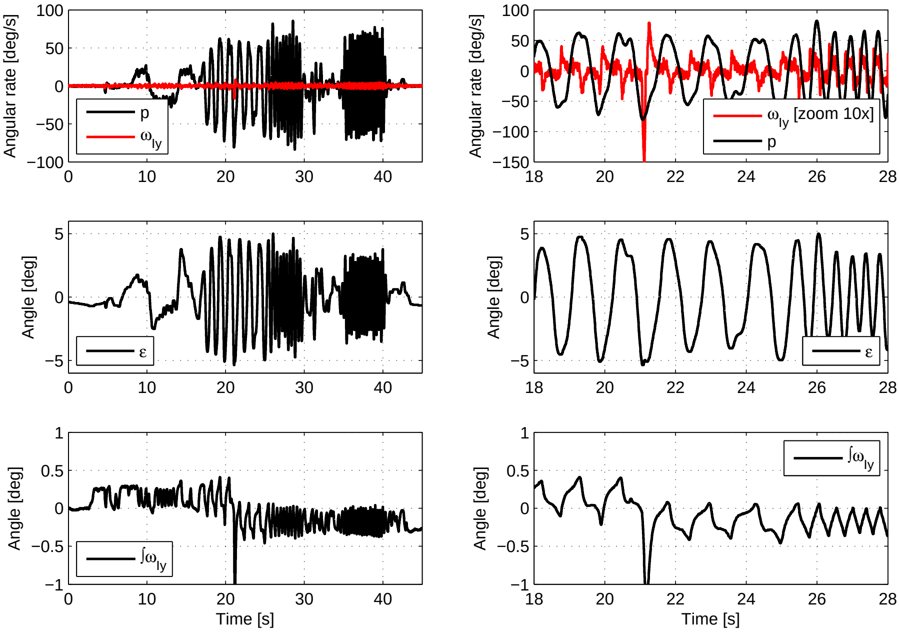


Figure 4.17: Experiment 1: Inertial line-of-sight stabilization in the elevation axis of the four-gimbal dual-stage platform. The experiment corresponds to the scheme in Fig. 4.15 where a disturbing angular rate p was created by manipulating the platform in hands. The azimuth angle was set up to $\psi = 90°$. The angular rate of the ω_{Iy} in the top-right figure is zoomed (multiplied by 10) in order to highlight the signal. The graph shows the fine attenuation of disturbing base motion, only at the time 21 s the inner-elevation angle ϵ reached its mechanical limit and the disturbance passed at into the inner gimbal. The bottom-left figure shows the integral of the payload's angular rate $\int \omega_{Iy}$. The integrated angular rate more less corresponds to the angle by that the line of sight was deflected in one axis (assuming no other movement around other payload axes — $\omega_{Iz} = \omega_{Ix} = 0$). $\int \omega_{Iy}$ was less than 0.5° except for the time of touching the limit in ϵ.

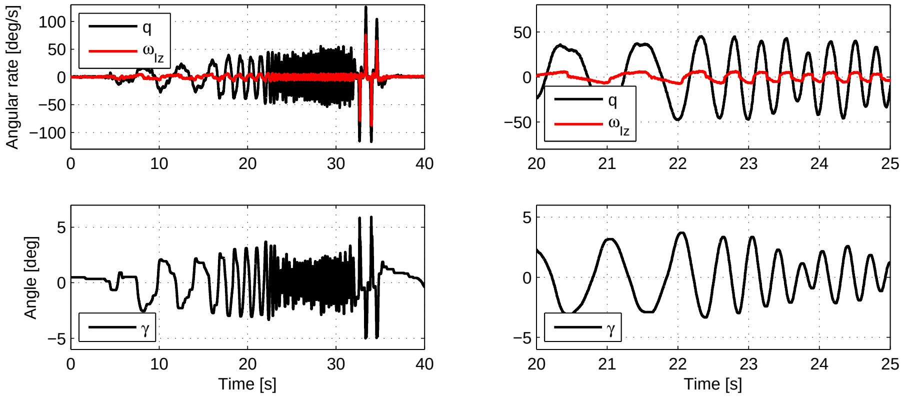


Figure 4.18: Experiment 2: Inertial line-of-sight stabilization in the cross-elevation axis of the four-gimbal dual-stage platform. The experiment corresponds to the scheme in Fig. 4.16 where the disturbing angular rate q was created by manipulating the platform in hands. The azimuth angle was set up to $\psi = -90°$, the elevation angle to $\theta \approx 55°$. The experiment shows the main benefit of the four-gimbal configuration — the disturbance is well rejected even at higher frequencies. The same experiment with the double gimbal showed the inability to reject such disturbances using double gimbal (see Fig. 4.6). The four-gimbal platform benefits from the missing disturbance that is entering the control loop directly. The only exception is when angle γ saturates and the disturbance enters the loop directly between times $33 - 34\,\text{s}$. Notice also that the disturbance enters the loop via term $\sin 55° {\cdot} \cos 0° = 0.82$ which is observable between times 33 – 34 s (ω_{Iz} is lower than q).

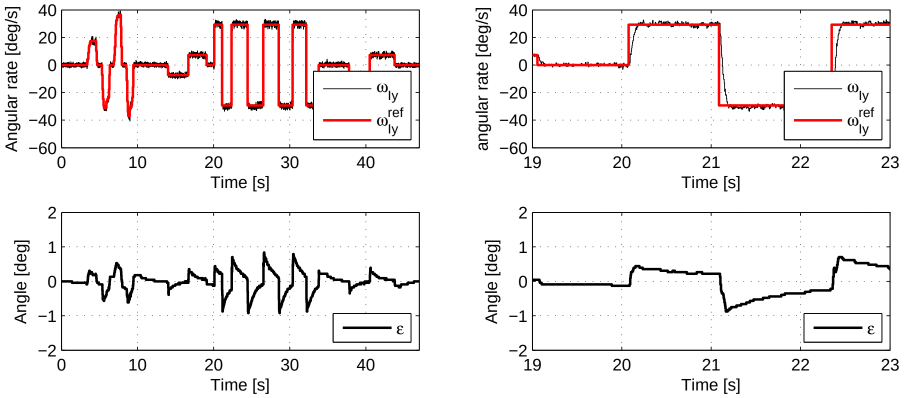
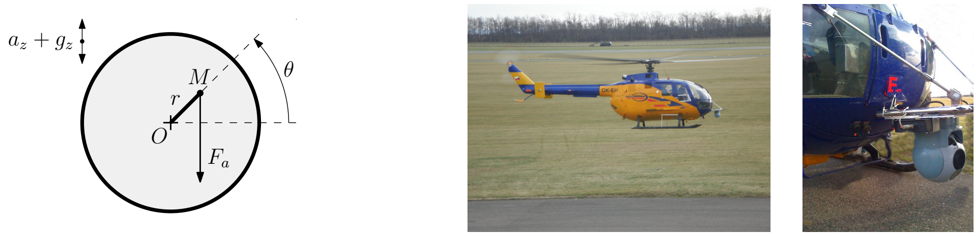


Figure 4.19: Experiment 3: Reference tracking in the elevation axis of the four-gimbal dual-stage platform. The experiment corresponds to the scheme in Fig. 4.15 where ω_{Iy}^{ref} was commanded and ϵ and ω_{Iy} were measured and recorded.

4.4 DISTURBANCE REJECTION BY ACCELERATION FEEDFORWARD

This section reports on a successful application of a feedforward scheme for rejection of a disturbing torque for an inertially stabilized double-gimbal platform. The key assumption for basic techniques of inertial line-of-sight stabilization is that the payload is perfectly balanced, that is, the gimbal axis goes through the center of gravity of the payload. However, this is totally unrealistic. Even with a very careful mechanical design, the perfect balancing cannot be achieved. Mechanical finetuning by attaching some extra pieces of material to the body of the payload does not solve the problem completely because the center of gravity can change during camera zooming. The geometry of the problem is sketched in Fig. 4.20 (a) for a single axis of rotation orthogonal to the direction of acceleration.

The content of this section is mainly based on the paper [9] by the author.



(a) Geometry of the problem for a single rotational axis perpendicular to the direction of the acceleration. In the case of the elevation gimbal in the azimuth-elevation (Az-El) double-gimbal system this is composed from both the gravitational and the translational acceleration.

(b) The inertially stabilized airborne camera platform attached to a helicopter has to withstand significant vibrations caused by the carrier.

Figure 4.20: Disturbing torque created via camera imbalance and vertical vibrations on a helicopter.

Even if the carrier (base) of the double-gimbal platform resides still on the ground, the static mass unbalance leads to a disturbing torque acting on the payload as demonstrated using experimental data in Fig. 4.21. While the control system keeps the inertial angular rate constant, the voltage applied to the motor armature winding exhibits a periodic pattern. This reveals that the inertial angular-rate control system has to cope with a periodic disturbance.

The situation is even worse when the carrier is exposed to a linear acceleration. This is the case when the platform is attached to a helicopter (as in Fig. 4.20 (b)). The unavoidable vibrations of the helicopter are passed to the platform and give rise to a more pronounced disturbing torque.

The situation for one rotational degree of freedom is modeled using a block diagram in Fig. 4.22. The model includes all the necessary components for a one-axis motion control system driven by a DC motor, including a model of a friction. It makes it clear how the

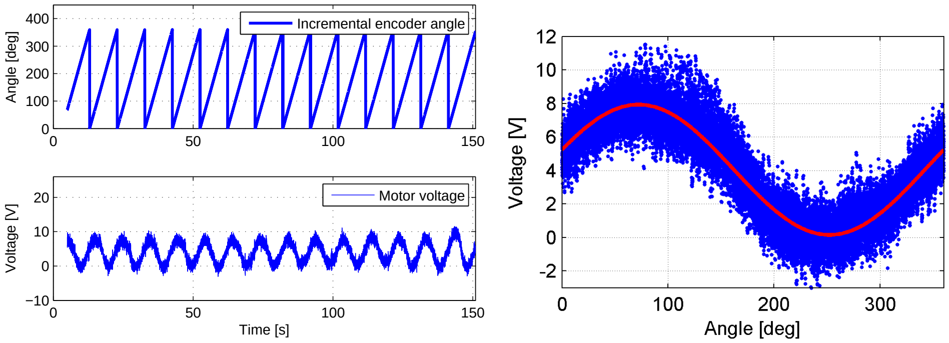


Figure 4.21: Left: When a constant angular rate of the gimbal is required, the periodic pattern is observed in the control voltage as a consequence of the gravity projection onto a disturbing torque for a statically unbalanced elevation gimbal seated on the ground.
Right: The result of an experimental identification of a static mass unbalance. Fitting the sine wave is described by (4.20). The sum of sine and cosine results in a shifted sine. The blue color represents the data of several turns of the elevation gimbal. The red line represents the result of the fitting.

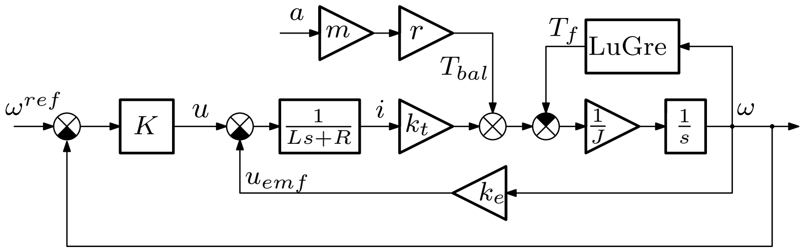


Figure 4.22: Block diagram of the feedback setup with the entry point of the disturbance T_{bal}.

translational acceleration projects into a disturbing torque.

Certainly this disturbing torque can be left for the existing inertial angular rate feedback to reject but great advantage can be taken from the often neglected feedforward compensation. The cause of the disturbing torque — the linear acceleration of the carrier — can be measured, filtered and fed forward to the two direct drive motors. This technique of acceleration feedforward is often used in the domain of hard disk drives, see [46] [47] [48], but has not been described in the area of inertial stabilization.

4.4.1 Projection of the base acceleration into other gimbals

Using transformation matrices R_B^A and $R_B^E = (R_A^B)^T$, which were introduced in section 2.2.3, the acceleration that is experienced by the gimbal base a_B (identical with the carrier) is projected to the outer (azimuth) gimbal as

$$a_A = R_B^A a_B \tag{4.13}$$

and to the inner (elevation) gimbal as

$$a_E = R_B^E a_B. \tag{4.14}$$

The acceleration a_E of the center of gravity of the inner gimbal generates a disturbing torque around the elevation y-axis in the elevation gimbal

$$T_{Ey} = [r_E \times m_E a_E]_y, \tag{4.15}$$

where r_E is a displacement vector (between the center of mass and the axle) and m_E is the mass of the inner gimbal including the payload.

Similarly, the acceleration a_A generates the disturbing torque T_{Az} around the outer (azimuth) gimbal axis. But now it is comprised of two components

$$T_{Az} = [r_A^A \times m_A R_B^A a_B]_z + [R_E^A r_E^E \times m_E R_B^A a_B]_z. \tag{4.16}$$

The first component is the unbalance of the azimuth gimbal described by the mass m_A and the vector r_A, and the second component represents the unbalance of the elevation gimbal described by the mass m_E and vector r_E, although projected to azimuth gimbal via rotation matrix R_E^A.

4.4.2 Experimental identification of elevation unbalance

In this section it is assumed that the (inertial) angular rate controller K from Fig. 4.22 is designed and fully working. It provides the reference tracking of the angular rate ω^{ref} up to some reasonably high frequency (velocity loop bandwidth). The controller achieves this by measuring the angular rate ω using MEMS inertial angular rate sensors and setting the control voltage u. Within this setting, when the constant reference angular rate ω^{ref} is required, the control voltage u should be constant (in steady state).

Nevertheless, the control voltage observed during the experiment clearly contains a sinusoidal content (see Fig. 4.21 or Fig. 4.23 on the left). This content corresponds to the gimbal unbalance since it is obviously dependent on the joint angle. Plotting the data from the left graph in the Fig. 4.21 as the voltage u being a function of the gimbal angle θ gives the right graph in the same Fig. 4.21. The sine wave originates from (4.15) with the base acceleration equal only to the gravity $a_B = \begin{bmatrix} 0 & 0 & g \end{bmatrix}^T$ as follows

$$\begin{aligned} T_{Ey} =& \left[r_E \times \left(m_E R_B^E \begin{bmatrix} 0 & 0 & g \end{bmatrix}^T \right) \right]_y && (4.17) \\ =& -m_E g (r_{Ez} \sin\theta + r_{Ex} \cos\theta) && (4.18) \\ =& \bar{A} \sin\theta + \bar{B} \cos\theta. && (4.19) \end{aligned}$$

In fact, it is not required to know the vector r_E and the mass m_E separately. The parameters $\bar{A}$ and $\bar{B}$ fully describe the torque necessary to reject the disturbance coming

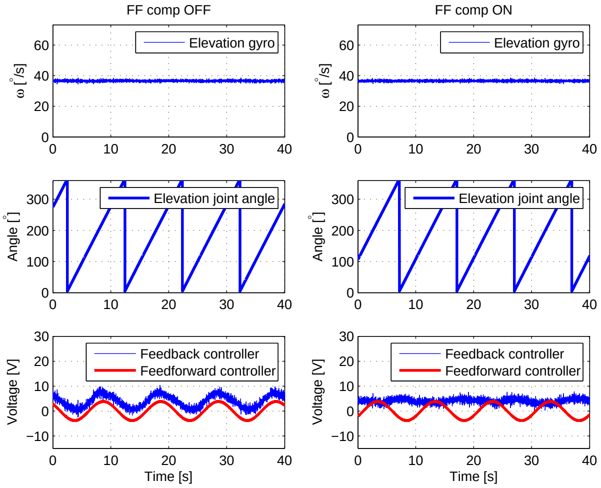


Figure 4.23: Identification experiment shows the impact of a limited accuracy of the static mass unbalance identification. The elevation gimbal is rotating at a constant speed by a rate controller. On the left: the unbalance is clearly visible in the feedback controller's voltage (the feedforward red sine curve is plotted here too but it is not applied). On the right: the identified sine is added to the feedback controller's output. Ideally, only the friction force should be left for the feedback to compensate. In reality, the control voltage is not perfectly constant, which reveals that the feedforward did not reject the disturbance completely.

from static mass unbalance scaled by the gravity. In order to compensate not only the gravity but also a nonzero translational acceleration a_B, the acceleration measurement must be related to the to the gravity. It is not even necessary to estimate the parameters $\bar{A}$ and $\bar{B}$ relating the gimbal angle and the disturbing torque, but instead the parameters A and B relating the angle and the voltage necessary to reject the disturbance

$$u(\theta) = A \sin\theta + B \cos\theta + C. \tag{4.20}$$

The parameter C represents the constant voltage level required to reject friction torque. The parameters A and B are defined by

$$A = -m_E \cdot g \cdot r_{Ez} \cdot \frac{k_t}{R} \tag{4.21}$$

and

$$B = -m_E \cdot g \cdot r_{Ex} \cdot \frac{k_t}{R}. \tag{4.22}$$

To estimate the parameters A, B and C, the least squares fitting was used since (4.20) is linear in these parameters. The resulting fit is visualized in Fig. 4.21 on the right.

4.4.3 Feedforward disturbance rejection

With a successfully identified and validated model of dynamics of the elevation gimbal, including the static mass unbalance, a feedforward disturbance rejection scheme based on measured acceleration may be devised. As discussed in the previous sections, an accelerometer is used to measure the vibrations.

A key decision is where to place the accelerometer. One possible choice is to place the sensor inside the elevation gimbal — to fix it onto the payload. Using this approach the values of a_{Ex} and a_{Ez} are measured directly. A major disadvantage of this choice is that the sensor must be placed exactly to the elevation-gimbal axis otherwise its signal is distorted by the centripetal acceleration during the elevation gimbal rotations.

Another option (the one chosen in this section) is to attach the accelerometer to the base, thus measuring a_B. The acceleration of the elevation gimbal a_E must be then computed. The situation is depicted in Fig. 4.24. The feedforward controller F is using the accelerometer signal a_B and feeds it forward as a voltage u_{ff} applied to the elevation motor (in addition to the feedback controller's voltage u_{fb}).

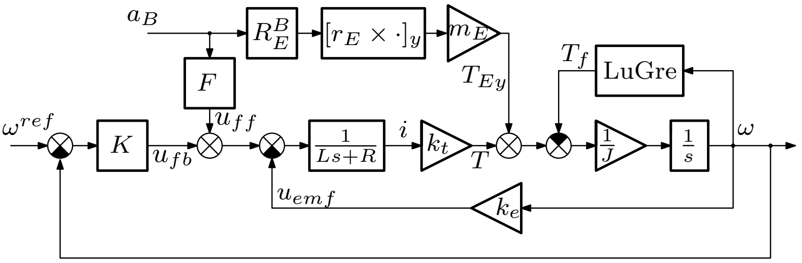


Figure 4.24: Block diagram of the feedback setup with added feedforward controller F.

Looking at the Fig. 4.24, the key idea for design of a feedforward controller F is to achieve the transfer function from a_B to T identical to the transfer function from a_B to T_{Ey}. Using the identified parameters A and B this is satisfied by the controller with the transfer function

$$F(s) = \frac{1}{g} \cdot \left(\frac{L}{R}s + 1\right) \cdot \Big[\begin{bmatrix} B & 0 & A \end{bmatrix} \times R_B^E a_B \Big]_y, \tag{4.23}$$

which is, however, not proper due to the motor inductance L. An additional low-pass filter must be added or the electrical dynamics of the motor may be neglected to obtain the reduced version of the controller in the form

$$F = \frac{1}{g} \cdot \Big[\begin{bmatrix} B & 0 & A \end{bmatrix} \times R_B^E a_B \Big]_y. \tag{4.24}$$

The second variant may be easily justified in the case when the sampling rate of the control loop is $f_s = 200\,\text{Hz}$, which is ten times lower than the frequency R/L.

A challenge in this type of feedforward disturbance rejection lies in the limited accuracy of the identified model, including the static mass unbalance. Possibly the dynamics of the accelerometer can not be neglected. The voltage u_{ff} produced by the feedforward controller F then does not reject the disturbance completely, as seen in the experimental data in Fig. 4.23. This calls for implementation of some adaptation.

4.4.4 Simulations

To assist in designing the feedforward compensator, numerical simulations using Matlab/Simulink were used (see the model scheme in Fig. 4.25 on the right). The feedback loop was always closed. The new feeforward control always acted as a complement to the MEMS gyro-based feedback control. The input to the simulations is the constant speed rotation of the elevation gimbal with simultaneously applied sinusoidal vertical vibrations a_{Bz} of amplitude $8\,\mathrm{ms}^{-2}$. The frequency of the vibrations was linearly changing from 0.1 Hz to 10 Hz. Fig. 4.26 shows the result of simulation with the feedforward controller F from (4.24) switched on and off. Results are complemented by the simulation where the original inner gimbal mass m_E was perturbed by 10%.

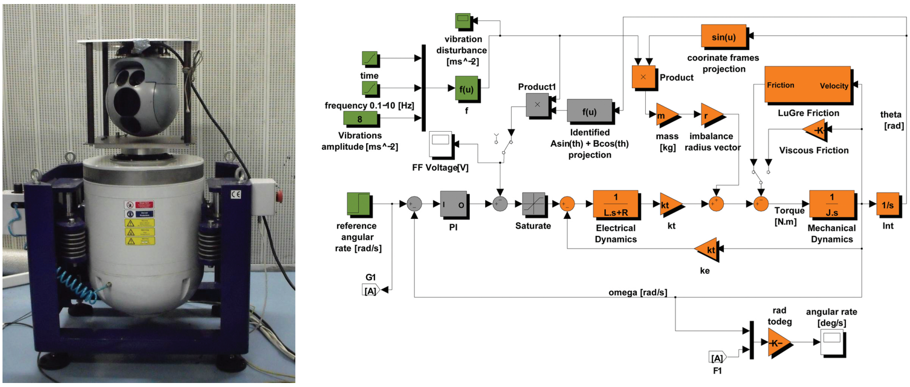


Figure 4.25: Right: The inertially stabilized platform placed on a vibration table.
Left: Simulink model demonstrating the feedforward controller structure. The elevation gimbal model with the friction and imbalance is depicted by orange, dark green represents the input signals (ω_{ref} and a_B) and grey are the couple of feedforward and feedback controller.

4.4.5 Experimental results

An experimental verification was done using a professional laboratory vibration table (see Fig. 4.25 on the left). The table provides a vertical motion with the frequency ranging from one to several tens of Hz with the amplitude of several centimeters. The experiment results are shown in Fig. 4.27. Consistently with the simulations, the reduced version of the feedforward controller (4.24) was implemented. Experimental results clearly confirm the usefulness of the feedforward compensator, even in its reduced form with the neglected accelerometer's dynamics.

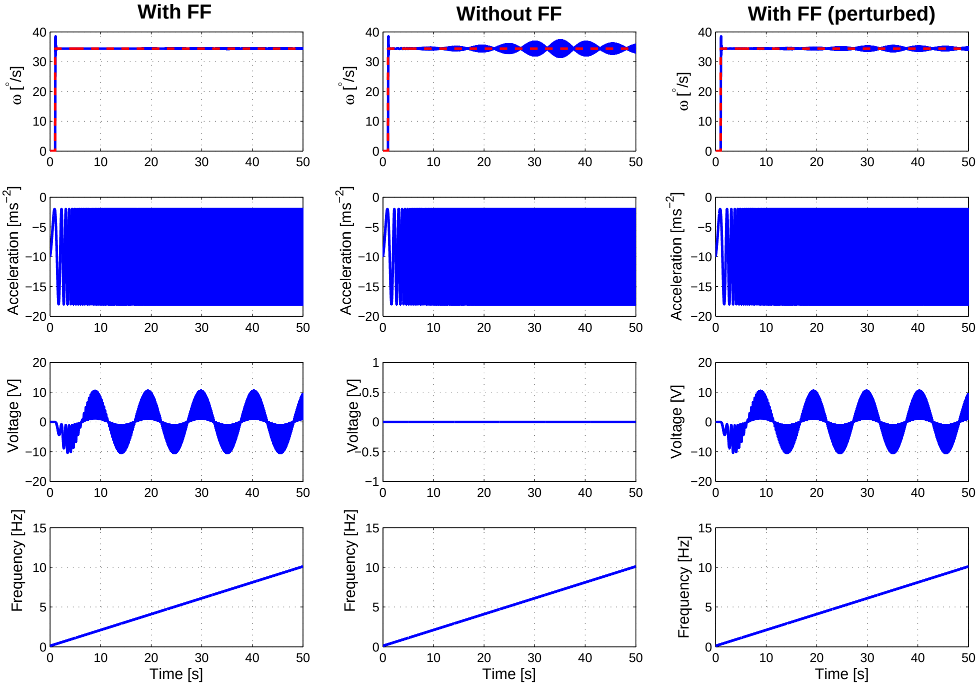


Figure 4.26: Simulation with the feedforward controller switched ON, OFF and ON with perturbed camera mass.

Left: Feedforward controller switched ON. The angular rate of the elevation gimbal (ω) is not disturbed by vibrations at all.

Middle: Feedforward controller switched OFF. The vibrations passing into the elevation gimbal are clearly visible. The amplitude of vibrations passing into ω is changing in accordance with the projection R_B^E as the gimbal is rotating by the constant speed.

Right: Simulation with the Feedforward controller switched ON but the mass of the camera is perturbed by 10 %. The vibrations passing into elevation gimbal are rejected but not completely.

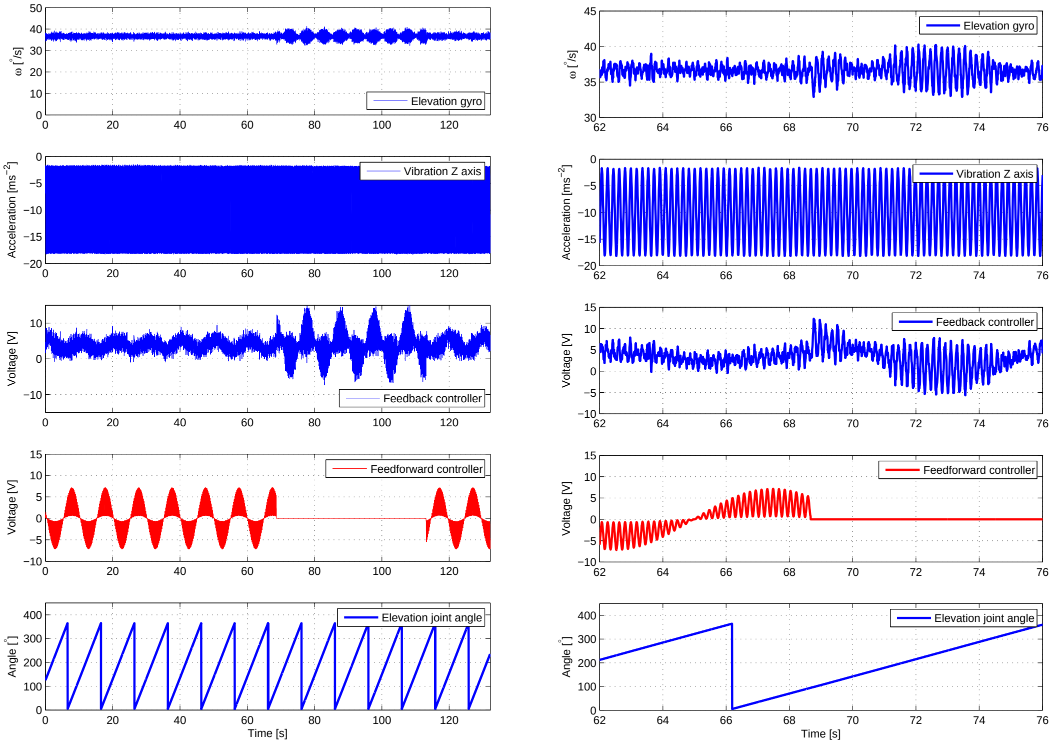


Figure 4.27: On the left: Comparison of the feedforward controller turned on and off during the experiment on the vibrational platform. See the feedforward controller signal to distinguish when it is turned on and off. On the right: The graph from the left in a detail.

4.5 CONCLUSIONS

This chapter introduced the concept of stabilization of the optical axis — the so-called line-of-sight stabilization. Within subsections it was shown how to design the control structure for line-of-sight stabilization for various mechanical configurations. Mathematical models of these configurations were already studied in chapter 2. It was shown, that it is possible to decouple the line-of-sight stabilization into the two axes — elevation and cross-elevation. Each axis was then studied separately — errors caused by neglecting coupling dynamics were regarded as yet another disturbance. Finally, each section contained an experiment with a practical verification of the line-of-sight stabilization.

Close attention was paid to the analysis of a way how disturbing rotation of the platform base is entering the stabilization loops. It was shown using block diagrams and also experimentally that in the case of a double-gimbal platform, there is always some disturbance that is entering the cross-elevation control loop directly. This means it is not possible to reject this disturbance at higher frequencies. This issue was overcome by introducing a four-gimbal dual-stage platform, where such disturbances were already significantly rejected mechanically.

The last section of this chapter dealt with the possibility to reject disturbances that appear during the line-of-sight stabilization of an unbalanced gimbal when the platform is exposed to translational vibrations. By identifying of such imbalance and introducing an accelerometer-based feedforward control, a significant rejection of the imbalance-based disturbing torque was achieved. Sufficient disturbance rejection was proved by both simulations and the laboratory experiment using the vibrational table.

Chapter 5

Visual tracking on top of inertial stabilization

This chapter describes a novel image-based pointing-tracking feedback control scheme for an inertially stabilized airborne camera platform combined with a computer vision system. The key idea is to enhance the intuitive decoupled controller structure with measurements of the camera's inertial angular rate around its optical axis. The resulting controller can also compensate for the apparent translation between the camera and the observed object, but then the velocity of this mutual translation must be measured or estimated. Numerical simulations of the control scheme are accompanied by laboratory experiments with a real benchmark system.

The content of this chapter is based on the modified version of the journal paper [2]. Preliminary versions of that paper were presented at [3] and [4].

Automatic visual tracking on mobile carriers

All the works cited in the introduction to chapter 4 (including the references made therein) mostly focus on the task of inertial stabilization only. The task of extending the inertial angular-rate feedback loop to visual tracking system is only dealt with at a rather simplistic level in [37] by suggesting the common cascaded control structure for every rotational degree of freedom: an inner SISO[1] loop (inertial rate stabilization) is accepting commands from the output of the corresponding outer (visual tracking) loop. There are some pitfalls hidden in this decoupled approach, though. This chapter describes the troubles that are encountered when using the classical double-gimbal platform or four-gimbal platform and offers a solution. To the best of the author's knowledge, this is the first formal treatment of the visual pointing and tracking for inertially stabilized camera systems.

[1] single-input-single-output

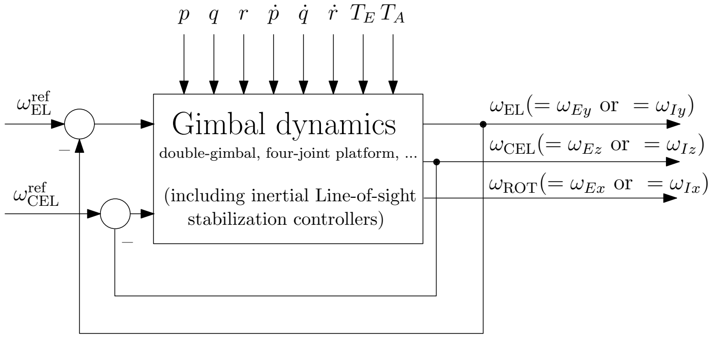

p q r $\dot{p}$ $\dot{q}$ $\dot{r}$ T_E T_A
$\omega_{\mathrm{EL}}^{\mathrm{ref}}$
$\omega_{\mathrm{CEL}}^{\mathrm{ref}}$
$-$
$-$
Gimbal dynamics
double-gimbal, four-joint platform, ...
(including inertial Line-of-sight stabilization controllers)
$\omega_{\mathrm{EL}}(=\omega_{Ey}$ or $=\omega_{Iy})$
$\omega_{\mathrm{CEL}}(=\omega_{Ez}$ or $=\omega_{Iz})$
$\omega_{\mathrm{ROT}}(=\omega_{Ex}$ or $=\omega_{Ix})$

Figure 5.1: Scheme is showing how line-of-sight stabilization using double gimbal platform and four-gimbal platform may be both encapsulated with the same inputs (references ω_{EL}^{ref} and ω_{CEL}^{ref}) and the same outputs (angular rates of the camera measured by rate gyroscopes ω_{EL}, ω_{CEL} and ω_{ROT}.)

Visual tracking on top of Inertial line-of-sight stabilization

The visual tracking algorithm presented in this chapter falls into the category of the position control within the cascade structure presented in section 3.1. It means it is required that the inertial line-of-sight stabilization has been already designed according to chapter 4. Independently of whether double-gimbal or full four-gimbal platform is to be used, the line-of-sight stabilization may be encapsulated into a single system as shown in Fig. 5.1. The visual tracking is then built around this system so that the visual controller can manipulate the line-of-sight by specifying the two reference inertial angular rates. In particular these are the reference angular rates $\omega_{Ey}^{ref} = \omega_{EL}^{ref}$ and $\omega_{Ez}^{ref} = \omega_{CEL}^{ref}$ for the classical double gimbal as presented in Fig. 4.2. In case of the full four-gimbal platform these references are the angular rates $\omega_{Iy}^{ref} = \omega_{EL}^{ref}$ and $\omega_{Iz}^{ref} = \omega_{CEL}^{ref}$ as presented in Fig. 4.15 and Fig. 4.16. To complete the notation, the angular rate measured around the optical axis is denoted by ω_{ROT} which is equal to either ω_{Ix} or ω_{Ex}.

Using the notation ω_{CEL}, ω_{EL} and ω_{ROT} for the line-of-sight angular rates, this chapter is written to be independent of the chosen platform. However, at several places it will be useful to highlight some behavior of the one or the other. Finally this chapter is concluded by the laboratory experiments with the double-gimbal platform H240, and flight test experiments with the four-gimbal platform S250.

5.1 IMAGE TRACKER DESCRIPTION

This section gives some details on the automatic image tracker, even though for the purpose of this work it can be viewed as a black-box device. It serves the purpose of a "relative displacement sensor". The image tracker is an algorithm (implemented in a software routine) which recognizes the target object in the input image frame sequence and returns its coordinates in every image frame of the acquired sequence. The target is usually selected by a human operator, who marks the target in the first image frame. The control system then steers the camera in order to get the (image of the) target into the center of the field of view (center of the image frame) and hold it there.

It appears that image tracking of a ground object for the purpose of airborne surveillance is one of the most difficult image tracking tasks [49], [50]. These are the reasons:

- Weak visual differences between the tracked object and the background.
- Many localization results assume that the background is static with respect to the camera, which is certainly not the case for cameras carried by aircraft.
- The image of the target object is usually very small. Tracker useful for airborne surveillance must work with objects which projects into images as patches smaller than 10 by 10 pixels.
- The relative movement of the image of the target can be very fast (multiples of object's size from frame to frame) mainly due to the rotation of the camera, either intentional or unwanted.

There are many algorithms for image tracking in different conditions and tasks. A few families of algorithms can be considered:

- *Pattern matching algorithms* compare the captured image of the target with its model. An error function is defined that measures the similarity between the model (represented by an image patch) and the actual image in a specified position. The algorithm searches for a position where the error is at minimum. The representative of this class of algorithms are SSD (sum of square differences) [51] and the basic K-L (Kanade-Lucas) algorithm [52].
- *Feature tracking algorithms* are based on extraction of a small number features (important points) from the image frame and their tracking. The features should satisfactorily describe the object and be easily recognized in the next image frame. There are many ways how to define and track the features. Mean-shift algorithm is based on a density analysis of feature space [53], but some more sophisticated algorithm were developed, the examples of which are KLT (Kanade-Lucas-Tomasi tracker) [54], SIFT (Scale Invariant Feature Transform) [55], MSER (Maximally stable extremal regions) [56] and [57].
- *Object recognition trackers* aim at recognizing an object in the image frame. Machine-learning techniques (for example [58]) are applied on features which are extracted in a preprocessing stage. The target object is recognized and localized in each image. The learning stage of the procedure could be run offline, but then only a "learned" class of objects could be tracked. On-line learning turns out more useful for the present application. An example of this class of algorithms is On-line Boost [59].

These algorithms differ in tracking capabilities when it comes to different types of objects, required conditions, machine time consumption and many other operational characteristics. The experiments conducted with both benchmark systems relied on the SSD algorithm. It is easy to implement and is capable of tracking small objects and recognizing similar objects. A major disadvantage of the SSD algorithm is its high machine time consumption.

From the control system viewpoint, the major complication that the image tracker brings into the feedback loop is the transportation delay. The delay consists of computational time for the actual tracker algorithm and the time for image capturing by the hardware. In the benchmark system, a standard PAL video camera with analog output was used, for which 5-20 ms were needed for image capturing. Then it takes another 40 ms (25 Hz frequency) to transmit an analog video signal from the camera. The analog video signal is then captured by a video grabber and after some 10ms is transferred to the computer. Only then the tracking algorithm could be started. Therefore the minimum transportation delay is around 60 ms plus the computing time of tracker algorithm. It is only the latter that could be minimized by implementing more efficient algorithms.

5.2 MODELING THE DYNAMICS FOR POINTING AND TRACKING

Before starting discussions on ways to design and implement a feedback controller for the task of pointing and tracking, a model must be developed. At the initial treatment, the inertial angular rate (feedback) loops can be regarded as perfect within the appropriate frequency range and saturation bounds, that is, the commanded inertial angular rates $\omega_{\text{EL}}^{\text{ref}}$ and $\omega_{\text{CEL}}^{\text{ref}}$ can be regarded as perfectly followed by the inner loops. To develop a mathematical model for this idealized situation, a few basic concepts from the established domain of visual servoing will be given. The next few paragraphs are fully based on the two chapters from [13] dedicated to the computer vision and vision-based control. They are given here just for a convenience of a reader nonacquainted with these concepts. Another comprehensive introductory material is [60].

5.2.1 Perspective projection

The objects to be observed are located in the full 3D world while the camera can only record their 2D image. The coordinates of the object in the world (on the ground) expressed in the camera frame are given by $P = [x, y, z]^T$. Simplifying a bit the model of the optics, the so-called pinhole assumption is made, which defines the image coordinate frame as follows. At a focal distance λ from the origin of the camera coordinate frame, consider the image plane orthogonal to the optical axis of the camera. The coordinates of the point of intersection of the line connecting the object with the origin are $p = [u, w, \lambda]^T$. The vector $s = [u, w]^T$ thus gives the image coordinates. All this is visualized in Fig. 5.2. Thanks to the pinhole

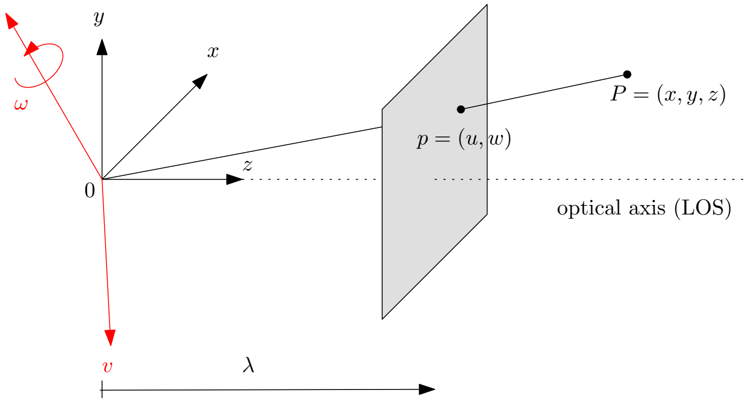


Figure 5.2: Coordinates of the object on the ground expressed in the coordinate frame attached to the camera and (after projection) in the image plane. Rotation $\omega_{R,C}^{C}$ and translation v_C of the camera frame within with respect to the inertial frame is also illustrated (redrawn from [13]).

assumption

$$k \begin{bmatrix} x & y & z \end{bmatrix}^T = \begin{bmatrix} u & w & \lambda \end{bmatrix}^T, \tag{5.1}$$

we have that

$$u = \lambda \frac{x}{z}, \quad w = \lambda \frac{y}{z}. \tag{5.2}$$

To make this story complete, the coordinates in the image plane should then be quantized and the origin should be moved to the lower left corner to obtain pixel coordinates $[r, c]^T$

$$-\frac{u}{s_x} = (r - o_r), \quad -\frac{w}{s_y} = (c - o_c), \tag{5.3}$$

where s_x and s_y are the pixel dimensions and o_r and o_c are half the width and height of the image frame in pixels.

Nonetheless, for the analysis we will stick to u and w variables to make the formulas less involved. In simulations and experiments presented here, the “pixelized” information will be considered centered (again, in the name of simplicity). That is, we will use

$$-\frac{u}{s_x} = r, \quad -\frac{w}{s_y} = c. \tag{5.4}$$

The computer vision system that processes the images captured by the camera can surely perform this centering before sending the data to the pointing-tracking controller.

5.2.2 Camera motion and the interaction matrix

This subsection is again extracted from the nice introduction to image-based visual servoing in the textbook [13]. Consider the movement of the camera in the inertial space characterized by its linear and rotational velocities $v_C = [v_{Cx}, v_{Cy}, v_{Cz}]^T$ and $\omega_C = [\omega_{Cx}, \omega_{Cy}, \omega_{Cz}]^T$, both expressed in the camera frame. Stack them together to form a time-dependent vector $\xi(t) = [v_C(t), \omega_C(t)]^T \in \mathbb{R}^6$. To be rigorous, we should write $\omega_{R,C}^C$ to emphasize that it is an angular rate of the camera frame with respect to the reference (inertial) frame, expressed in the camera frame, and similarly $v_{o_C}^C$ to emphasize that it is a translational velocity of the origin o_C of the camera coordinate frame with respect to the inertial frame, also expressed in the camera frame. But this would yield the equations illegible.

The motion of the object as viewed by the camera is described by the so-called image feature velocity $\dot{s}(t)$, which can be obtained as a derivative of the image feature vector (in the simplest case it is just a position of some significant point). The nice thing is that it is possible to relate ξ and $\dot{s}$ by a transform resembling the concept of Jacobian and denoted often an interaction matrix or image Jacobian

$$\dot{s}(t) = L(s, z, \lambda)\xi(t). \tag{5.5}$$

Next consider the simplest case of a single-point feature and assume that the ground object does not move. Extension of the results stated here to the case of a moving ground target is feasible, but the resulting interaction matrix will be a function of the velocity of the ground object, which is unknown to the inertial stabilization system (but it may be worth exploring if at least rough estimate of the object velocities can be used). This matrix is derived in [13], page 415, equation (12.14) as

$$\begin{bmatrix} \dot{u} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} -\frac{\lambda}{z} & 0 & \frac{u}{z} & \frac{uw}{\lambda} & -\frac{\lambda^2+u^2}{\lambda} & w \\ 0 & -\frac{\lambda}{z} & \frac{w}{z} & \frac{\lambda^2+w^2}{\lambda} & -\frac{uw}{\lambda} & -u \end{bmatrix} \begin{bmatrix} v_{Cx} \\ v_{Cy} \\ v_{Cz} \\ \omega_{Cx} \\ \omega_{Cy} \\ \omega_{Cz} \end{bmatrix}. \tag{5.6}$$

The procedure for the derivation is straightforward: first, express the position of a fixed (not moving) point on the ground in the coordinate frame of the moving (rotating and translating) camera, and then project these new coordinates to the image plane. (It is vital to keep in mind within which coordinate frame the velocity vectors are being expressed. This is quite tedious. In this case, both the translational and rotational velocities are indeed considered with respect to the inertial reference frame but are expressed in the camera frame).

It appears useful to highlight the structure in the interaction matrix by writing it as a composition of two parts

$$\dot{s} = L_v(u, w, z)v_C + L_\omega(u, w)\omega_C, \tag{5.7}$$

because it turns out that only the part corresponding to the translation of the camera coordinate frame depends on the image depth (distance to the observed ground object) z. The rotational part is independent of z. The focal length λ is regarded as a fixed parameter.

The three components ω_{Cx}, ω_{Cy} and ω_{Cz} define the inertial angular rate vector $\omega_C = \left[\omega_{Cx}, \omega_{Cy}, \omega_{Cz}\right]^{\mathrm{T}}$ in the camera coordinate frame [C], which is rotated with respect to the elevation gimbal frame [E] (in case of the dual-stage platform) using a fixed (constant) rotation matrix R_C

$$\omega_E = R_C^E \omega_C = R_C \omega_C = \begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \omega_C. \tag{5.8}$$

In case of the four-gimbal platform the situation is the same. Only the camera frame is attached to inner gimbal [I] instead

$$\omega_I = R_C^I \omega_C = R_C \omega_C. \tag{5.9}$$

While by using the two direct drive motors it is possible, at least partially, to affect the vector ω_E by commanding its two components $\omega_{Ey}(= \omega_{\mathrm{EL}})$ and $\omega_{Ez}(= \omega_{\mathrm{CEL}})$, it is rather unlikely that the translational velocity v_C will be commanded by the autopilot based on the needs of the pointing-tracking algorithm. (But some projects might allow it).

Therefore, in order to develop some insight into the model, forget v_C for a moment (assume $v_C = 0$ temporarily, it can be treated as a disturbance later, either estimated or not). Using the transformation (5.8) and the rotation part of the interaction matrix (5.6) we get

$$\begin{bmatrix} \dot{u} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} -\frac{uw}{\lambda} & \frac{\lambda^2+u^2}{\lambda} - w\tan\theta \\ \frac{\lambda^2+w^2}{\lambda} & \frac{uw}{\lambda} + u\tan\theta \end{bmatrix} \begin{bmatrix} \omega_{\mathrm{EL}}^{\mathrm{ref}} \\ \omega_{\mathrm{CEL}}^{\mathrm{ref}} \end{bmatrix} \tag{5.10}$$

and the camera tilt angle θ evolves according to

$$\dot{\theta}(t) = \omega_{\mathrm{EL}}^{\mathrm{ref}}(t). \tag{5.11}$$

5.2.3 Linearization at distinguished operating points

In order to develop an insight into the model (5.10), consider the situation when $\theta = 0$ (a wing-level flight and the camera pointing towards the horizon) and $w = 0$ (the observed object vertically centered on the screen). The dynamics is then constrained to one dimension and the equation simplifies to

$$\dot{u} = \frac{\lambda^2 + u^2}{\lambda} \omega_{\mathrm{CEL}}^{\mathrm{ref}}. \tag{5.12}$$

The term $(\lambda^2 + u^2)/\lambda$ expresses the nonlinear relationship between the angle and the line segment in the image plane. This is illustrated in Fig. 5.3

The focal length λ ranges for both benchmark systems in $[4.2, 42]$ mm. The width of the CCD camera chip is 3.2 mm. Hence, for the maximum zoom, the nonlinear term can be approximated by λ even for u approaching the maximum value, that is, the observed object is initially located near the borders of the field of view (and the control goal is to bring it to the center). The linear dynamics is then

$$\dot{u} = \lambda \omega_{\mathrm{CEL}}^{\mathrm{ref}}, \tag{5.13}$$

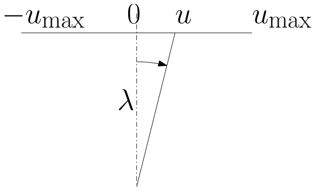


Figure 5.3: Relationship between the image coordinate system and the corresponding angle.

that is, the model of dynamics is represented by a pure integrator with a gain λ (given by the optics). For shorter focal lengths (approaching the lower limit of 4.2 mm) this approximation is only valid for correspondingly smaller u, that is, for tracking purposes only, not for (re)pointing over a large part of the image plane.

5.2.4 Analysis of achievable bandwidth for pointing and tracking

The computer vision system works at discrete time instants with the sampling period T_s ranging between something like 0.1s and 2 s (depending on complexity and performance of the algorithm), which is relatively long compared to 200 Hz of the inner inertial rate loop. This introduces a total delay τ of about $1.5T_s$ into the feedback loop.

It is known that the achievable bandwidth is limited by several properties of the system, delay being one of them. With the sampling period of the image tracker set to $T_s = 0.5\,s$, the achievable bandwidth is approximately limited by

$$\omega_{\mathrm{BW}} < \frac{1}{\tau} = \frac{1}{1.5T_s} = 1.3\,\mathrm{rad}/s = 0.2\,\mathrm{Hz}. \tag{5.14}$$

It is derived in [61] from ideal closed-loop transfer functions achievable for systems with a delay τ. Ideally, $T(s) = 1e^{-\tau s}$, therefore $S(s) = 1 - e^{-\tau s}$. By Taylor series expansion $S(s) \approx \tau s$. Therefore $|S(j\omega)|$ crosses 0 dB at about $1/\tau$.

This suggests that the fastest possible pointing-tracking loop will work up to a fraction of 1 Hz if the information from the image tracker is provided twice per second and is delayed one sample period. This roughly corresponds to the classical rule-of-thumb rules [62] for selection of a sampling rate for undelayed systems as 10 to 20 times the closed-loop bandwidth.

5.3 DECOUPLED POINTING AND TRACKING

Proceeding one step further beyond the mere inertial stabilization, the question of the most suitable feedback control configuration for automatic visual tracking pops up. Shall we use the immediate extension which closes a SISO tracking loop around the corresponding SISO inertial rate loop?

The cascade approach is justified: whereas the inner (inertial rate) loop aims to attenuate the disturbances at middle and high frequencies, the outer (pointing) loop should

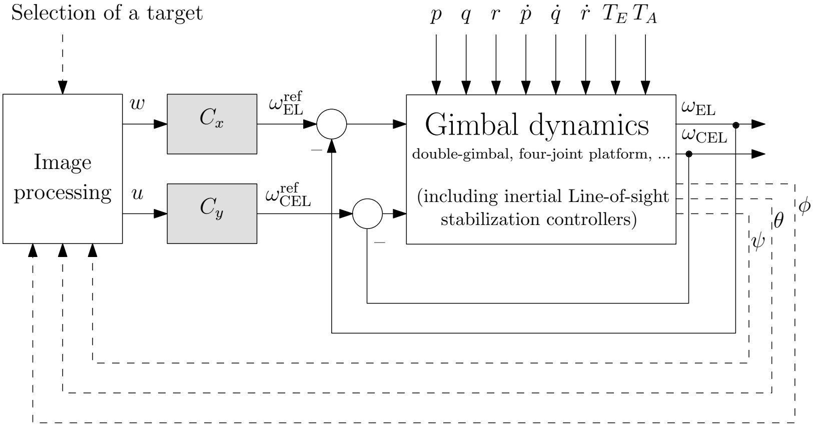

Selection of a target
p q r $\dot{p}$ $\dot{q}$ $\dot{r}$ T_E T_A
Image processing
w
u
C_x
C_y
$\omega_{\mathrm{EL}}^{\mathrm{ref}}$
$\omega_{\mathrm{CEL}}^{\mathrm{ref}}$
$-$
$-$
Gimbal dynamics
double-gimbal, four-joint platform, ...
(including inertial Line-of-sight stabilization controllers)
ω_{EL}
ω_{CEL}
ϕ
θ
ψ

Figure 5.4: Naive pointing-tracking system formed by two SISO loops closed around two inertial rate stabilization loops. The dashed lines are not signals truly fed back to the pointing-tracking controller. These are variables representing the orientation of the camera which affects the position and orientation of objects in the image plane.

be active at low frequencies. This straightforward but naive solution is in Fig. 5.4.

Insisting on decoupled controllers is plausible from an implementation viewpoint. There is a trick hidden here, though, as seen in Fig. 5.5. The best way for explanation is using a double-gimbal platform. When the automatic computer vision tracker detects a regulation error in the horizontal direction in the image plane while seeing no error in vertical direction, the simple cascaded structure of Fig. 5.4 would command the azimuth motor only. This motor alone, however, cannot create a purely horizontal motion in the image plane when $\theta \neq 0$. A geometric explanation can be found in Fig. 4.1: to steer the camera such that the image of an object moves horizontally in an image plane, one would need to command the cross-elevation inertial rate ω_{CEL} (denoted as ω_{Ez} in the figure). However, the motor can only affect the component of the inertial rate in the direction of the azimuth motor axis, that is, ω_{Az}. As soon as there is some misalignment between the two, that is, when the camera is tilted up or down to the ground while the aircraft is in level flight ($\theta \neq 0$), the vector oriented in the azimuth motor axis of length ω_{Az} has some nonzero projection ω_{Ex} to the camera optical axis. Consequently, some unwanted rotation of the image as well as vertical displacement are introduced. Curvilinear coordinate mesh in Fig. 5.5 is generated by the nonlinear dynamics (5.10).

However, with sampling rate of the outer (image-based pointing-tracking) loop fast enough, the error introduced by the coupling between the two camera axes would be corrected in the very next step, when the regulation error in vertical direction in the image plane is detected and a correcting command to the elevation motor would be sent. The currently implemented prototype system achieves sampling rate of 15 Hz, which seems enough to justify this naive approach. Having scanned the available literature, the authors can only suspect that some of the available commercial systems follow this approach too. The

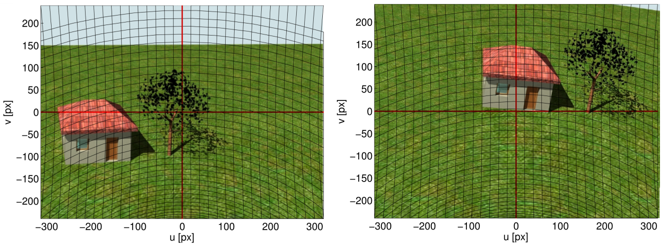


Figure 5.5: Illustration of how in an attempt to steer the camera such that the image of the roof of the house gets back to the middle of the field of view using azimuth motor only, the introduced rotation of the camera around its optical axis makes the horizontal movement curved. Consequently, correction in vertical direction using the elevation motor is needed. Curvilinear coordinate system in the image plane corresponds to the initial elevation of camera by $\theta = -54°$ with respect to the body of the aircraft.

motivation is to improve this scheme, because a bit more advanced and computationally intensive computer vision algorithms can slow down the sampling rate of the outer loop to something like 1 or 2 s.

5.4 FEEDBACK LINEARIZATION BASED VISUAL POINTING AND TRACKING

The key idea for an improvement described in the rest of this chapter is that the curvature of the coordinate axes as in Fig. 5.5 can be compensated for by measuring the third component of the inertial angular rate of the camera body, the one along its optical axis, the so far unused measurement ω_{Ex} (or ω_{Ix}). It is available at the sampling rate a few orders of magnitude faster than what the computer vision system provides. Using this information, exact feedback linearization can be implemented in the controller following standard techniques from image-based visual servoing introduced next.

The idea behind image-based visual servoing is that an error "sensed" in the image plane by the image tracker as

$$e(t) = s(t) - s^{\text{ref}} \tag{5.15}$$

can be eliminated by commanding a proper value of $\xi(t)$, which characterizes the velocity of the camera frame. Note that $s^{\text{ref}} = 0$ when the task is to bring the image of the object into the central position in the image plane by pushing $s(t) = [u(t), w(t)]^T$ to zero. How to find a proper ξ? Simply by inverting the interaction matrix L. In the case of a single-point feature, the matrix is 2×6, which suggests that such a solution will not be unique. Which one to pick? It will be shown shortly that there is one important constraint here which makes only one solution acceptable.

In contrast to common robotics tasks, here we cannot influence the translational position of the camera frame (unless there is a bidirectional communication between the

UAV autopilot and the inertial stabilization & visual tracking system). Hence the linear velocity $v_C = [v_{Cx}, v_{Cy}, v_{Cz}]^T$ of the camera coordinate origin needs to be taken as given (enforced) from the outside. But then the task of determining ξ at a given time instant consists in solving the linear system (5.7) with the term corresponding to translation moved to the right-hand side

$$L_\omega(u,w)\omega_C = \dot{s} - L_v(u,w,z)v_C. \tag{5.16}$$

The 2×3 matrix L_ω has a 1-dimensional nullspace parameterized by

$$\mathcal{N} = \{\omega_c = k \begin{bmatrix} u & w & \lambda \end{bmatrix}^T\}, \tag{5.17}$$

which can be interpreted quite intuitively: rotating the camera about the line connecting the observed point and the origin of the camera frame does not contribute to a change of the coordinates of the point in the image plane. With the right pseudoinverse of L_ω given by

$$L_\omega^\sharp = \begin{bmatrix} 0 & \frac{\lambda}{\lambda^2+u^2+w^2} \\ -\frac{\lambda}{\lambda^2+u^2+w^2} & 0 \\ \frac{w}{\lambda^2+u^2+w^2} & -\frac{u}{\lambda^2+u^2+w^2} \end{bmatrix}, \tag{5.18}$$

all solutions are parameterized by a single constant k

$$\omega_C = L_\omega^\sharp \dot{s} - L_\omega^\sharp L_v v_C + k \begin{bmatrix} u & w & \lambda \end{bmatrix}^T. \tag{5.19}$$

Substituting and abusing k since it is arbitrary we get

$$\omega_C = \frac{1}{z\left(\lambda^2+u^2+w^2\right)} \begin{bmatrix} \lambda^2 v_y - \lambda v_z w + \lambda \dot{w} z + ku \\ -\lambda^2 v_x + \lambda v_z u - \lambda \dot{u} z + kw \\ -\lambda u v_y + \lambda w v_x - u \dot{w} z + \dot{u} w z + k\lambda \end{bmatrix}. \tag{5.20}$$

What we have obtained so far is a procedure which for given velocities $\dot{u}$ and $\dot{w}$ of a point feature in the image plane computes the required angular velocity vector ω_C (it does not hurt now to use the full notation $\omega_{R,C}^C$) of the camera. A single arbitrary parameter k can be used to give some choice, which is the key idea to be exploited next.

5.4.1 Simple proportional image-based pointing and tracking

In order to pull $s(t)$ to the vicinity of $(0,0)$ in the image plane, a cascade control structure can be used: the pointing-tracking controller sets the reference rate vector $\dot{s}^{\text{ref}}(t)$ such that its actual value $s(t)$ goes to zero. One simple approach is to require exponential stability, that is, both $u(t)$ and $w(t)$ go to zero values according to

$$\dot{s}(t) = As(t), \tag{5.21}$$

where A has nonnegative eigenvalues. The simplest solution can be obtained by restricting A to a diagonal matrix $A = -\alpha I$ for some real positive α and then

$$\dot{s}(t) = -\alpha s(t). \tag{5.22}$$

The larger the α, the faster the error in the image plane goes to zero. Practical considerations of the choice of this parameter are discussed at the end of this section. Now, how can we force this error to evolve as in (5.22)? Noting that $\dot{s}(t)$ is related to the camera inertial angular velocities according to (5.20), we can conclude that asymptotically stable image error is guaranteed if the camera inertial velocities follow the reference value

$$\omega_C^{\text{ref}} = \frac{1}{z\left(\lambda^2+u^2+w^2\right)} \begin{bmatrix} \lambda^2 v_y - \lambda v_z w - \lambda\alpha w z + ku \\ -\lambda^2 v_x + \lambda v_z u + \lambda \alpha u z + kw \\ -\lambda u v_y + \lambda w v_x + k\lambda \end{bmatrix}. \tag{5.23}$$

It is not clear at this moment whether and how such a rotation rate of the camera can be established by the two motors. It is the free parameter k that can help to pick such a reference inertial velocity vector ω_C^{ref} of the camera that is realizable by platform motors.

5.4.2 Establishing the camera inertial rate using two motors

Once we know the required inertial angular rate of the camera, what remains is to express it via constant transformation R_C in elevation gimbal frame [E] (or inner gimbal frame [I] for the case of four-gimbal platform)

$$\begin{aligned} \omega_E^{ref}\left(=\omega_I^{ref}\right) &= R_C \omega_C^{\text{ref}} \\ &= \frac{1}{z\left(\lambda^2+u^2+w^2\right)} \begin{bmatrix} -\lambda u v_y + \lambda w v_x + k\lambda \\ -\lambda^2 v_y + \lambda v_z w + \lambda\alpha w z - ku \\ \lambda^2 v_x - \lambda v_z u - \lambda\alpha u z - kw \end{bmatrix}. \end{aligned} \tag{5.24}$$

The task for the inertial angular rate control system is to follow this velocity by commanding all gimbals. It is important to keep track of the corresponding frames. The resulting ω_E^{ref} (or ω_I^{ref}) is fully labeled as $\omega_{R,E}^{E,\text{ref}}$ (or $\omega_{R,I}^{I,\text{ref}}$) as it gives the required inertial rotation rate of the inner gimbal. Its true value is measured by the three-axis MEMS gyro fixed to the elevation gimbal [E] (or inner gimbal [I]).

Now comes the key part. Having only two degrees of freedom (ω_{EL}^{ref} and ω_{CEL}^{ref}), it is not possible to set all the three components of the vector of inertial angular velocity independently. But the free scalar parameter k can be used to pick a specific triple that requires no change with respect to the current value of ω_{ROT} (inertial angular rate of the camera around its optical axis — also denoted by ω_{Ix} or ω_{Ex}). The value of ω_{ROT} must be available to the controller then. Solving (5.24) for the value of k guaranteeing that the x component of the vector on the right is equal to the measured ω_{ROT} gives

$$k = -wv_x + uv_y + \frac{z\left(\lambda^2+u^2+w^2\right)}{\lambda}\omega_{ROT}. \tag{5.25}$$

Substituting this value back to the expressions for the other two components of the reference

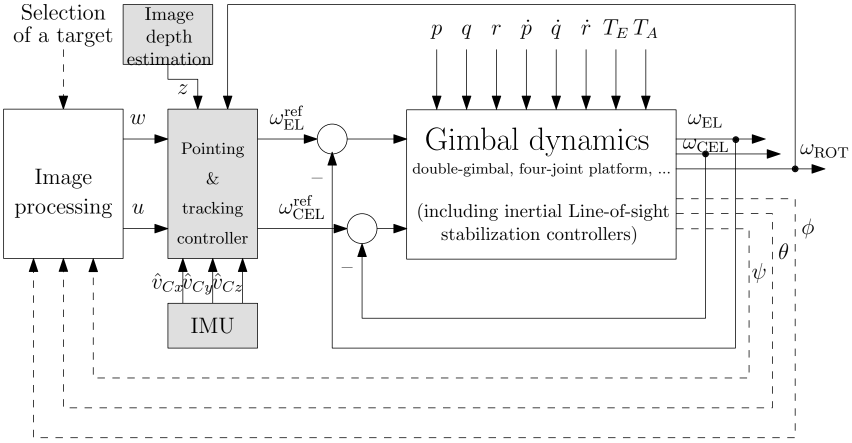

Selection of a target
Image depth estimation
z
Image processing
w
u
Pointing & tracking controller
$\hat{v}_{Cx}$ $\hat{v}_{Cy}$ $\hat{v}_{Cz}$
IMU
$\omega_{\mathrm{EL}}^{\mathrm{ref}}$
$\omega_{\mathrm{CEL}}^{\mathrm{ref}}$
$-$
$-$
p q r $\dot{p}$ $\dot{q}$ $\dot{r}$ T_E T_A
Gimbal dynamics
double-gimbal, four-joint platform, ...
(including inertial Line-of-sight stabilization controllers)
ω_{EL}
ω_{CEL}
ω_{ROT}
ϕ
θ
ψ

Figure 5.6: Full feedback system with an image-based pointing controller aware of the angular rate about the optical axis and the translational motion. The pointing-tracking controller implements (5.26) and (5.27).

angular rate vector, the expressions for the controller outputs follow

$$\omega_{\mathrm{EL}}^{\mathrm{ref}} = \frac{\alpha w \lambda}{\lambda^2 + u^2 + w^2} - \frac{\omega_{ROT} u}{\lambda} - \frac{\lambda^2 v_y - \lambda w v_z - u w v_x + u^2 v_y}{z(\lambda^2 + u^2 + w^2)}, \tag{5.26}$$

$$\omega_{\mathrm{CEL}}^{\mathrm{ref}} = -\frac{\alpha u \lambda}{\lambda^2 + u^2 + w^2} - \frac{\omega_{ROT} w}{\lambda} + \frac{\lambda^2 v_x - \lambda u v_z - w u v_y + w^2 v_x}{z(\lambda^2 + u^2 + w^2)}. \tag{5.27}$$

The expressions (5.26) and (5.27) for the controllers are structured such that three terms can be immediately recognized in each controller: a term corresponding to a regulation error in the corresponding axis as seen in the image plane, a term compensating for the rotation around the camera optical axis and finally a term attenuating the influence of mutual translational motion of the camera and the ground object.

In order to get an insight into this new controller and compare it with the originally proposed decoupled one, consider again the easy situation when the carrier is in level flight and the camera is pointing towards the horizon ($\theta = 0$). Neglect the translational velocities v_C. The observed object is vertically centered in the image plane, that is, $w = 0$. The expressions in (5.26) and (5.27) simplify to

$$\omega_{EL}^{\mathrm{ref}} = 0, \tag{5.28}$$

$$\omega_{CEL}^{\mathrm{ref}} = -\alpha \frac{\lambda}{\lambda^2 + u^2} u. \tag{5.29}$$

Compare this simplified controller and the model of the system (5.12) valid for the same conditions. Apparently, the nonlinear term $\lambda/(\lambda^2 + u^2)$ serves just to invert the nonlinearity in the model. And this is what the controller does in general. It inverts the nonlinearity. In other words, it performs the feedback linearization. Consideration of the inertial angular rate ω_{ROT} of the camera around its optical axis is another measure that the controller takes to invert the nonlinearity. For the maximum zoom ($\lambda = 42mm$), the nonlinear term $\lambda/(\lambda^2 + u^2)$ is sufficiently close to λ and therefore the controller's action is driven by

$$\omega_{CEL}^{\text{ref}} = -\alpha\frac{1}{\lambda}u. \tag{5.30}$$

The simplification can take place even in a more general situation $u, w \neq 0$ but small, and λ large (and v_C still neglected). The general expression for the controller output then reduces to

$$\omega_{EL}^{\text{ref}} = \frac{\alpha w}{\lambda} - \frac{\omega_{ROT}u}{\lambda} \tag{5.31}$$

$$\omega_{CEL}^{\text{ref}} = -\frac{\alpha u}{\lambda} - \frac{\omega_{ROT}w}{\lambda} \tag{5.32}$$

This reduced controller reveals the key enhancement with respect to the fully decoupled design: the controller output contains contribution from the angular rate of the camera around the optical axis!

5.4.3 Summary of controller structure for pointing and tracking

The feedback-linearizing pointing-tracking controller in (5.26) and (5.27) does not preserve the decoupled structure (no longer two separate pointing-tracking controllers). Each of the two controllers accepts not only both the "measured" position errors, that is, u and w, but also

1. ω_{ROT} describing the inertial angular rate of the camera around the optical axis,
2. estimates of the aircraft translational velocity with respect to the ground, expressed in the camera coordinate frame (v_{Cz} describes how fast the camera is approaching the target),
3. an estimate of depth z of the image, that is, the distance from the camera to the ground target.

Moreover, the technical parameter that the controller must be aware of is the focal length λ. An upgrade of the naive scheme proposed in Fig. 5.4 can thus be seen in Fig. 5.6.

The key challenge in implementing this controller fully is in providing the controller with the extra measurements and/or estimates of the three components of the translational velocity v_C and the distance z to the object. These could be approached using inertial measurement unit (IMU) in combination with a laser range-finder and possibly also in combination with a computer vision system. For instance, the depth z and the "towards the object" velocity v_{Cz} is sometimes estimated from the apparent size of an object in the image (covering the image of the object by some polygon and computing its area, which is suggested in [13]). This technique can turn out of limited use here, though, because the images of observed objects can span just a few pixels and determination of v_{Cz} is then very inaccurate.

On the other hand, these new ”complications” caused by the requirements of measuring the translational velocities are not really new and tied to the proposed control scheme. They are equally valid even with the (naive) decoupled control. Unless the translation velocity is known, one simply cannot distinguish whether the image is moving due to undamped aircraft oscillations or because the aircraft is approaching the object. But now, with the systematic analysis documented in this chapter, the structure of the ideal controller is known. It is up to an engineer to decide whether or not to ignore the translational motion and regard its effects as an unmeasured disturbance. Such disturbance is only significant at low frequencies and can be left for the image-based pointing loop to attenuate.

5.4.4 Practical considerations for setting the image dynamics

It is well known that the basic version of image-based visual servoing can be inefficient in terms of a requested manipulator movement needed to establish the requested image features trajectory. Several approaches have been proposed in the literature, see [63], [64], [65], that tackle the problem by separating the rotational and translational motion around and along the optical axis from the remaining controlled degrees of freedom. These issues have not been studied in the present situation since the only two controlled degrees of freedom are two rotations. The defective behavior of *camera retreat* is not present.

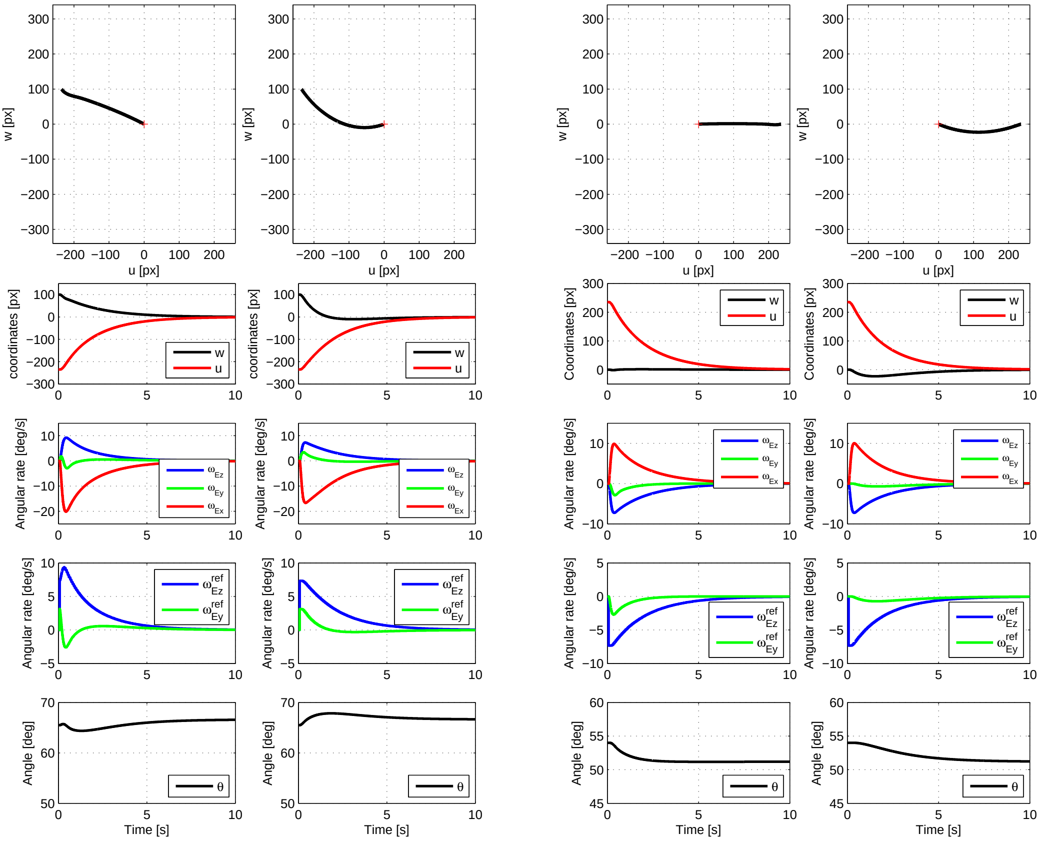
On the other hand, there is one practical aspect of setting the dynamics in the image plane that must be taken into consideration: placing the eigenvalues of the linearized system (5.22) too far in the left half plane makes the motion response of the system too fast for the image tracker. The image features then travel so fast in the image plane that the image tracker loses the grasp of the object (most image tracking algorithms such as [66] explore the nearest neighborhood only).

5.5 NUMERICAL SIMULATIONS

Closed-loop responses with the two controllers were simulated for double-gimbal platform. The image of the observed object was initially located out of the center of the image frame. The control goal is to bring the observed object into the center of the field of view.

Three simulations were run (always with both controllers): Two with the sampling rate 15 Hz for different initial locations of the image of the object. And one for the slower sampling at 2 Hz. Results are visualized in Fig. 5.7(a), Fig. 5.7(b) and 5.8.

Unlike in the design stage, in the simulations the inner (inertial rate) loops are not assumed to work ideally. That is, for the purpose of simulations their transfer functions are not identically equal to 1. The two SISO inertial rate loops are just standard feedback interconnections of a first-order system and P or PI regulator. Hence they can be modeled by low pass filters with a given bandwidth. However, the laboratory experiments described in the section 2.2.9 (see Fig. 2.11) reveal that this bandwidth depends on the magnitude of the step in the reference inertial rate. This behavior is a consequence of a constraint on the amplitude of the control voltage inside the rate loops; a nonlinear model should be used to



(a) Simulation 1: Responses of relevant quantities for $\theta(0) = 65.5°$, tracker sampling rate $f_{\mathrm{sp}} =$ 15Hz, $\alpha = 0.46$. Left: the proposed algorithm, right: the original decoupled approach.

(b) Simulation 2: Responses of relevant quantities for $\theta(0) = 54°$, tracker sampling rate $f_{\mathrm{sp}} = 15$Hz, $\alpha = 0.46$. Left: the proposed algorithm, right: the original decoupled approach.

Figure 5.7: Numerical simulations for two different initial conditions.

describe the rate loop more precisely. However, in order to obtain just a rough estimate of the system response, a low-order low pass filter seems satisfactory. The laboratory experiments with a real device then give a true assessment of the system performance.

No translation between the carrier and the observed object was assumed. The only physical parameters are those of the optics: focal length $\lambda = 42$ mm and resolution of the camera CCD chip is 640×480 pixels (u and w were scaled to pixels for visualization purposes).

As discussed in the section devoted to the image tracker, the process of extracting the pointing&tracking error from the visual information devours some computational time, hence u and w are only updated with the sampling rate ranging from 0.5 to 15 Hz, depending on the complexity of the algorithm. For simulations we choose 15 and 2 Hz. In addition, the

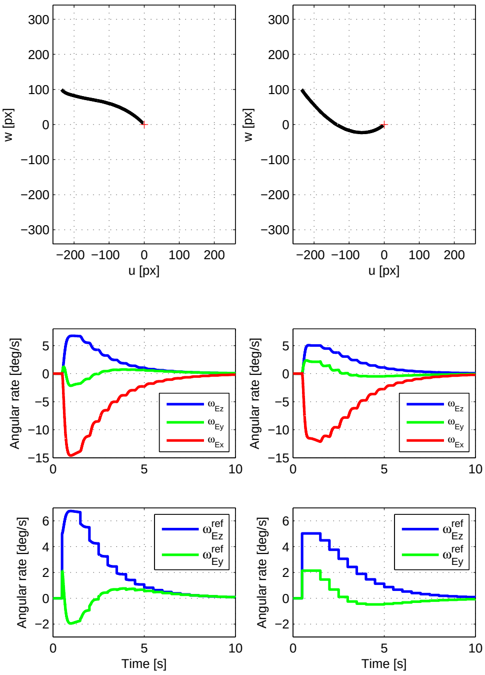


Figure 5.8: Simulation 3: Responses of relevant quantities for $\theta(0) = 65.5°$, tracker sampling frequency $f_{\mathrm{sp}} = 2\,\mathrm{Hz}$, $\alpha = 0.31$. Left: the proposed algorithm, right: the original decoupled approach.

data is always available to the controller with a delay of one sampling period. The sampling rate of the inner loop is up to two orders of magnitude higher (200 Hz, the maximum provided by the inertial angular rate sensors used in the project).

The simulation results confirm that the new controller struggles to follow a linear path in the image plane. Not only is the linear behavior easier to analyze but also is more plausible for a human operator. Some deterioration of a perfectly linear path in the image plane is visible. This is due to the nonideal inertial rate loops that were considered in the simulations.

The third simulation with the slower image tracker sampling rate of 2 Hz demonstrates that the new proposed algorithm updates the reference values for camera inertial rates at a faster sampling rate than the original decoupled controller.

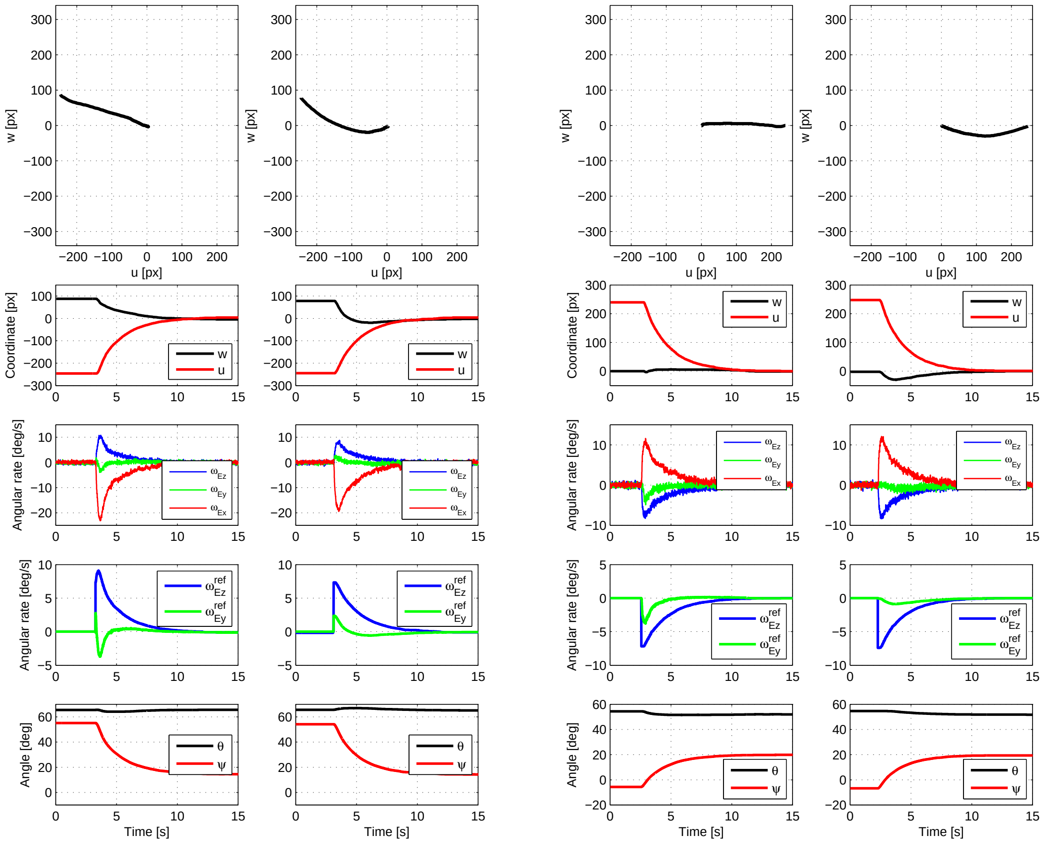
5.6 LABORATORY EXPERIMENTS

The benchmark system — the double-gimbal platform H240 — was used to validate the functionality of the proposed control scheme and compare its performance with the intuitive decoupled controller. The experimental test was conducted in an indoor lab while the camera platform was carried by a fixed laboratory stand. Therefore it was only the objects on the ceiling rather than on the floor that could be tracked conveniently. Both the new and the original (naive) decoupled controllers were tested only for the faster sampling rate of 15 Hz of the automatic image tracker.

Three experiments were conducted and the measurements are visualized here in Figs. 5.9(a), 5.9(b) and 5.10. The experimental data for the new algorithm are always on the left and the data for the original decoupled scheme on the right.

The first two experiments were quite similar: the platform was in both cases sitting peacefully on the desk and the only difference was the initial elevation of the camera ($\theta(0) = 65.5°$ and $\theta(0) = 54°$ as in the simulations) and the position of the point to be tracked in the image plane. The measurements are in Fig. 5.9(a) and Fig. 5.9(b) and can be easily compared with the simulation results in Fig. 5.7(a) and Fig. 5.7(b). Apparently, the responses for both control methodologies are quite similar as for the time scale and control magnitude (but see the difference in one of the controller outputs $\omega_{Ey}^{\mathrm{ref}}$ in Fig. 5.9(a)). The key difference is that the new algorithm achieves a linear trajectory in the image plane as desired.

The third experiment validates the pointing and tracking performance even in presence of a disturbing rotational motion of the carrier. Namely, the optical axis of the camera was initially pointing to the ceiling with the elevation $\theta(0) = 70°$ and the laboratory stand was rotated manually around its vertical axis (orthogonal to the surface of the desk). The measured outcomes are in Fig. 5.10 both for the new algorithm and for the decoupled scheme, both of which use the same inner (inertial angular rate stabilization) feedback loop. The conclusion is that both algorithms exhibit the same characteristic behavior already known from the simulations and static experiments. Actually, the measurements related to the new controller appear a bit more disturbed but this is only due to the fact that the disturbing motion was induced manually, hence a bit differently in both cases.



(a) Experiment 1: Responses of relevant quantities for $\theta(0) = 65.5°$, tracker sampling rate $f_{\text{sp}} = 15$Hz, $\alpha = 0.46$. Left: the proposed algorithm, right: the original decoupled approach.

(b) Experiment 2: Left: the proposed algorithm, right: the original decoupled approach. The initial elevation angle was set to $\theta(0) = 54°$. The image tracker sampling rate $f_{\text{sp}} = 15\,\text{Hz}$. The controller parameter $\alpha = 0.46$.

Figure 5.9: Numerical simulations for two different initial conditions.

5.7 EXPERIMENT ON A HELICOPTER

Experimental verification of pointing&tracking was also tested with the four-gimbal dual-stage platform S250. The platform was mounted underneath MI-17 helicopter during two test flights (see Fig. 5.11). The human operator, sitting inside the helicopter, was operating the platform using the touchscreen device (see Fig. 5.11 on the right). This device allows manipulating the line-of-sight using joystick, zooming the camera, viewing the output video from the camera so as the current map with GPS coordinates, and of course specifying the target to be tracked by clicking on it. After specifying the target, it is pulled to the center of the camera view exactly as studied in sections on numerical simulations and laboratory experiments. This proves the practical applicability of chosen laboratory experiments.

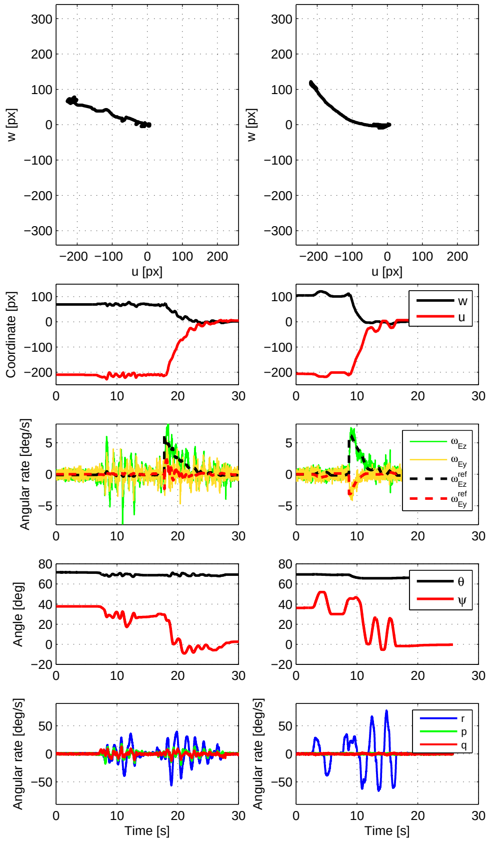


Figure 5.10: Experiment 3: Responses of relevant quantities for the pointing exposed to external (recorded) disturbances p, q, r. $\theta(0) = 70°$, tracker sampling rate $f_{\mathrm{sp}} = 15\mathrm{Hz}$, $\alpha = 0.46$. Left: the proposed algorithm, right: the original decoupled approach.

Print screen pictures of the camera video from the system during operation are shown in Fig. 5.12. Some of these videos are included as an attachment in CD-ROM in the booklet of this thesis (see the list of attached videos at the end of this thesis).

5.8 CONCLUSIONS

This chapter presented a systematic procedure for designing and implementing a pointing and tracking image-based controller for an airborne camera platform with an inertial line-of-sight stabilization already designed and implemented. The proposed scheme uses an extra information from an inertial angular rate sensor; namely, the angular velocity of the payload (camera, laser) around its optical axis. This extra measurement is provided by a MEMS gyro at a much faster sampling rate than the pointing-tracking error produced by a computer vision system. Moreover, the proposed controller can take into consideration

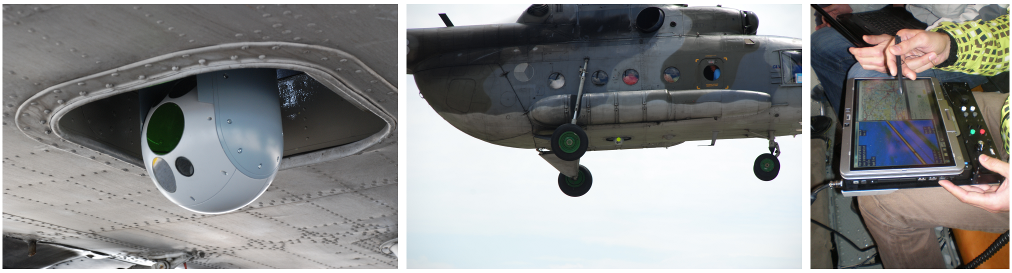


Figure 5.11: Left and Middle: Helicopter MI-17 with the platform S250 mounted underneath, Right: Touchscreen device used for operating the stabilized platform.

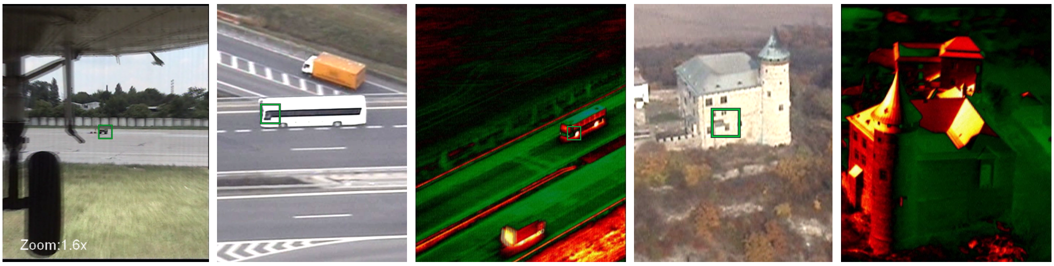


Figure 5.12: Several screenshots created from the recorded video of a target tracking during the flight experiment. Specifying the target is provided by a clicking the touchscreen device. The locked target is denoted by the green square — the aim of the control system is to keep this square (the target) in the center of the image.

the measured or estimated translation velocity of the aircraft with respect to the observed target (to compensate for the paralactic phenomenon).

The essence of the proposed design technique is that of a feedback linearization. The resulting controller enforces linear dynamics in the image plane. This not only makes the analysis and design systematic but putting it on the well-explored ground, but also makes the response of the system a bit more friendly for a human operator as the system follows linear paths in the image plane during (re)pointing.

The proposed scheme was thoroughly simulated and verified by practical laboratory experiments with a realistic benchmark system and compared against the more intuitive decoupled control scheme. Possible simplifications were discussed and practical pitfalls were highlighted.

Chapter 6

Delay compensation in visual servoing (for aerial surveillance)

This chapter presents a few simple techniques for compensation of a one-sampling-period delay in a slow-sampled outer (position or angle) loop within a cascade visual servomechanism that also includes a fast-sampled inner (velocity) feedback loop. The results are mainly relevant for visual servoing applications, since the velocity sensors such as tachometers or MEMS-based gyros usually achieve much higher sampling rates compared to computer vision systems used as position (or orientation) sensors. The proposed solutions only compensate for the motion of the camera and not the observed object; they are particularly useful when the visual servoing is combined with inertial stabilization. The problem is solved using two different formalisms: first, the problem is cast as an instance of a reset system with periodic resetting of the observer state. Second, a technique based on the concept of a modified Smith compensator is proposed wherein the undelayed output of a mathematical model is replaced by the measured rate signal from the inner loop. Numerical simulations are used to highlight the behavior of the proposed algorithms. Finally experimental results obtained with a real double gimbal camera system are presented.

The content of this chapter is based on the conference paper [5] and on the submitted version of the journal paper [8].

6.1 INTRODUCTION

6.1.1 Definition of the problem

A common control system configuration for motion control applications is that of a two-level cascade. Such a control structure was thoroughly studied in chapter 3 of this thesis.

The class of cascade systems that are considered in this chapter exhibits major disproportions between sampling rates of the inner and outer loops. Moreover, the outer loop contains a full one-sampling-period delay. This situation is typical of visual servoing

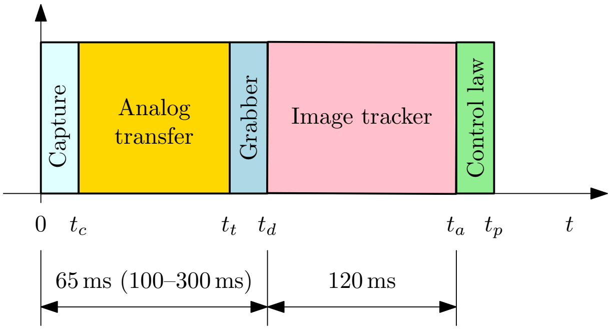


Figure 6.1: Composition of the total delay in the visual pointing and tracking feedback loop: t_c is the time needed to capture the image ($5-20$ ms), which depends on the scene brightness; $t_t - t_c$ is the time for an analog transfer of the image (40 ms); $t_d - t_t$ is the time for digitization by a grabber. These three activities together can take about 65 ms when using the proprietary drivers provided with the grabber, and $100-300$ ms when using DirectX technology. The next tasks finishing at t_a is the actual visual tracking algorithm (120 ms on a common PC platform). The final task finishing at t_p is responsible for the pointing and tracking control law and consumes a negligible computational time. The sampling period is then given by the total delay. Alternatively, another image capturing can start right after t_d.

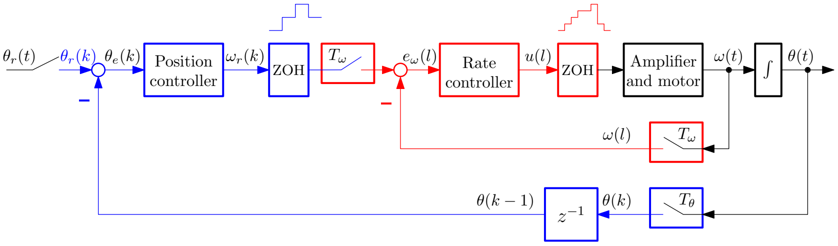


Figure 6.2: Case A: Problem definition — Cascade visual servoing structure: The inner (velocity, rate) loop works at a fast sampling rate. The outer position (or pointing) loop works at a slow rate and suffers from a one-sampling-period delay.

applications, where the role of the position error sensor in the outer loop is played by a camera accompanied with a computer vision system. The algorithms used to extract information from the captured video frames usually devour some computational time, which not only sets the sampling rate for the outer loop relatively slow but also enforces the one-sampling-period delay. With no modifications of the control scheme, the position controller literally *tracks the past values*. The detailed composition of the total delay in the visual pointing and tracking feedback loop is depicted in Fig. 6.1. The block diagram for the problem is in Fig. 6.2.

Some delay-compensation schemes have been proposed in the literature for the general visual servoing setup. A particular research motivation comes through a development of a control system for airborne camera platforms wherein the cameras are exposed to unwanted yet measured motions of the carrier. The inner feedback loop sampled at 200 Hz uses MEMS (Coriolis) gyros to measure the inertial angular rate of the optical payload and struggles to make the deviation from a reference rate small. The outer feedback loop uses an industrial PC to compute the pointing error by finding the deviations in the image plane of the camera, and keeps this error small by commanding the reference value for the inertial angular rate. The computation carried by the image recognition system takes about 1/10 s, but for more advanced computer vision algorithms the computation can easily take up to 2 s.

6.1.2 Notation

The ideas are communicated mainly using block diagrams. The sampling periods of the signal paths should be obvious: the inner loop works at T_ω sampling period, the outer loop at T_θ. In addition, colors are used to help distinguish between continuous-time, fast sampled and slow sampled discrete signals, using black, red and blue colors, respectively. A single z operator is abused for both sampling periods to avoid notational clutter. Similar (and common) abuse is that the same names are used for the original continuous-time variables such as $\theta(t)$ and their sampled versions such as $\theta(k)$ in place of a more proper $\theta(t_k)$ or $\theta(kT)$. The subsystems in block diagrams use the labels of G for the system, K for the rate controller in the inner loop and P for the position (or pointing) controller in the outer loop. The hat is used to denote an estimate such as $\hat{\theta}(k)$.

6.2 INTUITIVE WAY OF DELAY COMPENSATION

Realizing that the position (angle) controller receives measurements that are as old as one full sampling period T_θ, a simple solution comes into an engineer's mind: take an advantage of availability of the angular velocity measurements[1], which are available many times (about 10 up to 500 times) during the slow outer-loop sampling period. Integrating the frequently arriving values over the slow sampling period gives the desired correction that needs to be subtracted from the outdated measurements coming from the image-based orientation sensor as shown in Fig. 6.3. The orientation controller then needs to compensate for a smaller error than the image-based sensor suggests. Two variants of this scheme are possible.

6.2.1 Updating at the slow sampling rate (Case B)

The estimates $\hat{\theta}(t)$ of the angle $\theta(t)$ are only updated once the new measurements from the computer vision subsystem arrive, that is, at times $t_k = kT_\theta$, $k \in \mathbb{Z}$. Then the estimated value is fixed over the whole interval $[t_k, t_{k+1})$. It can be calculated according to the Fig. 6.4

[1] It must be emphasized that it is the angular velocity of the camera that is measured. This scheme is therefore particularly useful when the camera is carried by a mobile carrier.

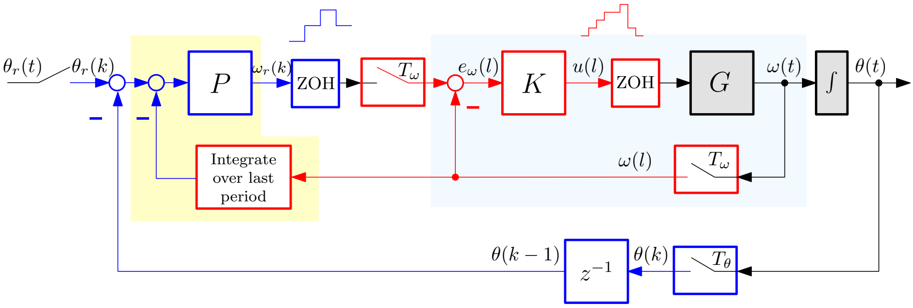


Figure 6.3: Case B: Intuitive solution to the problem of having a slow outer loop with a one-period delay: integrate the (undelayed and fast sampled) angular velocity signal ω over one *slow* sampling period T_θ and subtract from the computed orientation (angle) error once the delayed data $\theta(t_{k-1})$ from the computer-vision based sensor arrive. The two blocks in the yellow region now constitute the position controller.

as

$$\hat{\theta}(t_k) = \theta(t_{k-1}) + \underbrace{\int_{t_{k-1}}^{t_k} \omega(\tau)\mathrm{d}\tau}_{\Delta_{\theta(t_k)}}, \tag{6.1}$$

where $\theta(t_{k-1})$ is a crude estimate of the current angle at t_k taken as the delayed measurement performed at t_{k-1}, and $\Delta_{\theta(t_k)}$ is the integrated rate over the full last sampling period T_θ, which makes the crude estimate more accurate. In reality, the integration should be replaced by a finite sum because the inner loop also works at discrete time. For instance,

$$\hat{\theta}(t_k) = \theta(t_{k-1}) + \sum_{n=1}^{T_\theta/T_\omega} \omega(t_{k-1} + nT_\omega)\cdot T_\omega, \tag{6.2}$$

but other integration schemes can be used, including the noncausal trapezoidal one, because the integration over the previous period is performed once all the samples are available.

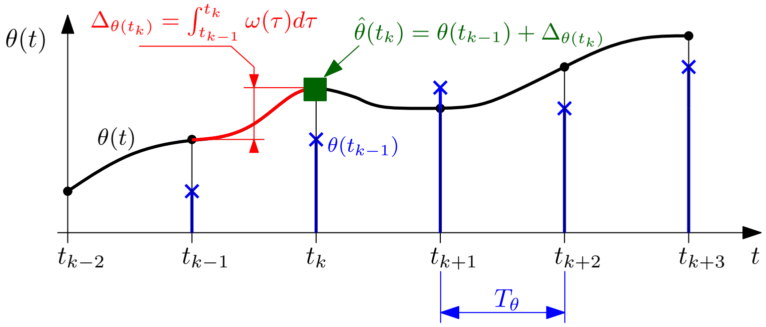
The block diagram is in Fig. 6.3. The rate integrator must be periodically reset at all t_k's, when new position measurements arrive. This can be formally stated as

$$\dot{\Delta}_\theta(t) = \omega(t), \qquad \text{for } t \neq t_k,\ k \in \mathbb{Z}, \tag{6.3}$$

$$\Delta_\theta(t_k^+) = 0, \qquad \text{for } t = t_k,\ k \in \mathbb{Z}, \tag{6.4}$$

where the notation $\Delta_\theta(t_k^+)$ stands for $\lim_{t\downarrow t_k} \Delta_\theta(t_k)$. This estimate is then used for the full next period.

This heuristic solution turns out efficient when implemented on a real experimental system. A scholarly challenge is to see whether and how this intuitive solution can be formulated using formal concepts from control theory. The major motivation is to use such formalization to find a hint for improving the performance even more, perhaps by exploiting some less intuitive property of the problem.



$\theta(t)$
$\Delta_{\theta(t_k)} = \int_{t_{k-1}}^{t_k} \omega(\tau)d\tau$
$\hat{\theta}(t_k) = \theta(t_{k-1}) + \Delta_{\theta(t_k)}$
$\theta(t)$
$\theta(t_{k-1})$
t_{k-2}
t_{k-1}
t_k
t_{k+1}
t_{k+2}
t_{k+3}
t
T_θ

Figure 6.4: Case B: The best estimate $\hat{\theta}(t_k)$ at time t_k is obtained from the last known position measurement $\theta(t_{k-1})$ plus the integral of the gyro signal over the last period (the red curve).

Resetting the velocity integrator is reminiscent of *reset control* methodology for control design for linear systems, which was introduced by [67] in late 1950s. Its development was documented in the recent surveys [68] and [69] and illustrated by the case studies [70] and [71]. The key principle of reset control is that reset of some controller states is triggered by a certain value of the measured signal. In the simplest case, the integrator in a PI controller is reset (set to zero) every time the regulation error signal crosses zero. When properly designed, the reset controller can beat some restrictions imposed on linear systems such as the water bed phenomenon (Bode integral theorems).

To adapt the reset control formalism to the present problem, it is crucial to realize that the integrator here is reset periodically, independently of values of any measured signal. Such situation was studied in [72]. In this thesis, it is the observer that is periodically reset.

6.2.2 Updating at the fast sampling rate (Case C)

In the previous solution, the estimated position $\hat{\theta}(t)$ is only updated at the slow sampling rate $1/T_\theta$. However, with the measurements arriving from the rate sensor at a much faster rate $1/T_\omega$, it is more efficient to update $\hat{\theta}(t)$ at this faster rate. An estimate $\hat{\theta}(t)$ at time t may be calculated as a sum of the just arrived one-sampling-period old position measurement $\theta(t_{k-1})$ and the numerical integral of the gyro signal as explained in Fig. 6.5

$$\hat{\theta}(t) = \theta(t_{k-1}) + \underbrace{\int_{t_{k-1}}^{t} \omega(\tau)d\tau}_{\Delta_{\theta(t)}}. \tag{6.5}$$

To implement this equation one has two options:

1. Reset the integrator every time t_k, when the new measured orientation (angle) arrives. However, do not reset to zero but just subtract $\hat{\theta}(t_{k-1}^+)$ from the integrated value $\hat{\theta}(t_k^-)$ right before the hit of sampling clock. The interval of integration can then be viewed

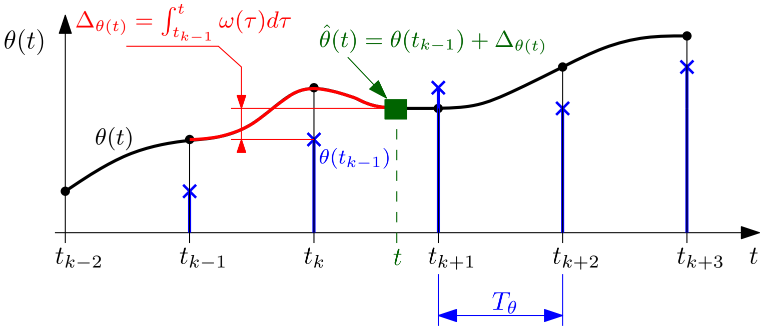

$\theta(t)$
$\Delta_{\theta(t)} = \int_{t_{k-1}}^{t} \omega(\tau)d\tau$
$\hat{\theta}(t) = \theta(t_{k-1}) + \Delta_{\theta(t)}$
$\theta(t)$
$\theta(t_{k-1})$
t_{k-2}
t_{k-1}
t_k
t
t_{k+1}
t_{k+2}
t_{k+3}
t
T_θ

Figure 6.5: Case C: Time diagram explaining the reset system formulation of the observer of angle. The best estimate $\hat{\theta}(t)$ at time t is obtained from last known position measurement $\theta(t_{k-1})$ plus numerical integral of the gyro signal represented by the red curve. The integration interval thus "breathes".

as if *breathing*, that is stretching from one full sampling period to two periods, and shrinking back

$$\dot{\Delta}_\theta(t) = \omega(t), \qquad \text{for } t \neq t_k,\ k \in \mathbb{Z}, \tag{6.6}$$

$$\Delta_\theta(t_k^+) = \Delta_\theta(t_k^-) - \Delta_\theta(t_{k-1}^+), \qquad \text{for } t = t_k,\ k \in \mathbb{Z}. \tag{6.7}$$

2. Use two integrators and reset them *alternately* to zero. Let the two integrator states be $\theta_A(t)$ and $\theta_B(t)$

$$\dot{\Delta}_A(t) = \omega(t), \qquad \text{for } t \neq t_k, \qquad k\,\text{even}, \tag{6.8}$$

$$\dot{\Delta}_B(t) = \omega(t), \qquad \text{for } t \neq t_k, \qquad k\,\text{odd}, \tag{6.9}$$

$$\Delta_A(t_k^+) = 0, \qquad \text{for } t = t_k, \qquad k\,\text{even}, \tag{6.10}$$

$$\Delta_B(t_k^+) = 0, \qquad \text{for } t = t_k, \qquad k\,\text{odd}, \tag{6.11}$$

then

$$\Delta_\theta(t) = \begin{cases} \Delta_A(t), & \text{for } t \in (t_k, t_{k+1}),\ k\,\text{even}, \\ \Delta_B(t), & \text{for } t \in (t_k, t_{k+1}),\ k\,\text{odd}. \end{cases}. \tag{6.12}$$

From the implementation viewpoint, the difference from the case B is that here one more variable $\Delta_\theta(t_{k-1}^+)$ needs to be stored or another buffer must be implemented. Although the first option looks cheaper, the second option will be shown advantageous when discussing the implementation issues for a double gimbal platform in section 6.7.1.

6.3 MODIFIED SMITH PREDICTOR

With delays in the loop, one is directed to the well-known formal technique of Smith predictor (or compensator). The essence of Smith predictor is to include a model of a delayed

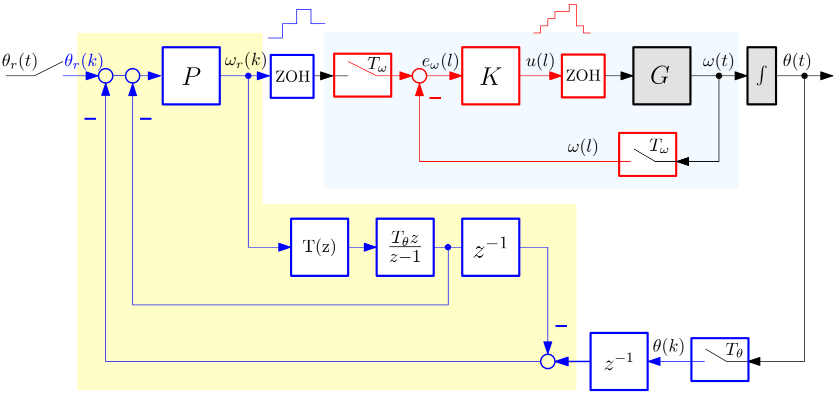


Figure 6.6: Smith compensator used in a cascade visual servomechanism.

system in the controller. Application of this concept to visual servoing is in Fig. 6.6. The inner closed-loop (red color for signals) is described by the transfer function $T(s)$ from the reference velocity ω_r (as produced by the position controller) to the true velocity ω as measured (neglecting the effect of noise and bias for the moment). Ideally this should be close to one, at least within the bandwidth of the velocity loop. Resampling to the slower sampling period T_θ (and abusing the notation by using the same letter T), the transfer function $T(s)$ is viewed as $T(z)$ by the slow controller.

A major deficiency of Smith compensator is its sensitivity to discrepancies between the model and the reality. In particular, if the system is subject to an unmeasured disturbance, the compensator is not aware of it and the performance is deteriorated. Modified Smith predictor has been proposed in literature, see [73, 74, 75], though it is difficult to give a proper credit to its inventor. The key idea is that when some other variable is also measured on the system, why not use it to make the output of the Smith predictor more accurate? In particular, if the rate (velocity) variable is measured for the purpose of rate stabilization in a cascade feedback configuration, why not use it in place of the output of the model $T(z)$ in the Smith compensator as in Fig. 6.7? The position controller then increases its sampling frequency to that of the inner loop. A concise form of Fig. 6.7 is in Fig. 6.8.

Though appearing justifiable, the output of this model does not compensate the delayed slow-sampled system output perfectly. The right model should *downsample* the output of the fast-sampled integrator and multiple-period delay before subtracting it from the delayed slow-sampled system output as visualized in Fig. 6.9. Comparing all the four compensation schemes mentioned so far, that is, B, C, D and E as in Fig. 6.4, 6.5, 6.8 and 6.9, it can be concluded that all the four schemes are based on some kind of fixed-interval integration of the rate signal and they differ in characterization of the interval:

- The intuitive *integrate-over-last-period* solution integrates over the fixed-length interval $[t_{k-1}, t_k)$ and keeps the estimate unchanged for the whole interval $[t_k, t_{k+1})$.
- The modified reset scheme integrates over an interval that stretches from $[t_{k-1}, t_k)$ to

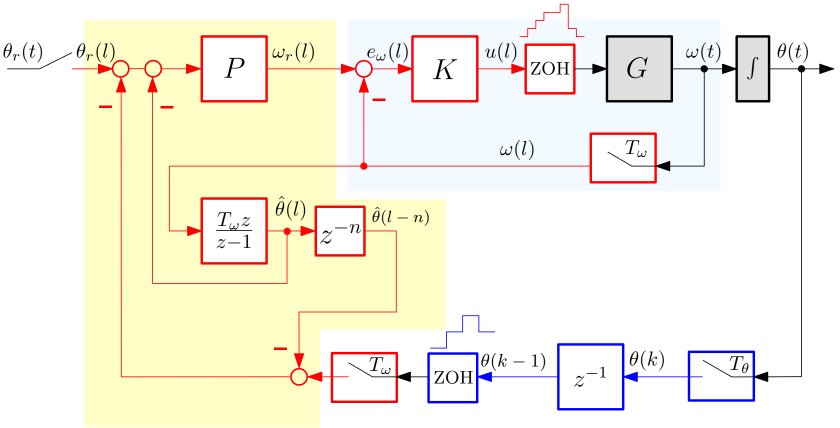


Figure 6.7: Modified Smith compensator. Why relying on a mathematical model when another variable is measured with no delay? The discrete integrator and the delay block use the fast sampling rate. Assume $T_\theta = nT_\omega$.

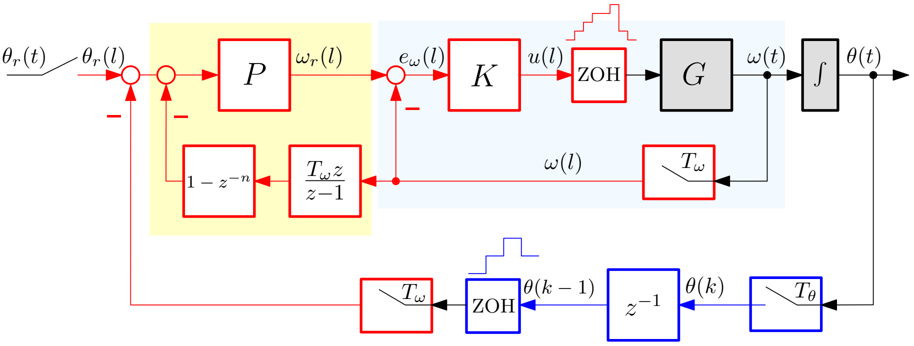


Figure 6.8: Case D: Modified Smith compensator redrawn into a more compact form.

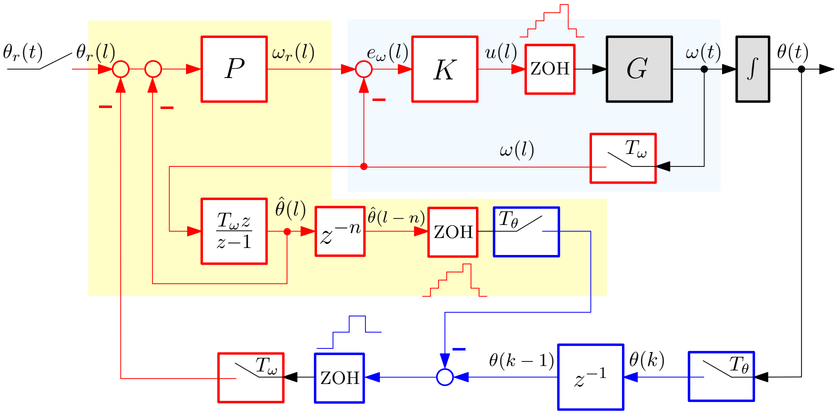


Figure 6.9: Case E: Modified Smith compensator from Fig. 6.8 with added re-sampling term.

$[t_{k-1}, t_{k+1})$ and then shrinks to $[t_k, t_{k+1})$ and repeats this “breathing” pattern.

- The modified Smith compensator running at the fast sampling rate performs the integration over the moving window (interval) of the width T_θ.
- The modified Smith compensator followed by downsampling actually exhibits the same “breathing” integration pattern as observed with the reset observer above. They are equivalent, as can be seen from Fig. 6.9 and after pondering over the integration pattern of the former.

In reality, it is inevitable that the measurement of the rate (velocity) signal is distorted by some constant or slowly varying offset (bias). This is particularly pronounced for inertial measurements performed by cheap MEMS devices with a limited possibility of real-time recalibration. The performance of the schemes D and E (in Fig. 6.8 and 6.9 respectively) is deteriorated in presence of bias, even though not completely unworkable. Since the integrator only integrates the bias over the finite interval T_θ, the estimate cannot completely diverge. However, such finite-horizon integrator must be implemented in a way to avoid an overflow; for instance, surely it cannot be realized as a subtraction of two integrators as suggested in the block diagram in Fig. 6.8.

6.4 MULTIRATE ESTIMATION APPROACHED VIA LIFTING TECHNIQUE

The solution presented in the previous section can be viewed as an estimator that is fed with the input to the inner rate loop, that is, a reference rate ω_r, and the measured output ω corrupted by noise. It computes an estimate $\hat{\omega}$, see Fig. 6.10. The particular estimator in Fig. 6.7 and Fig. 6.8 accomplishes this by directly using the measured value of ω as the estimate $\hat{\omega}$. Knowing the rate controller and the model of the system to be controlled, that is, the motor plus some electronics and gears, standard Kalman filter could be used to obtain an estimate $\hat{\omega}$ of ω. This can alleviate troubles with the measurement noise.

With the notion of an estimation brought to the forefront, a full estimation scheme can also be considered. This is depicted in Fig. 6.11. It assumes that the inner (rate) loop is not closed. The distinguished feature is that it must work with signals sampled with two different sampling periods. The presence of delay in the slower signal calls for an introduction of one extra state. The solution seems attractive since it would account for some more advanced features such as friction compensation via friction force estimation. However, the cascade structure, which might be preferred from an implementation viewpoint, is lost. If there is a strong requirement for keeping the cascade structure, the scheme in Fig. 6.12 can be used.

The field of multirate systems is broad and the tools are fairly advanced. In this section, one particular framework for analysis of multirate systems will be used, namely, the technique of *lifting*, which is described for example in [76] and [77].

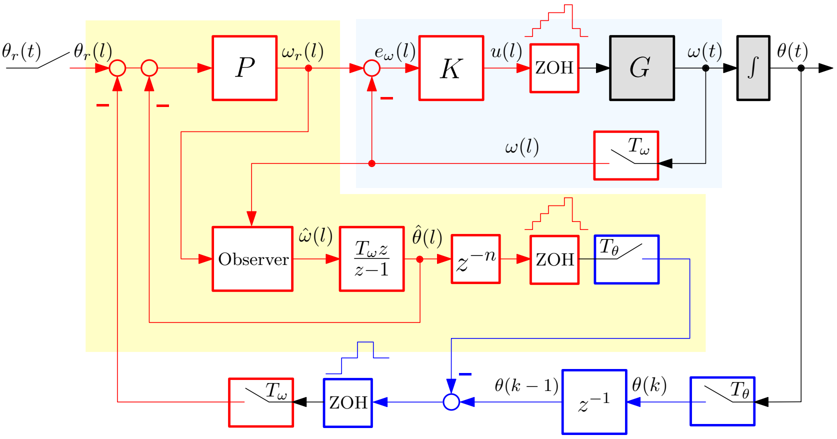


Figure 6.10: Modified Smith compensator with an estimator instead of a pure model.

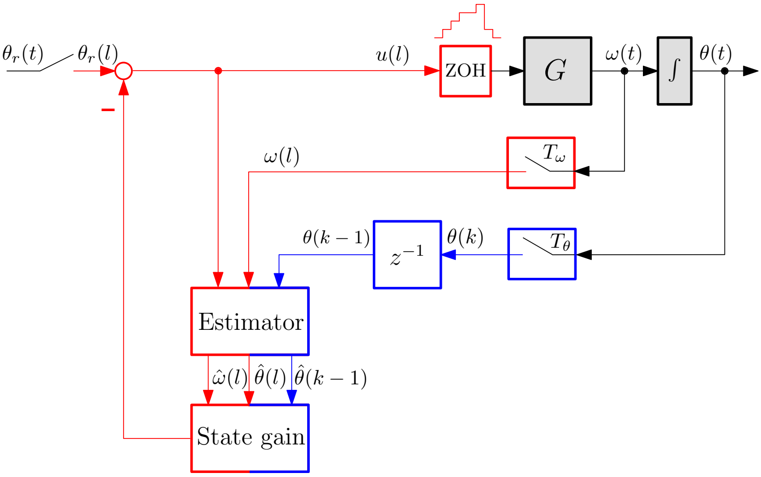


Figure 6.11: Multirate estimation of both the velocity and the position (and possibly some other states such as frictional force, bias of gyros, etc). State feedback can be then used to play the role of both rate and position (pointing) controllers.

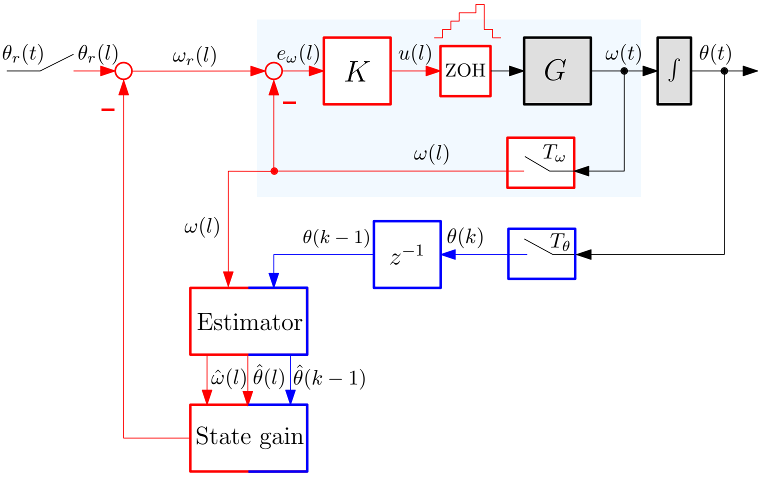
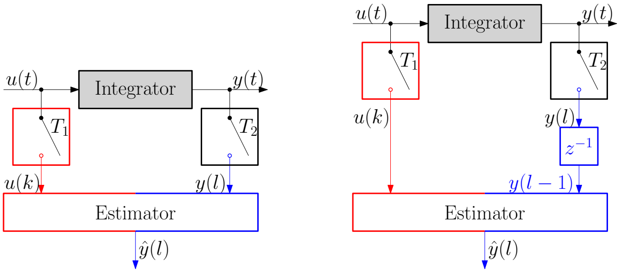


Figure 6.12: Multirate estimation with the inner loop already closed. States of the controller contribute to the full state vector.

As the goal in this section is mainly to gain some insight, the situation will be simplified a bit. The configuration as in Fig. 6.12 will be considered and it will be assumed that the inner loop is tuned so that the transfer function from ω_r to ω is 1. Disregarding the delay for a moment, the scheme then reduces to the one in Fig. 6.13(a). Addition of a measurement delay modifies the scheme to Fig. 6.13(b). In these diagrams and the subsequent text, the more familiar and technology-neutral notation u for the inputs and y for the outputs will be adopted temporarily.



(a) Dual-rate estimation of the state of an integrator.

(b) Dual-rate estimation of the state of an integrator with a delayed measurement.

Figure 6.13: Dual-rate estimation of the state of an integrator without and with a measurement delay.

6.4.1 Lifting of (the inputs of) the discrete-time integrator

The technique of lifting consists of structuring the fast-sampled input into batches of samples; these batches are then considered as arriving at the slower sampling rate. In other words, the dual-rate discrete-time model of an integrator is converted into a single-rate model operating at the slower sampling rate but with the input scalar signal turned into a vector one.

Assume the integer ratio between the slower and faster sampling periods:

$$T_2 = nT_1. \tag{6.13}$$

With the notation $u(k) = u(kT_1)$ for the fast-sampled variable, the samples of the lifted signal $\underline{u}(l) = \underline{u}(lT_2)$ are given by

$$\underline{u}(0) = \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(n-1) \end{bmatrix}, \ \underline{u}(1) = \begin{bmatrix} u(n) \\ u(n+1) \\ \vdots \\ u(2n-1) \end{bmatrix}, \ldots \tag{6.14}$$

A graphical block representation for the lifted integrator is in Fig. 6.14.

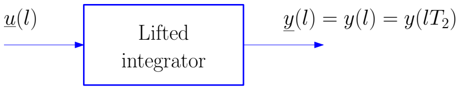


Figure 6.14: Integrator with a lifted input.

For the state-space model of a general discrete-time LTI system given by the quadruple of matrices

$$G = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right], \tag{6.15}$$

its counterpart with a lifted input is given by

$$G_{\text{lifted}} = \left[\begin{array}{c|cccc} A^n & A^{n-1}B & A^{n-2}B & \ldots & B \\ \hline C & D & 0 & \ldots & 0 \end{array}\right], \tag{6.16}$$

For a discrete-time model of an integrator, the state space matrices are given by $A = 1, B = T_1, C = 1, D = 0$ when forward Euler method is used, or $A = 1, B = T_1, C = 1, D = T_1$ when backward Euler method is used. Note that the fast sampling rate T_1 is used here. The lifted integrator is then

$$G_{\text{lifted forward integrator}} = \left[\begin{array}{c|cccc} 1 & T_1 & T_1 & \ldots & T_1 \\ \hline 1 & 0 & 0 & \ldots & 0 \end{array}\right], \tag{6.17}$$

or

$$G_{\text{lifted backward integrator}} = \left[\begin{array}{c|cccc} 1 & T_1 & T_1 & \dots & T_1 \\ \hline 1 & T_1 & 0 & \dots & 0 \end{array}\right]. \tag{6.18}$$

Now the aim is to append a one full-sampling-period delay to the lifted integrator. Remember that the state-space model of a discrete-time system with a one-sampling-period delay is

$$G_{\text{delayed}} = \left[\begin{array}{cc|c} A & 0 & B \\ C & 0 & D \\ \hline 0 & I & 0 \end{array}\right]. \tag{6.19}$$

The delayed lifted forward integrator is then

$$G_{\text{delayed lifted forward integrator}} = \left[\begin{array}{cc|cccc} 1 & 0 & T_1 & T_1 & \dots & T_1 \\ 1 & 0 & 0 & 0 & \dots & 0 \\ \hline 0 & 1 & 0 & 0 & \dots & 0 \end{array}\right], \tag{6.20}$$

and similarly the delayed lifted backward integrator

$$G_{\text{delayed lifted backward integrator}} = \left[\begin{array}{cc|cccc} 1 & 0 & T_1 & T_1 & \dots & T_1 \\ 1 & 0 & T_1 & 0 & \dots & 0 \\ \hline 0 & 1 & 0 & 0 & \dots & 0 \end{array}\right]. \tag{6.21}$$

6.4.2 Design of a reduced observer for a lifted delayed integrator

Having obtained the models of the “plant”, the next task is to design an observer. Reduced observer will be designed here because one of the two states (the delayed output of the integrator) is immediately measured. The next few lines give a little refresher on the topic of reduced observer design.

Consider a general state-space model of a discrete-time LTI system with the states split into two groups as in

$$G = \left[\begin{array}{cc|c} A_{11} & A_{12} & B_1 \\ A_{21} & A_{22} & B_2 \\ \hline C_1 & C_2 & D \end{array}\right]. \tag{6.22}$$

Now, assume x_2 is the state that need not be estimated because it is directly measured. The second state equation can then be rewritten as

$$\underbrace{x_2(l+1) - A_{22}x_2(l) - B_2\underline{u}(l)}_{\text{measured output}} = A_{21}x_1(l), \tag{6.23}$$

and, similarly, in the first equation the “known input” can be identified as in

$$x_1(l+1) = A_{11}x_1(l) + \underbrace{A_{12}x_2(l) + B_1\underline{u}(l)}_{\text{known input}}. \tag{6.24}$$

The reduced observer is

$$\hat{x}_1(l+1) = A_{11}\hat{x}_1(l) + A_{12}x_2(l) + B_1\underline{u}(l) + L\Big(x_2(l+1) - A_{22}x_2(l) - B_2\underline{u}(l)\Big) \tag{6.25}$$

Applied to the lifted delayed discrete-time integrator, where the x_2 state is the delayed measurement, the reduced observer turns out to be described by

$$\hat{x}_1(l+1) = \hat{x}_1(l) + \begin{bmatrix} T_1 & T_1 & \dots & T_1 \end{bmatrix} \overbrace{\begin{bmatrix} u(nl) \\ u(nl+1) \\ \vdots \\ u(nl+n-1) \end{bmatrix}}^{\underline{u}(l)} + L(\underbrace{x_2(l+1)}_{x_1(l)} - \hat{x}_1(l)). \tag{6.26}$$

Rearranging the right hand side (by putting together the pieces related to $\hat{x}_1(l)$ one gets

$$\hat{x}_1(l+1) = (1-L)\hat{x}_1(l) + \sum_{k=0}^{n-1} T_1 u(nl+k) + L \underbrace{x_2(l+1)}_{\text{just measured}} . \tag{6.27}$$

The error dynamics of a general observer is given by the state matrix $(A-LC)$, which should have all its eigenvalues inside the unit disc in the complex plane for the estimate to be stable. In the present case this condition boils down to

$$|1-L| < 1 \;\Rightarrow\; L \in (0,2). \tag{6.28}$$

Picking the value $L = 1$ well inside the interval, the reduced observer turns into the familiar "integrate of the last period" scheme introduced at the beginning of this chapter

$$\hat{x}_1(l+1) = \sum T_1 u + x_2(l+1). \tag{6.29}$$

This finally proves that such observer is stable. Certainly the lifting approach not only proved the stability of the practically appealing and simple compensation technique but also it showed that the intuitive method can be made more flexible by weighting the estimated and the "just measured" values under the same guarantee of stability.

Extension of the above theoretical analysis of stability for the two other presented compensation techniques remains to be done.

6.5 SIMULATIONS

In order to demonstrate the usefulness of proposed delay compensation schemes, numerical simulations were carried out in Simulink. For this purpose the elevation gimbal model from

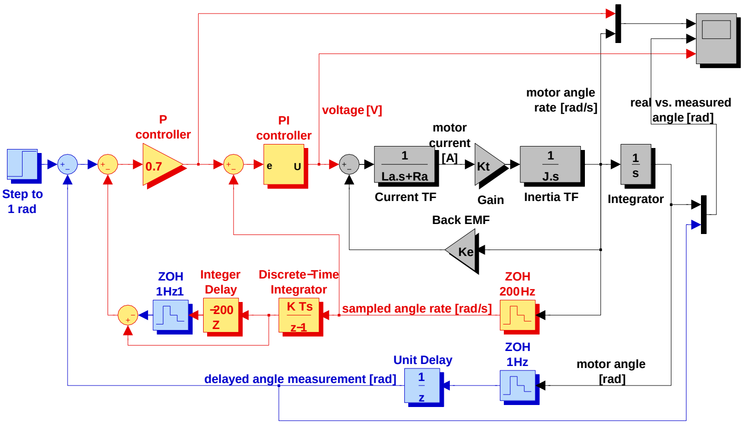


Figure 6.15: Simulink model demonstrating the proposed delay-compensated visual servoing scheme E. Signal and block colors are in compliance with Fig. 6.9, with 200 Hz inner loop frequency and 1 Hz outer loop frequency with a one-sample delay.

the double gimbal (Fig. 2.6(b)) was used. The model was introduced in section 2.2 and identified parameters are listed in table 2.1.

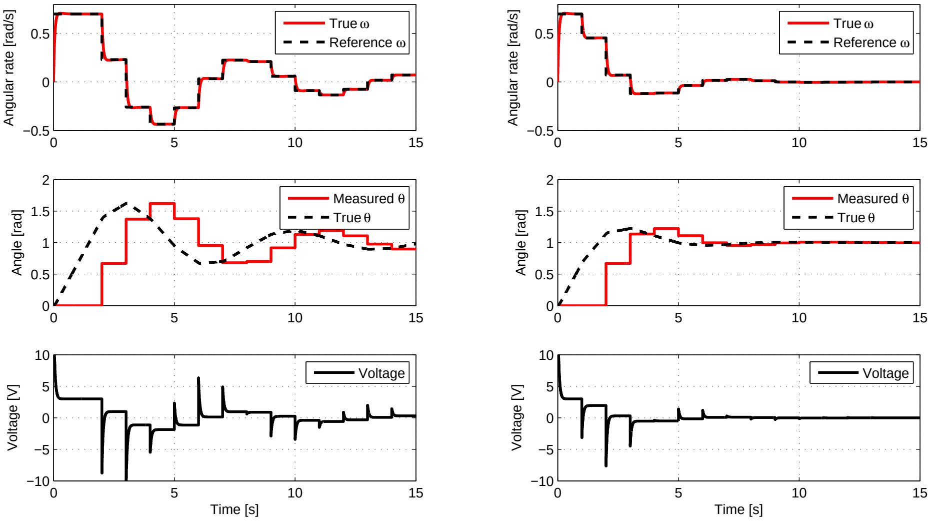
Fig. 6.15 shows a Simulink model with the cascade structure: the inner control loop for angular rate and the outer loop for the angle, which suffers from a one-sample delay in measurements. Both loops are served by classical controllers. Angular rate is regulated by a discrete PI controller (with the equivalent continuous transfer function $K(s) = \frac{25s+8}{s}$) and sampled at $T_\omega = 1/200\,\mathrm{s}$, while the angle (orientation, pointing) is controlled by a simple proportional controller $P = 0.7$ sampled at $T_\theta = 1\,\mathrm{s}$ period.

The simulation results for the classical two-loop design with no compensation of the delay is presented in Fig. 6.16(a). The task for the controller is to track a step of 1 rad in the reference angle. System responses under the same conditions with various proposed compensators plugged into the closed loops are in Fig. 6.16(b) and in Fig. 6.17. Apparently, the control design must be rather conservative if no delay compensation is included and larger overshoot is not acceptable.

6.6 EXPERIMENTAL RESULTS

To demonstrate the functionality of the proposed algorithms, laboratory experiments have been conducted. This section reports on the results achieved with the implementation of the delay compensation algorithm in the experimental double-gimbal camera platform pictured in Fig. 2.6(b). In order to obtain easy-to-interpret results which could be compared to the numerical simulations from the previous section, only single-axis (elevation) experiments were realized.

In total three experiments for different outer loop sampling rates were performed, all of them with the compensation scheme C presented in section 6.2.2. The performance



(a) Case A: Step responses of the closed-loop system with no compensation of the vision-system induced delay.

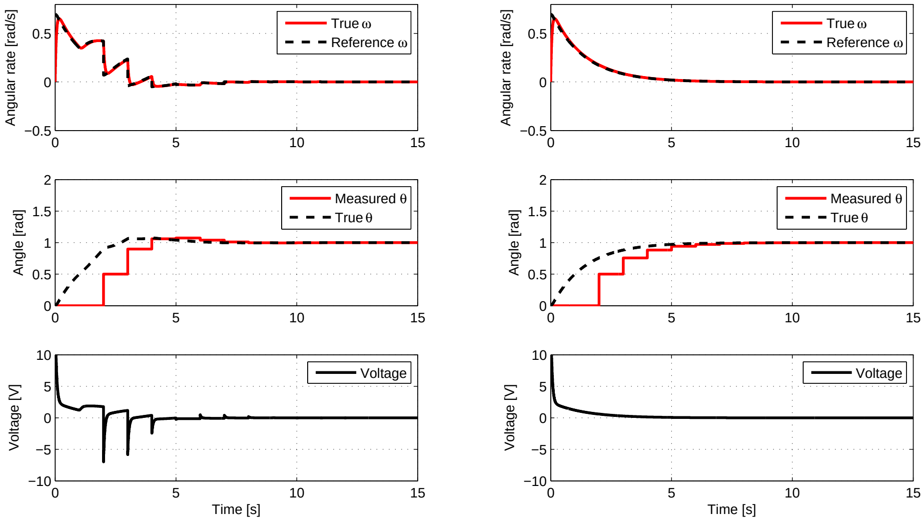
(b) Case B: Step responses of the closed-loop system with a compensation ”integrate over last period” according to Fig. 6.3.

Figure 6.16: Numerical simulations of delay compensation schemes.

of the compensated system is always compared to that of a system without any delay compensation (scheme A in Fig. 6.2). The results from the first experiment, for which the sampling rate of the outer loop was set to 1 Hz, are presented in Fig. 6.18(a). Comparing the graphs on the left and right (with the compensation switched on and off, respectively), the advantage of the compensation is apparent — when the compensation is switched on, the response is not oscillatory at all. Note that identical controllers K and P were used in both situations. Similar results were obtained for a faster sampling frequency $f_\theta = 1/T_\theta = 2\,\mathrm{Hz}$, see Fig. 6.18(b), and for $f_\theta = 5\,\mathrm{Hz}$, see Fig. 6.19. For the latter, the difference between compensated and uncompensated system is almost negligible, which is expected since the corresponding delay is smaller.

In the laboratory experiments the camera platform was seated on the ground, its base not moving, therefore it was possible to measure the true (undelayed) angle using a dedicated sensor — an incremental encoder, which need not be the case in a real inertially stabilized visual tracker. These data are in the second rows in the figures.

The third rows in the figures show the uncompensated regulation error $e_\theta(t_k) = \theta_\mathrm{r}(t_k) - \theta_\mathrm{measured}(t_k) = \theta_\mathrm{r}(t_k) - \theta(t_{k-1})$ and the regulation error $\hat{e}_\theta(t) = e_\theta(t_k) - \Delta_{\theta(t)}$ after the delay compensation based on integrating the rate. Apparently, the uncompensated error is evaluated only once every (slow) sampling period, whereas the compensation of this error is conducted at the fast sampling rate of the inner loop.



(a) Case D: Step responses of the closed-loop system with a compensation of the vision-system induced delay by the modified Smith predictor (the one without downsampling the integrated rate).

(b) Case E = C: Step responses of the closed-loop system with a compensation formulated via reset control systems.

Figure 6.17: Numerical simulations of delay compensation schemes.

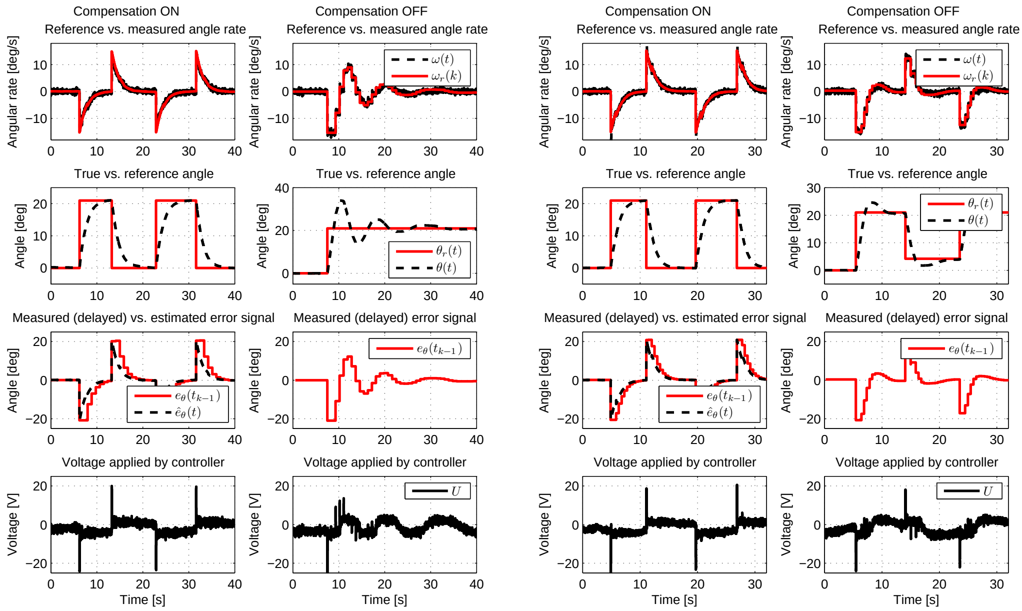
6.7 IMPLEMENTATION ISSUES ON REAL SYSTEM

6.7.1 Extension for camera gimbal

All presented compensation schemes including all the simulations and experiments so far assumed only a single-axis configuration. This section introduces important steps that are necessary to extend schemes from section 6.2 (in particular the scheme labeled as Case C) to be applicable for a general camera gimbal system. To carry out this extension, the important equations of the perspective projection and the so-called *interaction matrix* (or *image Jacobian*) are required. Both of them were introduced in sections 5.2.1 and 5.2.2. The interaction matrix shows the dependence of the velocity of the *features* in the image plane on the camera motion in the inertial space.

Integrating measured angular rates using the interaction matrix

The compensation scheme presented in the section 6.2 based on the integration of the angular rate (6.5) may be extended to the full 3D motion by replacing the integration of the single angular rate by integration of the expression involving the interaction matrix



(a) Experiment with the outer loop sampling rate **1Hz**. On the left the reference angle signal $\theta_r(t)$ makes several steps, and thanks to delay compensation it is very well followed by the true output $\theta(t)$. On the right $\theta_r(t)$ makes only a single step, and since both controllers P and K remained the same, it is no surprise that the output behaves oscillatory.

(b) Experiment with the outer loop sampling rate **2Hz**. Comparing to Fig. 6.18(a), the response is not so oscillatory even when the compensation is switched OFF, but still some overshoot is present.

Figure 6.18: Two experiments with different outer loop sampling rate settings. The delay compensation is switched ON (on the left in both figures) and OFF (on the right in both figures).

$$\begin{bmatrix} \dot{u} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} \frac{uw}{\lambda} & -\frac{\lambda^2+u^2}{\lambda} & w \\ \frac{\lambda^2+w^2}{\lambda} & -\frac{uw}{\lambda} & -u \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} + \begin{bmatrix} -\frac{\lambda}{z} & 0 & \frac{u}{z} \\ 0 & -\frac{\lambda}{z} & \frac{w}{z} \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}, \tag{6.30}$$

where instead of the single angular velocity ω now the full angular rate vector $[\omega_x, \omega_y, \omega_z]^T$ appears (the translation component will be ignored and left as a disturbance to be attenuated). The initial conditions in this equation are the image features $[u(t_{k-1}), w(t_{k-1})]^T$ which correspond to the delayed angle (orientation, pointing) measurements $\theta(t_{k-1})$ in (6.5).

Equation (6.30) thus relates the angular rate vector ω and the translation velocity vector v of the camera to the velocity of the image features $[u, w]^T$ in the image plane. The detailed derivation of the equation is given for instance in [13].

The essential problem of integrating the above differential equation resides in the necessity of knowing the initial values of $[u(t_{k-1}), w(t_{k-1})]^T$ before the numerical integration

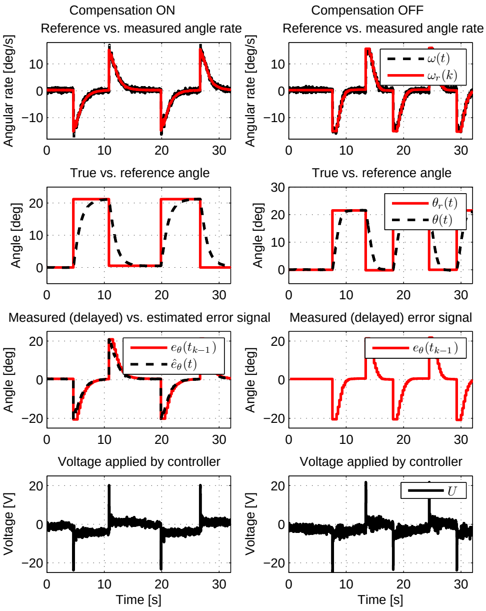


Figure 6.19: Experiment with the outer loop sampling rate **5Hz**. Compared to Fig. 6.18(a) and Fig. 6.18(b), here the advantage of using the delay compensation is negligible since the delay to be compensated is smaller.

may start. This is very restricting because according to (6.5) and Fig. 6.5 these values are known only with the one-period (T_θ) delay. To implement the integration one has to introduce three buffers for all the values from vectors ω_x, ω_y, ω_z, and the integration may only start after the coordinates $[u(t_{k-1}), w(t_{k-1})]^T$ are known. This issue becomes even more serious in case when the delay time is not known in advance, thus the buffer must be of the variable length. The situation is not investigated any further here because the problem with buffering is to be solved by a different way in next section.

Integrating camera angular rates into quaternions

If the translation speed v of the camera in its coordinate frame is zero or regarded as an unknown disturbance to be attenuated, a different approach to estimate the motion of the features in the image plane can be followed. The aim is to estimate the image feature coordinates when the camera is subject to the rotational motion. So that, it is possible to first integrate the sequence of the camera angular rate $\omega(t)$ to obtain rotation matrix representing the rotation until the features $[u(t_{k-1}), w(t_{k-1})]^T$ are known. Estimated coordinates $[\hat{u}(t), \hat{w}(t)]^T$ may be then calculated using the known camera rotation in time interval $(t_{k-1}, t\rangle$ represented by a rotation matrix.

In order to implement integration of the camera rotation to obtain rotation matrix quaternions are often used to store the information about the current orientation [78]. The following equation shows how to integrate inertial angular rate of the camera to obtain quaternion $q = [q_0, q_1, q_2, q_3]^T$ representing the rotation

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}. \tag{6.31}$$

Although quaternions are advantageous to represent rotations, to rotate vectors it is useful to convert quaternion into the rotation matrix

$$R(q) = \begin{bmatrix} (1-2q_2^2-2q_3^2) & 2(q_0q_3+q_1q_2) & 2(q_1q_3-q_0q_2) \\ 2(q_1q_2-q_0q_3) & (1-q_1^2-q_3^2) & 2(q_0q_1+q_2q_3) \\ 2(q_0q_2+q_1q_3) & 2(q_2q_3-q_0q_1) & (1-q_1^2-q_2^2) \end{bmatrix}. \tag{6.32}$$

The overall procedure for calculating the new coordinates $[u(t), w(t)]^T$ with known camera coordinates $[u(t_{k-1}), w(t_{k-1})]^T$ at time t_{k-1} and known rotation over the time $(t_{k-1}, t\rangle$ described by quaternion q may be then summarized by the following steps:

1. Calculate rotation matrix $R(q)$ from the known quaternion q using (6.32).
2. Pick up one point on the line connecting the origin with the observed point P (see Fig. 5.2) using the known $[u(t_{k-1}), w(t_{k-1})]^T$. For example the one with the z component of the point P equal to one, use (5.2).
3. Use the calculated rotation matrix $R(q)$ from the step 1 to calculate point P in the new coordinates $[\hat{x}, \hat{y}, \hat{z}]^T$.

$$[\hat{x}, \hat{y}, \hat{z}]^T = R(q)[x, y, z]^T \tag{6.33}$$

4. Calculate estimated image features $[\hat{u}(t), \hat{w}(t)]^T$ using $[\hat{x}, \hat{y}, \hat{z}]^T$ and (5.2).

Even when in this case quaternions are used to store the information about the camera rotation, still there are two options how to implement the resetting observer, equivalently to section 6.2.2. Either use one integration state q be reset according to 6.7, or to use two integration states q_A and q_B alternately reset to zero and the output of the observer be equal alternately the one or the other.

The first option, using a single state for the quaternion q requires to implement reset equation (6.7) for quaternions, which concerns mainly the extracting. In the quaternions case the extracting corresponds to multiplication by inverse quaternion

$$q(t_k^+) = q(t_k^-) \cdot q(t_{k-1})^{-1} \tag{6.34}$$

The second option is to introduce two separate quadruple of states for quaternions q_A and q_B. q_A is reset to zero rotation, which is $q_A(t_k^+) = [1, 0, 0, 0]^T$ for even k, while q_B is reset to zero rotation $q_B(t_k^+) = [1, 0, 0, 0]^T$ for odd k. The advantage of this option is that it does not require computation of multiplication by the inverse quaternion.

6.7.2 Bias present in gyro rate signal

It was already shown in section 3.6 that the bias which is usually present in gyro rate signal can significantly affect the tracking performance of the outer position loop. The performance of the schemes D and E (in Fig. 6.8 and 6.9 respectively) is significantly deteriorated, even though not completely unworkable. Since the integrator only integrates the bias over the finite interval T_θ, the estimate cannot completely diverge in presence of bias. However, such finite horizon integrator must be implemented in a way to avoid an overflow (surely not as a subtraction of two integrators as suggested in the block diagrams).

Staying assured there is no threat of instability, the influence of a bias in the rate signal can be investigated in terms of a steady-state tracking error similarly as studied in section 3.6. It was shown there that the only way to achieve zero steady-state error in position loop was to introduce integrator in the position controller.

Now, redo the analysis while considering the delay $D_\theta = e^{-sT_\theta}$ and the zero-order-hold blocks $H_\omega = \frac{1-e^{-sT_\omega}}{sT_\omega}$ and $H_\theta = \frac{1-e^{-sT_\theta}}{sT_\theta}$. The transfer function from the bias $b(t)$ to the output angle $\theta(t)$ can be calculated for the case A (the problem definition in Fig. 6.2) as

$$S_P(s) = \frac{-GKH_\omega}{s + GKH_\omega(s + PD_\theta H_\theta)}. \tag{6.35}$$

Its steady state value is

$$\lim_{s\to 0} S_P(s) = -\frac{1}{P_p}. \tag{6.36}$$

Replacing the proportional controller P by a PI controller $P(s) = P_p + P_i/s$ leads to the steady-state limit

$$\lim_{s\to 0} S_{PI}(s) = 0. \tag{6.37}$$

Presence of an integrator in the position controller does not help when any of the other compensation schemes is implemented as documented in Fig. 6.20(a). In schemes D and E the steady-state output value is never zero if a bias is present inside the inner loop. In fact, it is not appropriate to search for a limit in the scheme D, since the output angle experiences a limit cycle behavior as illustrated in Fig. 6.20(b) . Steady-state values were not derived for the reset observer formulations.

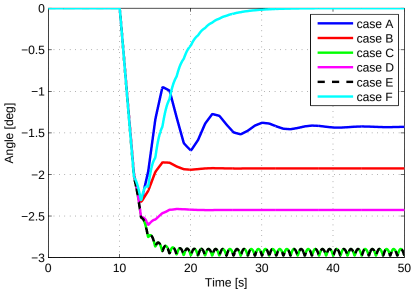
6.7.3 Bias estimation based on tracking image background

The previous section showed that the compensation schemes B, C, D and E cannot guarantee a zero steady state error in tracking a reference position (or angle, pointing, direction, in the rotational case) when the inner (rate) loop is exposed to a constant but unknown (or even slowly varying) bias.

To compensate for the unknown bias, it must be estimated first. A simple technique denoted as complementary filtering is widely used in the inertial estimation domain. A short introduction to the complementary filtering is made using [79] later in section 7.3.1. In a simplified way, complementary filter fuses the information from rate sensors (typically rate gyros) together with some position sensor(s).

Case	$P(s) = P_p$	$P(s) = P_p + \frac{P_i}{s}$
A (Fig. 6.2)	$-\frac{1}{P_p}$	0
B (Sec. 6.2.1)	N/A	N/A
C (Sec. 6.2.2)	N/A	N/A
D (Fig. 6.8)	$-T_\theta - \frac{1}{P_p}$	$-T_\theta$
E (Fig. 6.9)	$-\frac{3}{2}T_\theta - \frac{1}{2}T_\omega - \frac{1}{P_p}$	$-\frac{3}{2}T_\theta - \frac{1}{2}T_\omega$
F (Fig. 6.21)	0	0

(a) Steady state values of transfer function from bias to output — the only way to reach zero steady state error position is to employ a bias observer (see the case F in section 6.7.3) or not to use any delay compensator with simultaneous applying the PI position controller P.



(b) Step response from bias to output angle position with controller P. Bias makes a step change by 1 deg/s at time $t = 10$ s. Simulation parameters are $T_\theta = 1$ s, $T_\omega = 1/200$ s, $P = 0.7$, $K = \frac{25s+8}{s}$ and $k_b = 0.2$.

Figure 6.20: Analysis of various schemes behavior in presence of bias in gyro rate signal. Bias is assumed to enter the loop according to Fig. 3.17.

In this section it is a computer-vision system assumed to play a role of the ”position sensor”. However, the signal produced by the computer-vision tracker algorithm cannot be immediately used to estimate the gyro bias. This is due to the difference between the position estimate obtained by integrating the gyro output and the target position in the image plane as obtained by the computer-vision algorithm. The correct interpretation of the integrated gyro output is the movement of the image background θ_{bg} rather than the (image of the) target (assuming the camera is only allowed to rotate and not to translate). Application of the complementary filter to the case E (Fig. 6.9) leads to the scheme in Fig. 6.21.

It is known from [79, 80] that the filter with the structure (7.8) – (7.9) works with a steady-state estimation error in the signal $\hat{\theta}_{bg}$. In order to achieve a zero steady-state estimation error, the filter is usually extended by an integrator term. In this case it is not necessary to include another integrator term into the filter, since the filter is only used to estimate the gyro bias and not the estimate $\hat{\theta}(t)$ of the ”position”. As a proof of the functionality of the bias estimation scheme in Fig. 6.21, the steady-state values of transfer functions from bias $b(t)$ to the output $\theta(t)$ were derived both for P and PI position controllers

$$\lim_{s\to 0} S_P(s) = \lim_{s\to 0} S_{PI}(s) = 0, \tag{6.38}$$

which is also summarized by a table in Fig. 6.20(a) and visualized using a step response in Fig. 6.20(b).

6.8 CONCLUSIONS

This chapter investigated the problem often encountered in a cascade visual servoing system: the outer loop (angle, orientation, pointing, position) works at a much slower sampling

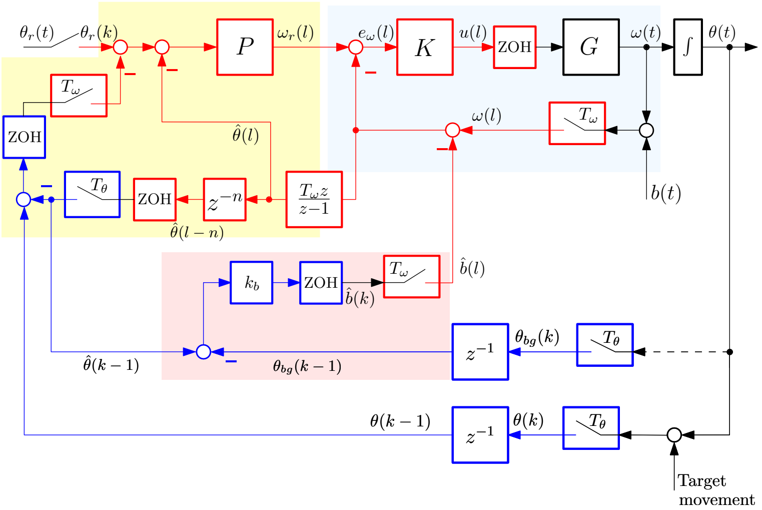


Figure 6.21: Case F: Variant of compensator form Fig. 6.9 extended for the bias estimation. Tracker algorithm must produce in this case not only the feature position but moreover the image background movement θ_{bg}.

rate than the inner loop (velocity, angular rate) loop. Moreover, it includes a one-sampling-period delay. These two properties are induced by the fact that a computer vision system is used as the sensor for the outer loop. Availability of the fast-sampled velocity measurement is exploited to compensate for the delay in the outer loop. However, the proposed compensation schemes can only be useful when the camera itself is attached to a moving carrier. The inertial (angular rate) sensors are then used to compensate for delays introduced by processing of the image information accompanied by disturbing rotation of the carrier of the camera. The compensators were designed and plugged into the already designed visual servoing system. The functionality of the proposed compensation schemes was demonstrated both using simulations and laboratory experiments. These simple solutions were formulated as special instances of some advanced control schemes, namely, reset observer methodology and modified Smith compensation. The powerful technique of lifting borrowed from the multirate systems domain was used to prove stability of some of the proposed intuitive schemes. Implementation issues were discussed for double-gimbal platforms.

An open issue is the proof of stability of the remaining observers. Possibly, some kind of *majorizing* approach could be used since stability is now proved for the weakest possible compensation.

Chapter 7

Attitude estimation using inertial measurements

This chapter describes a design and implementation aspects of a low-cost inertial estimation module based on commercially available inertial sensors. The primary task for the module described in this chapter is to estimate the attitude (orientation, pose).

Inertial measurements from sensors (accelerometers, gyroscopes, magnetometers and GPS) are fused to obtain the attitude estimate. In the following text the two filtering paradigms are compared — extended Kalman filter and output-injection-based linearization filter. No model of the dynamics of the carrier (aircraft, mobile robot, etc.) is relied upon; the only modeled dynamics is that of sensors, such as the bias and noise.

Because the inclination (direction of the gravity) is measured using accelerometer that is assumed to measure only gravity field, in presence of any translational accelerations is the estimate distorted. To prevent this issue, the algorithm presented in this chapter is extended for taking the advantage of the second derivative of GPS position which is then subtracted from accelerometer measurements. Experience with the implementation of the proposed schemes on ARM7 based microcontroller is presented so as the functionality of the device is demonstrated in experiments.

This chapter is partially based on the paper [10], but it is broaden for the feedback linearization based filter and the usage of the second derivative of GPS position.

7.1 INTRODUCTION

7.1.1 Motivation and goals

Estimating the attitude (orientation, pose) of a mobile carrier or a flying vehicle is an important task for any inertially stabilized camera system that requires to know the coordinates of the observed target or that offers the feature of pointing the camera to specified GPS coordinates. The calculation of coordinates of the observed object naturally requires GPS coordinates and the attitude of the carrier, distance to the object and angles of all of the platform gimbals. While the GPS coordinates of the carrier and distance to the object may be easily measured by a GPS receiver and a laser rangefinder, the attitude of the carrier requires an *Inertial measurement/estimation unit.*

Advances in MEMS technologies in the past decade make inertial sensors such as accelerometers and angular rate detectors affordable even for low-cost applications. When the decreasing cost of GPS receivers is taken into account, the boom in applications that can be witnessed recently is explained. This new technology brings about two challenging features into inertial estimation: the measurements are inherently very noisy and the computational resources are limited.

This chapter reports on a design, implementation and testing of one particular inertial measurement and estimation module developed at FEE CTU Prague. It presents the achievements on the path towards a full inertial unit for attitude and position estimation based on off-the-shelf inertial sensors and GPS receiver.

7.1.2 Short survey of attitude estimation literature

The literature on inertial estimation is vast. This chapter uses the established methodology of Extended Kalman Filtering described in a numerous monographs on optimal filtering and estimation, such as [81, 82]. Some practical hints relevant for navigation technologies can be found in book [83]. The principles of most low-cost sensors for UAV applications and some basic ideas of the state estimation using Kalman filtering for air vehicles are discussed in [84]. The works in [85, 80] on complementary filters can be regarded as particularly noteworthy as they lead to significant simplification and lower computational burden. This chapter, though, does not take advantage of the tricks described therein.

Another work on complementary filtering was presented in [86] and was followed by more exact solution using optimal Kalman filtering [87]. In that reference, the two concepts are proposed. The first approach is based on a nonlinear model similar to the one presented in this chapter. The second approach is based on at first completing the attitude from accelerometers and magnetometers measurements and then applying standard linear estimation methods. The task of calculating the attitude from 6 sensors (3 accelerometers and 3 magnetometers) requires solving 6 nonlinear equations with 4 unknown which calls for numerical solution of a general nonlinear least squares problem.

Yet another documented design of a low-cost inertial measurement unit can be found in [88], which is based on Sigma Points Kalman Filter.

The approach presented in this chapter took inspiration in a few papers focused on low-cost MEMS-based solutions such as [89, 90], which describe inertial estimation units based on Extended Kalman Filter paradigm. The model used for estimation is in both of them is very similar, except for dealing with the constraint on the quaternion amplitude.

7.2 SENSORS MODELING

In order to proceed further, models of sensors are needed. The model of the angular rate sensor (MEMS gyro) is

$$\boldsymbol{y} = \boldsymbol{\omega} + \boldsymbol{\eta} + \boldsymbol{b}, \tag{7.1}$$

where $\boldsymbol{y}$ is the gyro rate output (vector), $\boldsymbol{\omega}$ is the true angular rate, $\boldsymbol{\eta}$ is a gaussian sensor noise with zero mean and $\boldsymbol{b}$ is a vector of sensor bias. The sensor is assumed to be calibrated so that there is no need to consider other than unity gain. The time indices were omitted for brevity.

The accelerometer gives a measurement of the total acceleration of the sensor, which is composed of the contributions by the gravity (gravity acceleration vector $\boldsymbol{G}$ rotated via sensor attitude $\boldsymbol{R}_\varphi$), centrifugal acceleration ($\boldsymbol{\omega} \times \boldsymbol{v}$, while moving along a curved/circular path) and finally the translational acceleration $\dot{\boldsymbol{v}}$. This is given in (7.2).

$$\boldsymbol{a} = \dot{\boldsymbol{v}} + \boldsymbol{\omega} \times \boldsymbol{v} - \boldsymbol{R}_\varphi \boldsymbol{G} + \boldsymbol{\eta} \tag{7.2}$$

Apparently, the major trouble in using accelerometers is that left alone they cannot distinguish between the three contributions. The algorithm implemented in the proposed module assumes that the carrier does not accelerate (with the exception of short term variations which can be handled as disturbances), which is a significant limitation for UAV applications. A simple solution may be to discard the accelerometer measurements once they excess the 9.81 value significantly, but this can only be used occasionally because magnetometers alone cannot specify the orientation in space completely. A more systematic approach would be to perform the fusion of all the sensors such that the measurements of the accelerometers are continuously confronted with the model of the system. Such model is, however, not available in the present situation. The proposed module is planned to be used for various carriers without knowing their dynamics.

Finally, magnetometer measures the influence of the Earth magnetic field. More precisely, the projection of the Earth magnetic field vector $\boldsymbol{M} = \left[M_x\, M_y\, M_z\right]^T$ into local sensor coordinates. This projection is described by attitude rotation matrix $\boldsymbol{R}_\varphi$.

$$\boldsymbol{m} = \boldsymbol{R}_\varphi \cdot \left[M_x\, M_y\, M_z\right]^T + \boldsymbol{\eta} \tag{7.3}$$

The formats of outputs of the magnetometer and the accelerometer are similar. Both of them measure the projection of some external force field (gravity or magnetic field) into the local sensor coordinates. With the assumptions on acceleration discussed above, the measured acceleration is

$$\boldsymbol{a} = \boldsymbol{R}_\varphi \cdot \left[\begin{array}{ccc} 0 & 0 & G_z \end{array}\right]^T + \boldsymbol{\eta} \tag{7.4}$$

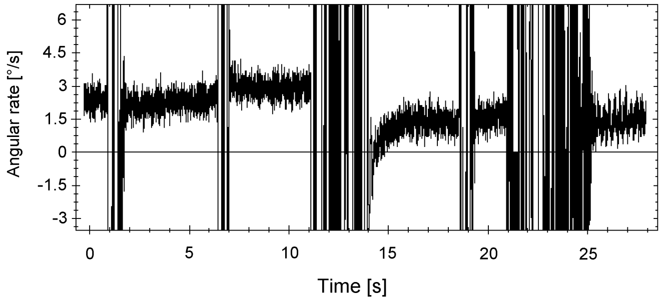


Figure 7.1: Angular rate signal measured by a MEMS inertial sensor ADIS 16400 placed on a desk. Its orientation on the desk was changed manually every five seconds. Dependence of bias on the sensor orientation is clearly visible.

7.3 ATTITUDE ESTIMATION

A naive way to estimate the attitude is to integrate the angular rate signal provided by the gyro. This is highly susceptible to bias in the sensor as illustrated in Fig. 7.1. Not only does a small bias integrate into a significant error but it also changes significantly with an attitude. To overcome this issue a fusion of several other sensors is considered. A simple technique denoted as complementary filtering is widely used in the inertial estimation domain [79].

7.3.1 The single-axis complementary filter

A complementary filter fuses the information from the rate and position sensors while simultaneously estimating the bias of the rate sensor. Both sensors are working in different — complementary — frequency regions. A short introduction to the complementary filtering is made here using [79] just for readers' convenience.

Abusing the notation, let $\theta(s)$ and $\omega(s)$ be Laplace transforms of $\theta(t)$ and $\omega(t)$, respectively. Then for every $k_1 > 0$, $\theta(s)$ admits the stable decomposition

$$\theta(s) = \frac{s+k_1}{s+k_1}\theta(s) = \frac{k_1}{s+k_1}\theta(s) + \frac{s}{s+k_1}\theta(s). \tag{7.5}$$

Using the relationship $\omega(s) = s\theta(s)$, it follows from the above equation that

$$\theta(s) = \frac{k_1}{s+k_1}\theta(s) + \frac{1}{s+k_1}\omega(s), \tag{7.6}$$

which suggests a filter with the structure (subscript m denotes measurements)

$$\hat{\theta}(s) = \frac{k_1}{s+k_1}\theta_m(s) + \frac{1}{s+k_1}\omega_m(s). \tag{7.7}$$

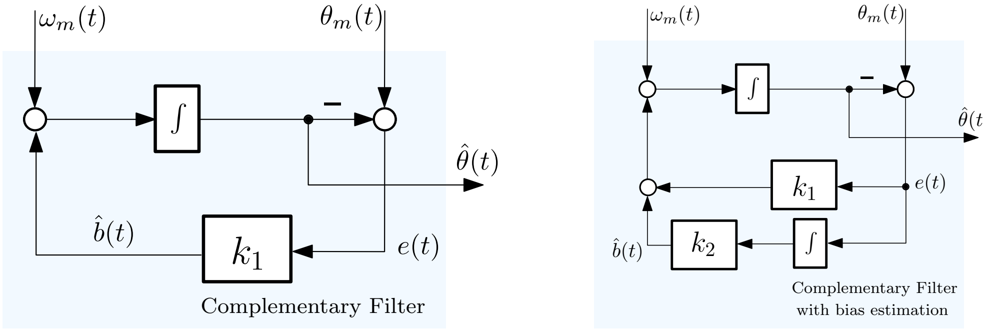


Figure 7.2: Structure of the complementary filter (see, e.g., [79]).

Clearly this filter admits the state space realization

$$\dot{\hat{\theta}}(s) = -k_1\hat{\theta} + k_1\theta_m + \omega_m \tag{7.8}$$

$$= \omega_m + k_1(\theta_m + \hat{\theta}), \tag{7.9}$$

which is represented in Fig. 7.2. Notice the following important properties.

1. $T_1(s)$ is low pass. The filter relies on the information provided by the position sensor at low frequency only.

2. $T_2(s) = 1 - T_1(s)$. The filter blends the information provided by the position sensor in the low frequency region with that available from the rate gyro in the complementary region.

3. The break frequency is simply determined by the choice of the parameter k_1.

The frequency decomposition induced by the complementary filter structure holds the key to its practical success, since it mimics the natural frequency decomposition induced by the physical nature of the sensors themselves. For instance, rate gyros are usually fairly precise at higher frequencies, but they are rather poor at low frequencies. This is especially true for the zero frequency, at which the gyro bias is present. On the other hand, the directional (or pointing) sensors such as magnetometers are precise at low frequencies.

Complementary filter design is then reduced to the computation of the gain k_1. It is crucial to point out here that computation of gain k_1 may be done using whatever filtering technique is of will, such as pole placement or Kalman filtering.

In practice, the simple complementary structure described above can be modified to meet additional constraints. For example, to achieve steady state rejection of the rate gyro bias, the filter must be augmented with an integrator to obtain the new complementary filter depicted in Fig. 7.2 on the right.

7.3.2 From 1D to 3D

In previous section the complementary filter for the single axis was introduced. To proceed further to motion in 3D, a proper representation of the orientation must be chosen first. In other words, while in the single-axis case the integration of ω_m to angle θ was done using a single integrator

$$\dot{\theta} = 1 \cdot \omega_m,$$

in the 3D motion a various representations for the attitude may be chosen. Traditionally, Euler angles would be used. As an alternative, quaternions can be considered. The relationship between the quaternions and the vector of inertial angular rate is

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}, \tag{7.10}$$

where the vector $\boldsymbol{q} = [q_0, q_1, q_2, q_3]^T$ represents a quaternion vector and ω_x, ω_y, ω_z are the three components of inertial angular rate vector.

When Euler angles formalism is to be followed, the relationship between the inertial angular rate vector and derivatives of Euler angles ϕ, θ, ψ is

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \tan\theta \sin\phi & \tan\theta\cos\phi \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi / \cos\theta & \cos\phi / \cos\theta \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}. \tag{7.11}$$

Comparing (7.10), which describes the integration through quaternions, to (7.11), which describes the integration through Euler angles, the advantage of using quaternions is evident. Both equations are nonlinear although the former contains only multiplication of quaternion components with inputs, while the latter contains trigonometric functions of angles and even fractions. This leads to singularities for certain angle values. The complications with gimbal lock due to these singularities are a nightmare in inertial estimation and control. For this reason, quaternion representation was used.

7.3.3 The resulting model to be used for estimation

The resulting model, which will be used for estimation in 3D may be explained using Fig. 7.3. The quaternion estimate (denoted $\hat{\boldsymbol{q}}$) is obtained by integrating the gyro outputs using (7.10). The predicted sensors outputs are the accelerometer estimate $\hat{\boldsymbol{a}}$ assumed to measure only the gravity and the magnetometer estimate $\hat{\boldsymbol{m}}$ assumed to measure the Earth's magnetic field directed towards the north pole. In fact, both of predicted sensors are obtained using the projection of the Earth's vector fields into the current inertial module's attitude specified by quaternion $\hat{\boldsymbol{q}}$. The accelerometer measurements are denoted by $\boldsymbol{a}$, the magnetometer measurements by $\boldsymbol{m}$.

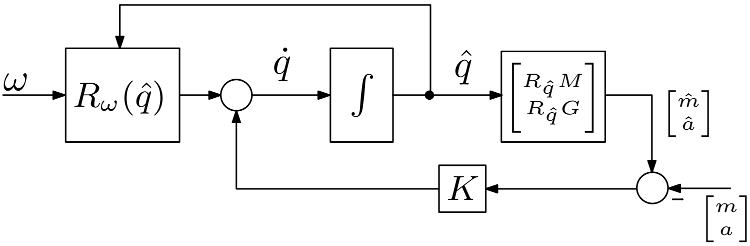


Figure 7.3: The filter model without bias estimation, where angular rates are taken as an input and accelerometers with magnetometers signals as the output.

The error between output estimates ($\hat{\boldsymbol{a}}$, $\hat{\boldsymbol{m}}$) and measured sensors ($\boldsymbol{a}, \boldsymbol{m}$) are then multiplied by a correction gain K and fed into the current derivative of the quaternion estimate $\dot{\boldsymbol{q}}$. In fact, this is the standard scheme for an observer.

Equations of the complete model without the feedback K are listed in (7.12) with the definition for projection $R(\boldsymbol{q})$ in (7.13). The model has 4 state variables — components of the quaternion vector $\boldsymbol{q} = [q_0, q_1, q_2, q_3]^T$ and 6 measured outputs — a_x, a_y, a_z for measured accelerations in body frame and m_x, m_y, m_z for measured magnetic field also in body frame. The rotational matrix $R(\boldsymbol{q})$ defines the projection of the vector in the reference frame into the body frame. The two vectors defined in the reference frame are: the gravity vector $\boldsymbol{G} = [0\,0\,G_z]^T$, where $G_z \doteq 9.81\,\mathrm{ms}^{-2}$, which is supposed to be a constant in a given place on the Earth, and the Earth's magnetic field vector $\boldsymbol{M} = [M_x, M_y, M_z]^T$, which is also assumed to be a constant in a given time and place on the Earth.

State equations: **Output equations:**

$$\begin{array}{rcl} \dot{q}_0 &=& \frac{1}{2}(-\omega_x \cdot q_1 - \omega_y \cdot q_2 - \omega_z \cdot q_3) \\ \dot{q}_1 &=& \frac{1}{2}(+\omega_x \cdot q_0 + \omega_z \cdot q_2 - \omega_y \cdot q_3) \\ \dot{q}_2 &=& \frac{1}{2}(+\omega_y \cdot q_0 - \omega_z \cdot q_1 + \omega_x \cdot q_3) \\ \dot{q}_3 &=& \frac{1}{2}(+\omega_z \cdot q_0 + \omega_y \cdot q_1 - \omega_x \cdot q_2) \end{array} \qquad \begin{array}{l} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = R(\boldsymbol{q}) \cdot \begin{bmatrix} 0 & 0 & G_z \end{bmatrix}^T \\ \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} = R(\boldsymbol{q}) \cdot \begin{bmatrix} M_x \, M_y \, M_z \end{bmatrix}^T \end{array} \tag{7.12}$$

$$R(\boldsymbol{q}) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2 \cdot (q_0 \cdot q_3 + q_1 \cdot q_2) & 2 \cdot (q_1 \cdot q_3 - q_0 \cdot q_2) \\ 2 \cdot (q_1 \cdot q_2 - q_0 \cdot q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2 \cdot (q_0 \cdot q_1 + q_2 \cdot q_3) \\ 2 \cdot (q_0 \cdot q_2 + q_1 \cdot q_3) & 2 \cdot (q_2 \cdot q_3 - q_0 \cdot q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{7.13}$$

Now, having the model, the only missing part to implement the observer is the feedback gain K. Since the model is nonlinear, the Extended Kalman filter is usually the first choice.

7.3.4 Filtering using Extended Kalman Filter

A short summary for the popular estimation methodology for nonlinear systems known as Extended Kalman Filter (EKF) is given in this section, just for the completeness and better orientation of the reader in all computational steps. For more details, see the arbitrary book on optimal estimation [81], [82].

Extended Kalman Filter summary

Assume a discrete dynamical system modeled as

$$\begin{aligned}
\boldsymbol{x_t} &= \boldsymbol{f}(\boldsymbol{x_{t-1}}, \boldsymbol{u_{t-1}}) + \boldsymbol{w_{t-1}} \\
\boldsymbol{y_t} &= \boldsymbol{h}(\boldsymbol{x_t}, \boldsymbol{u_t}) + \boldsymbol{v_t} \\
\boldsymbol{w_t} &\sim (0, \boldsymbol{Q}) \\
\boldsymbol{v_t} &\sim (0, \boldsymbol{R}),
\end{aligned}$$

where $\boldsymbol{x_t}$ is the state vector, $\boldsymbol{u_t}$ is the (known) input to the system, $\boldsymbol{y_t}$ is the measured output, $\boldsymbol{w_t}$ is a stochastic and unmeasured disturbance acting on the system and $\boldsymbol{v_t}$ is a stochastic noise that corrupts the measurements of the output. The two stochastic processes are assumed to be best approximated by a white noise with zero mean value and covariance matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$, respectively.

The essential principle of the Extended Kalman filter is to *linearize* the nonlinear system dynamics at every time step around the best estimate of the current state and apply the matrices defining the linear model ($\boldsymbol{A_{t-1}}$, $\boldsymbol{C_t}$) for both the *time update* and *measurement update steps*. These two steps, which are repetitively performed (the whole procedure being initialized by the state estimate and the estimate of the covariance matrix) can be described as

- Compute the partial derivative of the state equation and use it to perform the time update of the state estimate and estimation-error covariance as follows:

$$\begin{aligned}
\boldsymbol{A_{t-1}} &= \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{\hat{x}^{+}_{t-1}},\ \boldsymbol{u}=\boldsymbol{u_{t-1}}} && (7.14) \\
\boldsymbol{P^{-}_{t}} &= \boldsymbol{A_{t-1}} \boldsymbol{P^{+}_{t-1}} \boldsymbol{A^{T}_{t-1}} + \boldsymbol{Q}, && (7.15) \\
\boldsymbol{\hat{x}^{-}_{t}} &= \boldsymbol{f}(\boldsymbol{\hat{x}^{+}_{t-1}}, \boldsymbol{u_{t-1}}) && (7.16)
\end{aligned}$$

- Compute the partial derivative of the output equation and use it to perform the measurement update of the state estimate and estimation-error covariance as follows:

$$\begin{aligned}
\boldsymbol{C_t} &= \left.\frac{\partial \boldsymbol{h}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{\boldsymbol{t}}^{-},\, \boldsymbol{u}=\boldsymbol{u}_{\boldsymbol{t}}} && (7.17)\\
\boldsymbol{K_t} &= \boldsymbol{P_t^{-}} \boldsymbol{C_t^{T}} \Big(\boldsymbol{C_t} \boldsymbol{P_t^{-}} \boldsymbol{C_t^{T}} + \boldsymbol{R}\Big)^{-1} && (7.18)\\
\hat{\boldsymbol{x}}_{\boldsymbol{t}}^{+} &= \hat{\boldsymbol{x}}_{\boldsymbol{t}}^{-} + \boldsymbol{K_t}\Big(\boldsymbol{y_t} - \boldsymbol{h}(\hat{\boldsymbol{x}}_{\boldsymbol{t}}^{-}, \boldsymbol{u_t})\Big) && (7.19)\\
\boldsymbol{P_t^{+}} &= \boldsymbol{P_t^{-}} - \boldsymbol{P_t^{-}} \boldsymbol{C_t^{T}} \boldsymbol{K_t^{T}}, && (7.20)
\end{aligned}$$

where $\boldsymbol{P}$ is the state covariance matrix, $\boldsymbol{K}$ is known as Kalman gain, $\boldsymbol{A}$ is a state Jacobian, $\boldsymbol{C}$ is an output Jacobian and $\hat{\boldsymbol{x}}_{\boldsymbol{t}}$ denotes a state estimate. Superscripts in $\boldsymbol{P}^{-}$ and $\boldsymbol{P}^{+}$ stand for a posteriori and a priori estimates.

Using EKF for filtering

Algorithm of the attitude estimation with the model (7.12) and employing extended Kalman filter was implemented in the hardware (see details of the developed hardware in section 7.4.1). All the results and details are also reported in the conference paper [10] by the author.

To make the description of development of the attitude observer complete, a few words on software issues. The code was written in C and compiled with the popular and free GNU GCC compiler suite (which supports ARM based processors). On the coding side, the key operations were matrix multiplications and matrix inversion such as the inversion of the matrix $\boldsymbol{A'}$ given as

$$(\boldsymbol{A'})^{-1} = \Big(\boldsymbol{C}\boldsymbol{P}\boldsymbol{C}^{T} + \boldsymbol{R}\Big)^{-1},$$

which is the part of (7.18). To speed this routine up one can notice that there is no need to compute the inversion $(\boldsymbol{A'})^{-1}$ but it suffices to solve the equation $\boldsymbol{A'} \cdot \boldsymbol{K} = \boldsymbol{P}\boldsymbol{C}^{T}$. And even more, the fact that matrix $\boldsymbol{A'}$ is symmetric and positive definite leads to use Cholesky decomposition [91].

Notice that equations (7.19) and (7.16) provide corrections of the current states — quaternions. These corrections though do not respect the restriction on unit absolute value of the quaternion. To avoid numerical problems, quaternion normalization must be always implemented (dividing all four quaternion components by absolute value of the quaternion).

Finally, the computational speed achieved with the extended Kalman filter (7.12) was only 20 Hz. The low computational speed thus initiated a research for a more efficient algorithm of filtering.

7.3.5 Filtering using feedback linearization method

Starting with the idea of replacement the computationally most complicated part of EKF — the Kalman gain computation — by a different method, one may think of using Jacobian inverse in the place of Kalman gain matrix. Intuitive interpretation of such choice is simple

— multiply the error signal e (see the Fig. 7.4) by some gain, which is for each component of error vector proportional (with constant k_1) to its impact on current state (quaternion). This impact is actually inverse of output Jacobian ($\frac{\partial h(q)}{\partial q}$) at current operating point. An illustration of this approach is seen in Fig. 7.4. A hint of what this filter does may also be seen from the filter error dynamics of the linearized system, which is

$$\dot{\boldsymbol{e}} = (\boldsymbol{A} - \boldsymbol{K}\boldsymbol{C})\boldsymbol{e} = (\boldsymbol{A} - k_1 \cdot (\boldsymbol{C}^T\boldsymbol{C})^{-1}\boldsymbol{C}^T \cdot \boldsymbol{C})\boldsymbol{e} = (\boldsymbol{A} - k_1\boldsymbol{I})\boldsymbol{e}. \tag{7.21}$$

Of course this equation holds in general only for linear systems, so that also so vaguely it must be understood only as an illustration of the principle (mathematical proof of convergence would be complex, but that is indeed the same issue with extended Kalman filter). The proportional constant k_1 adjusts the time constant of the estimation error reaching zero. Comparing to EKF, here instead of solving covariance matrix inversion (7.18), the pseudoinverse of general matrix C needs to be computed, which leads to use QR factorization [91]. The most remarkable advantage of the algorithm is that comparing to the former EKF algorithm, there is no need here to solve (7.14), (7.15), (7.19) and (7.20). Finally, the computational speed achieved with this algorithm was 200 Hz, which brings another advantage — a possibility to process almost all data coming from the sensor and to perform smoother gyro rates integration.

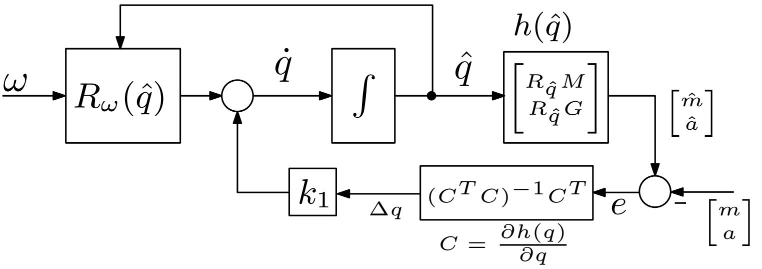

ω
$R_\omega(\hat{q})$
$\dot{q}$
$\int$
$\hat{q}$
$h(\hat{q})$
$\begin{bmatrix} R_{\hat{q}}M \\ R_{\hat{q}}G \end{bmatrix}$
$\begin{bmatrix} \hat{m} \\ \hat{a} \end{bmatrix}$
k_1
Δq
$(C^TC)^{-1}C^T$
$C = \frac{\partial h(q)}{\partial q}$
e
-
$\begin{bmatrix} m \\ a \end{bmatrix}$

Figure 7.4: Block diagram of feedback linearization based scheme, redrawn from Fig. 7.3. The Kalman gain is replaced by the term proportional to system output Jacobian inverse.

After this algorithm was devised and implemented, the same principle was found already published in the conference paper [86].

7.3.6 Extension for bias estimation

As it was demonstrated in Fig. 7.1, the bias present in gyro rate signal is significantly varying and may decrease the estimation precision. Moreover, the bias leads to nonzero steady state estimation error. This may be proved by calculation of the transfer function from the bias (present in ω_m) to the estimation error $e(t)$ in Fig. 7.2. This transfer function is obviously

$$P(s) = -\frac{1}{s + k_1}.$$

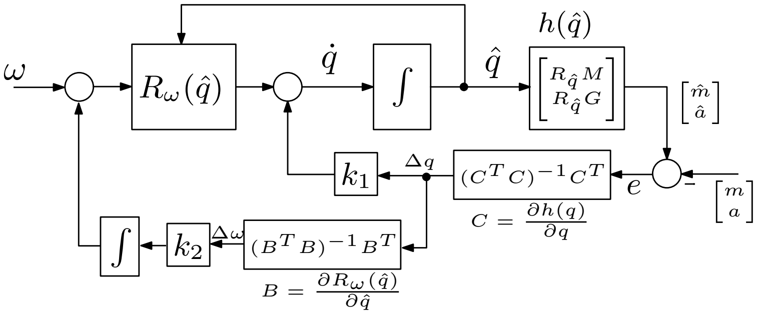

$h(\hat{q})$
ω
$R_\omega(\hat{q})$
$\dot{q}$
$\int$
$\hat{q}$
$\begin{bmatrix} R_{\hat{q}}M \\ R_{\hat{q}}G \end{bmatrix}$
$\begin{bmatrix} \hat{m} \\ \hat{a} \end{bmatrix}$
k_1
Δq
$(C^TC)^{-1}C^T$
e
$\begin{bmatrix} m \\ a \end{bmatrix}$
$C = \frac{\partial h(q)}{\partial q}$
$\int$
k_2
$\Delta\omega$
$(B^TB)^{-1}B^T$
$B = \frac{\partial R_\omega(\hat{q})}{\partial \hat{q}}$

Figure 7.5: The filter model extended for bias estimation.

The steady state which is

$$\lim_{s\to 0} P(s) = -\frac{1}{k_1},$$

is thus nonzero.

In the section 7.3.1 the extension for a bias estimator was presented in the single-axis case. The idea was to add an integrator into the feedback. This approach may be used even in the case of full 3D movement, but coordinate frames must be taken into account properly. The resulting scheme used for bias estimation is depicted in Fig. 7.5. The scheme extends the feedback linearization based approach. The impact of bias (present in ω) to quaternions $\dot{q}$ is

$$B = \frac{\partial R_\omega(\hat{q})}{\partial \omega}.$$

The feedback is closed using the inverse of B times some stabilizing constant k_2. In fact, pseudo-inverse may be calculated here symbolically so that there is no need to calculate QR decomposition.

$$B^{\#} = (B^TB)^{-1}B^T = 2 \cdot \begin{bmatrix} -q_1 & q_0 & q_3 & -q_2 \\ -q_2 & -q_3 & q_0 & q_1 \\ -q_3 & q_2 & q_1 & q_0 \end{bmatrix}$$

7.4 IMPLEMENTATION AND EXPERIMENTS

7.4.1 Designed hardware

A block diagram of designed hardware can be found in the Fig. 7.6. Main components are ADIS 16405 inertial measurement sensor (IMU) from Analog devices. The hardware is also equipped with GPS receiver Venus 634 from Sparkfun. The main computational core of the hardware is the popular LPC2368 microcontroller belonging to the ARM7 family, which can have the maximum 72 MHz clock speed and 52 kB RAM memory. The processor contains all the common peripherals like A/D converter, SPI, CAN and RS232 interfaces.

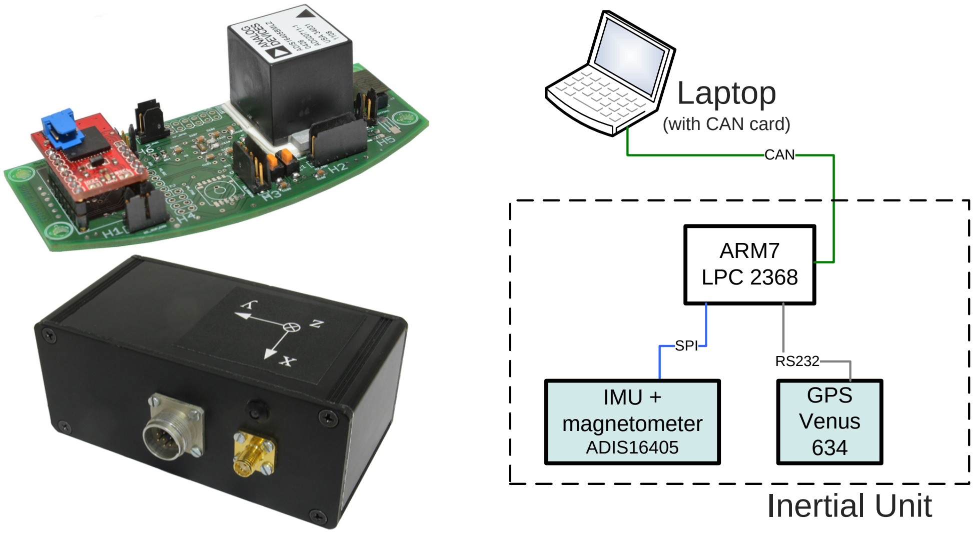


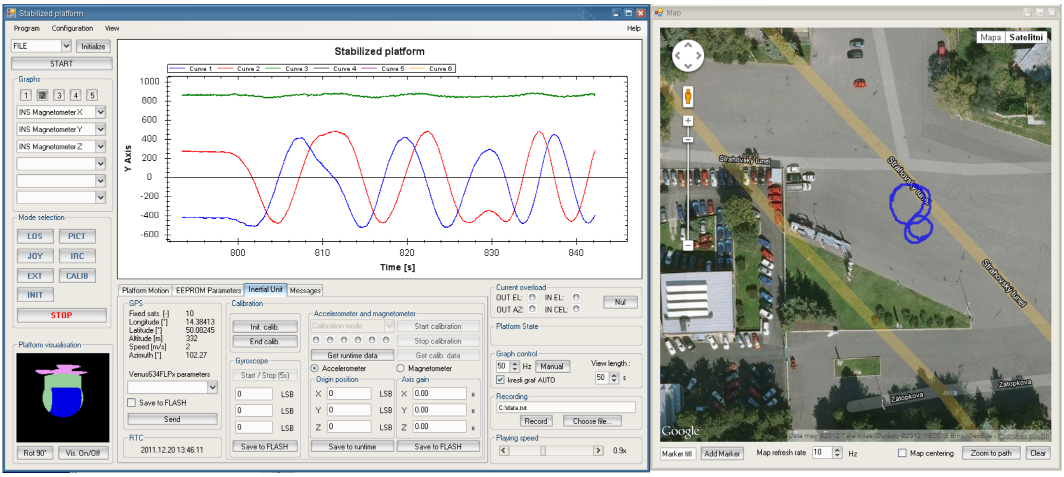
Figure 7.6: Designed module hardware and its block diagram.

IMU ADIS 16400 contains a 14-bit 3-axis gyroscope with a range of ±300 deg/s, 14-bits 3-axis accelerometer with a range of ±10 g and a 14-bits 3-axis magnetometer. Analog bandwidth of the accelerometers and gyros is 350 Hz and the maximum sampling frequency is 1.2 kHz. Analog bandwidth of magnetometer is even higher — as high as 1.54 kHz.

An essential part of the development and use of the module is a graphical user interface (GUI). It was designed for both realtime display of the measured inertial data and the estimated attitudes, but can also be used for off-line scrutinizing the data logged into an SD card. A screenshot of the GUI can be seen in the Fig. 7.7. GUI shows the measured data from 3-axis magnetometer together with GPS position drawn into a map and 3D model showing actual attitude all during several turns with the module mounted onto a car.

7.4.2 Experimental results - indoor tests

This section reports on some experimental results achieved with the proposed estimation module with a real-time ”feedback linearization” based filter. In the right bottom corner of the Fig. 7.8 are results of the bias estimation during the first tens of seconds after the algorithm started. At that moment, the IMU was left on the table at rest, thus any non-zero rate $\omega_{x,y,z}$ constitute a gyro bias. In about 20 s all three biases are automatically estimated and suppressed from the further integration. The rest of the graphs in the Fig. 7.8 show another experiment. At first, the module was held by a person at rest for 10 s and then rotated into another direction and back in sequence around all three body axes to about 70 °. The graph shows the data from all the three sensors, quaternion estimates and also the Euler angles computed offline for better visualization.



GPS
Fixed sats. [-] 10
Longitude [°] 14.38413
Latitude [°] 50.08245
Altitude [m] 332
Speed [m/s] 2
Azimuth [°] 102.27

Venus634FLPx parameters
☐ Save to FLASH
Send

RTC
2011.12.20 13:46:11

Gyroscope: 0 LSB, 0 LSB, 0 LSB

● Accelerometer ○ Magnetometer
Origin position: X 0 LSB, Y 0 LSB, Z 0 LSB
Axis gain: X 0.00 x, Y 0.00 x, Z 0.00 x

Current overload: OUT EL: ○ IN EL: ○ OUT AZ: ○ IN CEL: ○

Graph control: 50 Hz, ☑ kresli graf AUTO, View length : 50 s

Recording: C:\data.txt

Playing speed: 0.9x

Map refresh rate 10 Hz
☐ Map centering

Figure 7.7: Graphical user interface for realtime and offline visualization of the measured data and estimated attitudes.

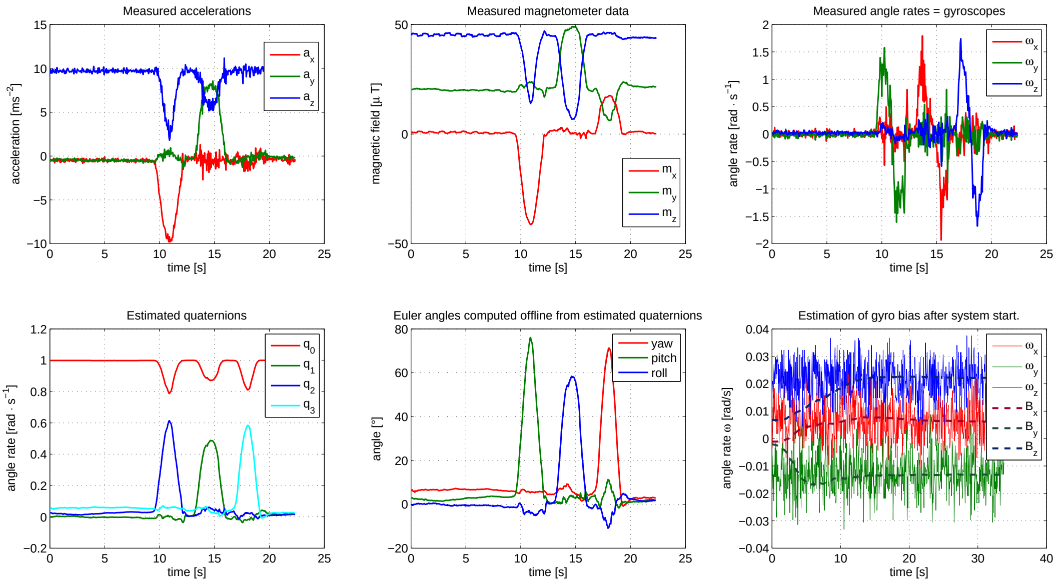


Figure 7.8: Bottom right corner: Measured gyro angle rates and appropriate bias estimation during first 30 s after startup, module is all time at rest. All other figures: Data of movement with module in laboratory conditions. The module was held by a person at rest for 10 s and then it was rotated one way and then back in sequence around all three body axes to about 70 °.

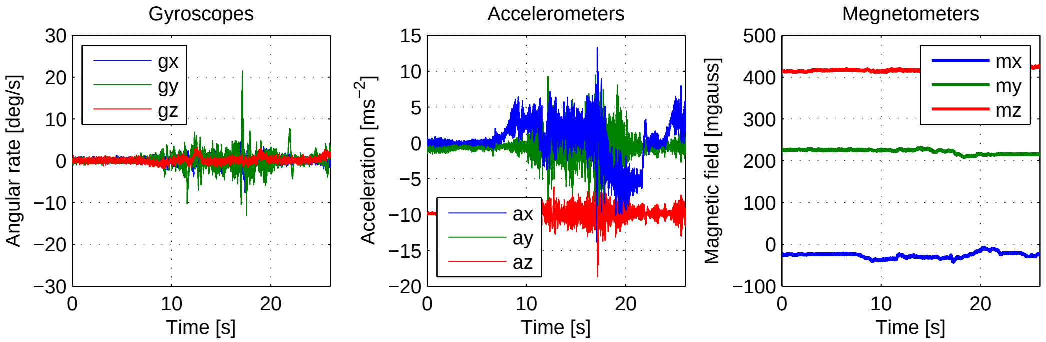


Figure 7.9: Inertial data recorded during driving the straight horizontal road. Notice that gyros and magnetometers show no rotational movement at all. The only sensor affected by translational acceleration is the accelerometer copying the car acceleration and braking in its x axis.

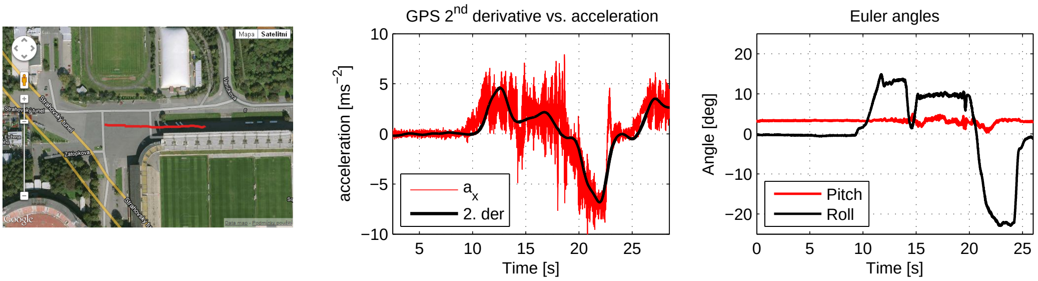


Figure 7.10: Left: GPS data recorded during driving the straight horizontal road. The bold red straight line is the GPS position. Middle: Recorded acceleration in x axis plotted together with the second derivative of the GPS signal in the corresponding coordinate frame. Right: Estimated Euler angles during the experiment.

7.4.3 Troubles with disturbing translational accelerations

Designed and implemented algorithm of the inertial attitude estimation (section 7.3.5) with the bias estimation (section 7.3.6) was subsequently tested in an outdoor experiment. The module was mounted on top of a car and several manoeuvres were carried out. Because no reference attitude is known, the evaluation of the precision may be done only using the rough knowledge of the experiment trajectory. That is, the car was performing acceleration using the first gear, changing to the second gear and finally braking, all during the straight driving on a horizontal road. It is evident that Euler angles should not change (may change only a little due to the car's suspension). Recorded inertial measurements from the experiment are visualized in Fig. 7.9. The proof that the car swinging during acceleration is negligible (ie. rotation around any axis is negligible) may be seen in the graph of gyroscopes. Notice also that the acceleration measurement a_x clearly contains acceleration and deceleration behavior, which causes that the measurement of the gravity vector is deflected.

Recorded GPS position that is showing the straight driving is shown in Fig. 7.10 on the left. In the same figure on the right corresponding estimated Euler angles are shown. It is no surprise that the estimation of the pitch angle is markedly affected by acceleration a_x. The deviation of about 20° is not admissible in this kind of application. Of course, reducing the feedback gain k_1 (see section 7.3.5) would lead to certain improvement in attenuation of the disturbing translational acceleration. This is though not the case of centripetal accelerations. See for example the manoeuvre recorded in Fig. 7.7 where the car with the module mounted on the roof is performing several rotations on a flat parking place. The disturbing centripetal acceleration is in this case permanent.

7.5 SECOND GPS DERIVATIVE

Searching for the way how to compensate the permanent disturbing acceleration, one can come up with the idea of taking the advantage of the signal from GPS module — its second derivative. The comparison graph of the translational acceleration a_x and the second derivative of the GPS for the experiment with the straight driving is plotted in Fig. 7.10 in the middle. Notice that slowly changing translational acceleration can be very well predicted and thus even compensated by introducing the second derivative of GPS position. This section explores the approach of filtering out the unwanted translational acceleration from the accelerometer using GPS measurements.

7.5.1 Implementation of the second derivative

The numerical implementation of the derivative always comes followed by low-pass filtering. The cut-off frequency of the low-pass filter is in this case as low as about $1 - 2\,\mathrm{Hz}$. Otherwise the output derivative signal was noisy and inapplicable. The cut-off frequency thus sets the frequency region where the GPS compensation attenuates the disturbing movement. Clearly, higher frequencies in translational acceleration movement cannot be compensated. As such they are though compensated very well by complementary filter.

The second derivative of the GPS signal, shown in Fig. 7.10 in the middle, was generated by processing all data from the experiment using Matlab's `filtfilt` command. Employing the noncausal filtration is essential, because otherwise a phase delay is always introduced and the compensation does not work well. The trouble though lies in the implementation of the zero-phase filtering in real time.

A simple solution may take advantage of delaying all input data (accelerometer, magnetometer and gyroscopes) for some time and store them in a FIFO buffer. Such a way the attitude estimation filter will operate on delayed data and thus will produce a delayed attitude estimate. The advantage is that now the zero-phase derivative calculation and filtering may be implemented. Let's live with this idea for a while. How to obtain undelayed estimates in current time will be shown in the following subsection.

Since the delay is assumed only for a limited time, at each time instant only a few samples from the "future" are known. Thus, the filter can count with the arbitrary

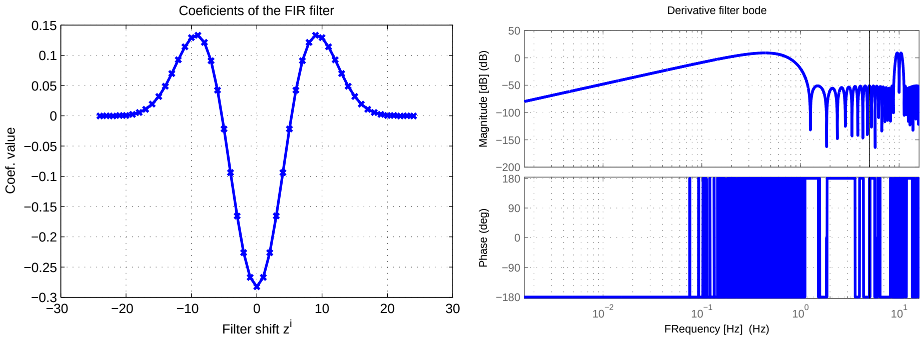


Figure 7.11: Left: Impulse response of the noncausal second derivative filter $F(z)$. To calculate the output, filter needs in total 50 input values — 25 from the past and 25 from the future.
Right: Bode plot of the filter $F(z)$. Filter has no phase delay. Phase plot is equal to exactly either $-180°$ or $+180°$, which is the plotting bug of Matlab.

number of values in the past (both FIR and IIR filters types are feasible for the past), but it must count only with a limited number of samples in future, in other words only FIR filter is feasible. Among other methods suitable for design of the zero-phase filter, the most straightforward is the filter of the form

$$F(z) = A(z) \cdot A(z^{-1}), \tag{7.22}$$

where $A(z)$ is arbitrary discrete transfer function in Z-transform. Filter $F(z)$ has the zero phase delay. The requirement on only a limited number of samples from future can be fulfilled by $A(z)$ of the FIR type (Finite Impulse Response transfer function). The maximum allowed order of the transfer function $A(z)$ is limited by the length of input data delay. The 10 Hz sampling rate of GPS module at the same time with the requirement for low-pass filtering with cut-off frequency of 1 Hz leads to the 25th order of FIR filter. The impulse response of the resulting filter $F(z)$ with the order of corresponding $A(z)$ equal to 25 is shown in Fig. 7.11. Notice that the order 25 and sampling rate 10 Hz directly determine the required algorithm delay to be 2.5 s.

7.5.2 Obtaining the attitude estimate at current time

In most of applications it not sufficient to supply the estimate of the attitude delayed for 2.5 s. To obtain the estimate at the current time one may take gyroscope signals stored in the buffer and integrate through the entire buffer. The size of the buffer is determined by the algorithm delay in seconds times the algorithm sampling rate $1/T = 200$ Hz. The buffer length is thus 500. Starting with the delayed attitude estimate $\hat{\boldsymbol{q}}(-500)$ at discrete time $k = -500$ (zero time k means now, the sampling frequency is 200 Hz), the estimate at time

$k = 0$ is obtained by recursive solving the equation

$$\begin{bmatrix} q_0(k+1) \\ q_1(k+1) \\ q_2(k+1) \\ q_3(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & -T/2 \cdot \omega_x(k) & -T/2 \cdot \omega_y(k) & -T/2 \cdot \omega_z(k) \\ T/2 \cdot \omega_x(k) & 1 & T/2 \cdot \omega_z(k) & -T/2 \cdot \omega_y(k) \\ T/2 \cdot \omega_y(k) & -T/2 \cdot \omega_z(k) & 1 & T/2 \cdot \omega_x(k) \\ T/2 \cdot \omega_z(k) & T/2 \cdot \omega_y(k) & -T/2 \cdot \omega_x(k) & 1 \end{bmatrix}}_{A(\omega(k))} \begin{bmatrix} q_0(k) \\ q_1(k) \\ q_2(k) \\ q_3(k) \end{bmatrix}. \tag{7.23}$$

This equation is in fact only the discrete version of (7.10) with Euler discretization with period T, and $\omega_x(k)$, $\omega_y(k)$ and $\omega_z(k)$ are values stored in FIFO buffer. Of course, this equation must be run 500 times always when the attitude estimate at current time is needed — usually 500×200 times in every second. The target microcontroller is not powerful enough to do this job. Instead, one may look at the problem the other way. Recursive solving of the equation (7.23) may be written using (7.24) – (7.26).

$$q(k+1) = A(\omega_k)q(k) \tag{7.24}$$

$$q(k+2) = A(\omega_{k+1})q(k+1) = A(\omega_{k+1}) \cdot A(\omega_k)q(k) \tag{7.25}$$

$$q(k+500) = \underbrace{A(\omega_{k+499}) \cdot A(\omega_{k+498}) \cdots A(\omega_k)}_{A_{500}} q(k) \tag{7.26}$$

Matrix $A(\omega_k)$ is defined in (7.23) and is completely parameterized by the gyro data in the buffer. The overall process of the integration may be thus written as multiplication of 500 matrices. The matrices compounded together are called A_{500}. By using this definition, adding a new sample $\boldsymbol{\omega_{k+500}}$ and a new actualized estimate $\hat{q}(k+1)$ means solving

$$q(k+501) = A(\omega_{k+500}) \cdot A_{500} \cdot A(\omega_k)^{-1}\hat{q}(k+1). \tag{7.27}$$

Moreover, since the structure of the matrix $A(\omega_k)$ is not changing over a time, the inverse $A(\omega_k)^{-1}$ may be calculated symbolically. Also notice that now the computational burden is comparing to recursive solving of (7.23) much improved.

After the implementation of this algorithm one issue was uncovered. Both of filtering algorithms (EKF and "feedback linearization") required that corrected quaternion is step by step normalized. Multiplication of 500 matrices in (7.26) though does not contain the normalization. From (7.26) it is clear that if both vectors $q(k+500)$ and $q(k)$ have absolute value (quadratic norm) equal to one, then also matrix A_{500} should have the identity quadratic norm. Performing a calculation of the matrix quadratic norm is though computationally very expansive. Using the Frobenius norm instead will do the same job.

7.5.3 Estimation improvement results

The GPS derivative-correction algorithm that was introduced in this section and successfully implemented in the hardware was tested in the same experiments as showed in section 7.4.3.

The first performed was the experiment with acceleration and deceleration on the straight horizontal road. The results are shown in Fig. 7.12, where in total three runs by car were done. Graphs clearly prove that introducing a GPS correction helped to reduce the error from true attitude — which should be in this experiment constant and not changing during acceleration. Graph also shows that reducing the feedback gain k_1 is another options to reduce the impact of linear accelerations on attitude estimate.

Results of the second experiment are shown in Fig. 7.13. It is the experiment with the long term centrifugal acceleration acting on the car. The amplitude of the acceleration about 5ms^{-2} is acting for more than 20s. Also this experiment proved that introducing the GPS-based compensation improved the results of the attitude estimation.

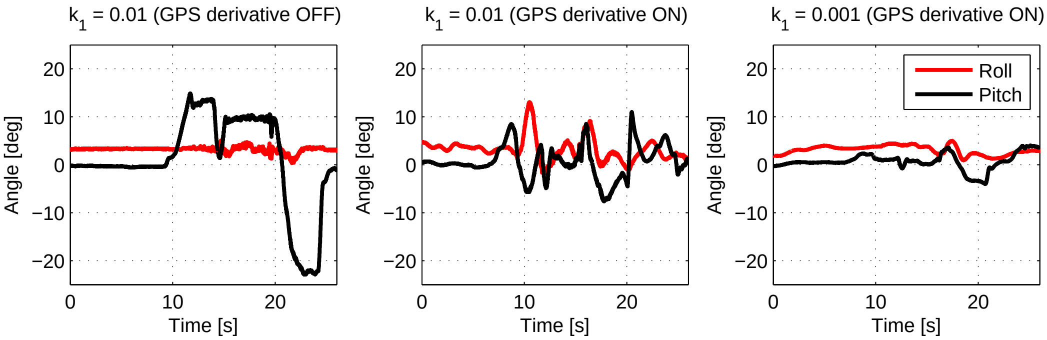


Figure 7.12: Results of three runs of the same experiment as in Fig. 7.10 — acceleration and deceleration on a straight horizontal road. Notice that turning the GPS compensation ON (middle graph) helped to reduce translational acceleration impact on estimated angles. By reducing feedback gain k_1 the estimate is even better — both angles are almost not changing at all as expected.

7.6 CONCLUSION

This chapter presented an application of the state estimation theory to the attitude estimation by employing low-cost inertial sensors. Chapter presented the two algorithms that were studied in terms of the quality and computational burden. Since both algorithms assumed that accelerometer is measuring only the Earth's gravity field, the attitude estimates were deviated when subjected to disturbing accelerations. To cope with this issue the second derivative of GPS signal was proposed as a possible way to correct the attitude estimates. Practical function of the inertial attitude estimation unit was demonstrated by several indoor and outdoor experiments.

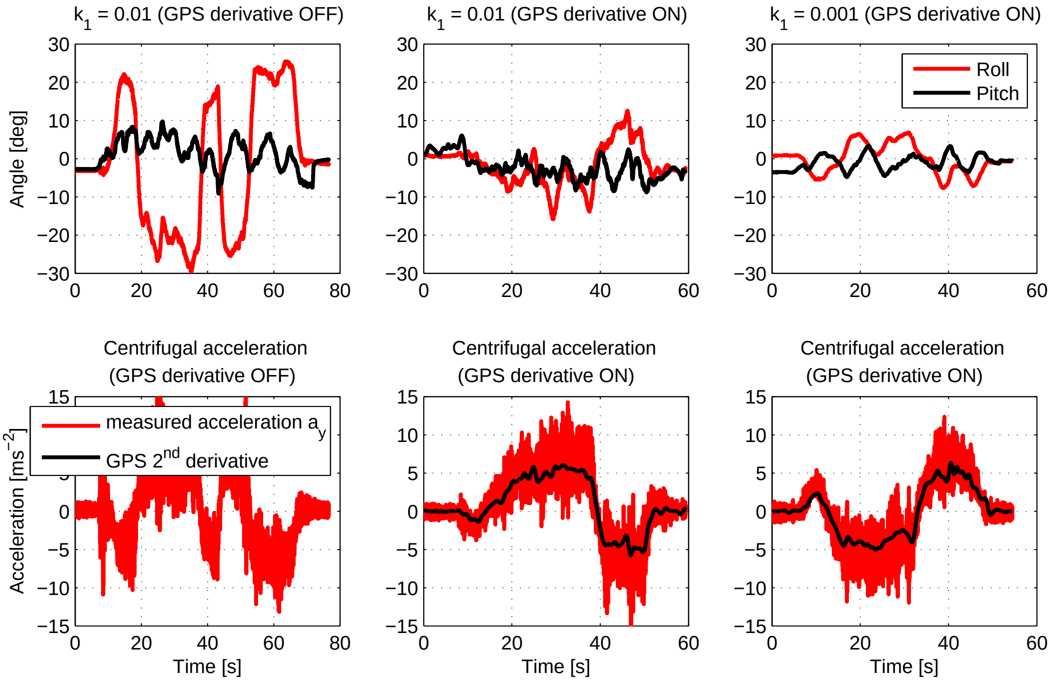


Figure 7.13: Results of three runs of the same experiment as in Fig. 7.7 — the car was driven in several circles. Top graphs show the attitude estimation — pitch and roll angles. At the bottom the centrifugal acceleration measured by accelerometer and computed through second derivative of GPS in shown.

Chapter 8

Conclusion

This thesis documented a comprehensive investigation of various research topics that were all related to the task of inertial stabilization of the camera mounted on a mobile carrier, typically an aircraft. Besides some necessary introductory (and essentially routine engineering) material which included discussion of basic concepts, mechanical configuration and the corresponding mathematical models, the thesis presented a few contributions that were published at prestigious international conferences such as IEEE CDC and IFAC World Congress and a few papers (some accepted, some submitted) at solid journals such as IEEE Transactions on Control Systems Technology, IFAC Mechatronics and IFAC Control Engineering Practice.

The scope of the thesis was fairly wide, the included topics are diverse as visual servoing, delay compensation, structured MIMO controller design and inertial estimation. This character of the doctoral research (and finally the thesis) was certainly shaped by fact that a development of series of inertially stabilized platforms was under way in the supervisors group (in collaboration with other groups including industrial partners). This helped to keep the research focused on relevant problems and provided a unique opportunity to test the results of the work in a very realistic environment, including flight tests.

Whereas in this thesis a few problems lying at the intersection between an inertial stabilization and visual servoing were systematically investigated, it seems promising to explore similarly the area at the intersection between the unmanned aerial vehicle (UAV) path planning and the onboard camera pointing and visual tracking. This could immediately initiate a new research thread in the area of UAV path planning, wherein the planning of the path of the UAV is realized in such a way that the tracked object remains observed, that is, well in the camera field of view and in desirable distance. Considering that the camera can be rotated around two or more axes adds new degrees of freedom into the optimization task. In other words, instead of commanding the UAV to *visit* this and that point on a map, the task is to *observe* this and that object.

A few more pragmatic control related issues remain in the list as well. The ever annoying issue of friction in the joints/gimbals is apparently the number one among them.

Unlike in most motion control applications here the special feature is that the system mostly operates in the velocity region close to zero. The Stribeck effect is then pronounced. A perfect mechanical design can alleviate a lot of these troubles, nonetheless, armed with the modern control theory tools one may be challenged to model and compensate for this friction. The key trouble is, however, that friction is a relative phenomenon, that is, it is associated with a relative motion of a rotor with respect to the stator, the inner gimbal with respect to the outer gimbal. Unfortunately, our inertial stabilization loops are always based on absolute velocity measured by gyros! Incorporating measurements of the relative velocities of the gimbals in the inertial angular rate stabilization loops is a challenge that does not appear to be discussed in the literature.

Bibliography

[1] J. M. Hilkert, "Inertially stabilized platform technology: concepts and principles," *Control Systems Magazine, IEEE*, vol. 28, pp. 26–46, Feb. 2008.

[2] Z. Hurák and M. Řezáč, "Image-based pointing and tracking for inertially stabilized airborne camera platform," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 5, pp. 1146 –1159, Sep. 2012.

[3] ——, "Combined line-of-sight inertial stabilization and visual tracking: application to an airborne camera platform," in *Proc. of the 48th IEEE Conference on Decision and Control*, Shanghai, China, December 2009.

[4] ——, "Control design for image tracking with an inertially stabilized airborne camera platform," in *Proc. of SPIE Deference, Security, and Sensing 2010*, Orlando, Florida, USA, April 2010.

[5] ——, "Delay compensation in a dual-rate cascade visual servomechanism," in *Proc. of the 49th Conference on Decision and Control*, Atlanta, GA, USA, December 2010.

[6] M. Řezáč and Z. Hurák, "Delay compensation in dual-rate cascade visual servomechanisms for inertial stabilization," Submitted to *Control Engineering Practice* on 16th January 2013.

[7] ——, "Structured mimo H-infinity design for dual-stage inertial stabilization: Case study for HIFOO," in *IFAC World Congress*, vol. 18, no. 1, 2011, pp. 7456–7461.

[8] M. Řezáč and Z. Hurák, "Structured mimo $\mathcal{H}_\infty$ design for dual-stage inertial stabilization: Case study for hifoo and hinfstruct solvers," Accepted for publishing in *Mechatronics* on 23rd August 2013.

[9] M. Řezáč and Z. Hurák, "Vibration rejection for inertially stabilized double gimbal platform using acceleration feedforward," in *2011 IEEE International Conference on Control Applications (CCA)*. IEEE, Sep. 2011, pp. 363–368.

[10] ——, "Low-cost inertial estimation unit based on extended kalman filtering," in *Automatic Target Recognition XX; Acquisition, Tracking, Pointing, and Laser Systems Technologies XXIV; and Optical Pattern Recognition XXI*, vol. 7696. Orlando, Florida, USA: SPIE, Apr. 2010, pp. 76 961F–10.

[11] A. Rue, “Stabilization of precision electrooptical pointing and tracking systems,” *Aerospace and Electronic Systems, IEEE Transactions on*, vol. AES-5, pp. 805–819, 1969.

[12] ——, “Precision stabilization systems,” *Aerospace and Electronic Systems, IEEE Transactions on*, vol. AES-10, pp. 34–42, 1974.

[13] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control.* Wiley, 2006.

[14] J. Osborne, G. Hicks, and R. Fuentes, “Global analysis of the double-gimbal mechanism,” *Control Systems Magazine, IEEE*, vol. 28, pp. 44–64, 2008.

[15] B. Ekstrand, “Equations of motion for a two-axes gimbal system,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 37, no. 3, pp. 1083–1091, Jul. 2001.

[16] C. de Wit, H. Olsson, K. J. Åström, and P. Lischinsky, “A new model for control of systems with friction,” *IEEE Transactions on automatic control*, vol. 40, no. 3, pp. 419–425, 1995.

[17] F. Altpeter, “Friction modeling, identification and compensation,” Ph.D. dissertation, École Polytechnique Fédérale de Lausanne, 1999.

[18] H. Olsson, K. J. Åström, C. C. D. Wit, M. Gäfvert, and P. Lischinsky, “Friction models and friction compensation,” *European Journal of Control*, vol. 4, pp. 176–195, 1998.

[19] R. Hensen, “Controlled mechanical systems with friction,” Ph.D. dissertation, Technische Universiteit Eindhoven (TU/e), 2002.

[20] F. Al-Bender and J. Swevers, “Characterization of friction force dynamics,” *Control Systems Magazine, IEEE*, vol. 28, no. 6, pp. 64–81, 2008.

[21] K. Åström and C. Canudas-de-Wit, “Revisiting the LuGre friction model,” *Control Systems Magazine, IEEE*, vol. 28, no. 6, pp. 101–114, 2008.

[22] A. Harnoy, B. Friedland, and S. Cohn, “Modeling and measuring friction effects,” *Control Systems Magazine, IEEE*, vol. 28, no. 6, pp. 82–91, 2008.

[23] G. Herrmann and G. Guo, “HDD dual-stage servo-controller design using a μ-analysis tool,” *Control Engineering Practice*, vol. 12, no. 3, pp. 241–251, Mar. 2004.

[24] R. Horowitz, Y. Li, K. Oldham, S. Kon, and X. Huang, “Dual-stage servo systems and vibration compensation in computer hard disk drives,” *Control Engineering Practice*, vol. 15, no. 3, pp. 291–305, 2007.

[25] T. Suthasun, I. Mareels, and A. Al-Mamun, “System identification and controller design for dual actuated hard disk drive,” *Control engineering practice*, vol. 12, no. 6, pp. 665–676, 2004.

[26] S. Schroeck, W. Messner, and R. McNab, “On compensator design for linear time-invariant dual-input single-output systems,” *IEEE/ASME Transactions on Mechatronics*, vol. 6, no. 1, pp. 50–57, 2001.

[27] S. Woody and S. Smith, “Design and performance of a dual drive system for tip-tilt angular control of a 300 mm diameter mirror,” *Mechatronics*, vol. 16, no. 7, pp. 389–397, 2006.

[28] J. Zheng, A. Salton, and M. Fu, “Design and control of a rotary dual-stage actuator positioning system,” *Mechatronics*, vol. 21, no. 6, pp. 1003–1012, September 2011.

[29] J. Lyou, M. Kang, H. Kwak, and Y. Choi, “Dual stage and an image processing-based method for sight stabilization,” *Journal of Mechanical Science and Technology*, vol. 23, no. 8, pp. 2097–2106, 2009.

[30] K. J. Åström and T. Hägglund, *Advanced PID Control.* The Instrumentation, Systems, and Automation Society, Aug. 2005.

[31] L. Ray, A. Ramasubramanian, and J. Townsend, “Adaptive friction compensation using extended kalman-bucy filter friction estimation,” *Control Engineering Practice*, vol. 9, no. 2, pp. 169–179, 2001.

[32] B. Francis and W. Wonham, “The internal model principle of control theory,” *Automatica*, vol. 12, no. 5, pp. 457–465, Sep. 1976.

[33] P. Lambrechts, M. Boerlage, and M. Steinbuch, “Trajectory planning and feedforward design for electromechanical motion systems,” *Control Engineering Practice*, vol. 13, no. 2, pp. 145–157, Feb. 2005.

[34] R. Eloundou, “Interpretation of smooth reference commands as input-shaped functions,” in *American Control Conference, 2002. Proceedings of the 2002*, vol. 6, 2002, pp. 4948–4953 vol.6.

[35] K. Astrom and B. Wittenmark, *Computer-controlled systems: theory and design.* prentice hall Englewood Cliffs, NJ, 1990, vol. 3.

[36] M. Masten and L. Stockum, Eds., *Selected Papers on Precision Stabilization and Tracking Systems for Acquisition, Pointing and Control Applications*, ser. SPIE Milestone Series, vol. MS 123. SPIE, 1996.

[37] M. Masten, “Inertially stabilized platforms for optical imaging systems: Tracking dynamic dynamic targets with mobile sensors,” *Control Systems Magazine, IEEE*, vol. 28, pp. 47–64, Feb. 2008.

[38] J. Debruin, “Control systems for mobile satcom antennas,” *Control Systems Magazine, IEEE*, vol. 28, pp. 86–101, 2008.

[39] J. Burke, D. Henrion, A. Lewis, and M. Overton, “HIFOO - a Matlab package for fixed-order controller design and H_∞ optimization,” in *Proceedings of the 5th IFAC Symposium on Robust Control Design*, D. Arzelier, Ed. Toulouse: IFAC, Jul. 2006, pp. 339–344.

[40] S. Gumussoy and M. L. Overton, “Fixed-order $\mathcal{H}_\infty$ controller design via hifoo, a specialized nonsmooth optimization package,” in *American Control Conference, 2008.* IEEE, 2008, pp. 2750–2754.

[41] S. Gumussoy, D. Henrion, M. Millstone, and M. Overton, “Multiobjective robust control with HIFOO 2.0,” in *Proceedings of the IFAC Symp. on Robust Control Design*, Haifa, June 2009.

[42] J. Burke, D. Henrion, A. Lewis, and M. Overton, “Stabilization via nonsmooth, nonconvex optimization,” *IEEE Transactions on Automatic Control*, vol. 51, no. 11, pp. 1760–1769, 2006.

[43] A. Popov, H. Werner, and M. Millstone, “Fixed-structure discrete-time $\mathcal{H}_\infty$ controller synthesis with HIFOO,” in *2010 49th IEEE Conference on Decision and Control (CDC)*, 2010, pp. 3152–3155.

[44] P. Gahinet and P. Apkarian, “Structured h_∞ synthesis in Matlab,” in *Proceedings of the 18th IFAC World Congress.* Milan, Italy: IFAC, Aug. 2011, pp. 1435–1440.

[45] P. Apkarian and D. Noll, “Nonsmooth $\mathcal{H}_\infty$ synthesis,” *Automatic Control, IEEE Transactions on*, vol. 51, no. 1, pp. 71–86, 2006.

[46] C. S. Chen and Y. T. Teng, “Multirate digital servo drive based on acceleration observer and disturbance compensator,” in *Proceedings of the 18th IFAC World Congress*, 2008, pp. 9278–9283.

[47] S. Pannu and R. Horowitz, “Adaptive accelerometer feedforward servo for disk drives,” in *Proceedings of the 36th IEEE Conference on Decision and Control*, vol. 5, 1997, pp. 4216–4218 vol.5.

[48] M. T. White and M. Tomizuka, “Increased disturbance rejection in magnetic disk drives by acceleration feedforward control and parameter adaptation,” *Control Engineering Practice*, vol. 5, no. 6, pp. 741–751, Jun. 1997.

[49] R. Kumar, H. Sawhney, S. Samarasekera, S. Hsu, H. Tao, Y. Guo, K. Hanna, A. Pope, R. Wildes, D. Hirvonen, M. Hansen, and P. Burt, “Aerial video surveillance and exploitation,” *Proceedings of the IEEE*, vol. 89, no. 10, pp. 1518–1539, 2001.

[50] J. Nygårds, P. Skoglar, M. Ulvklo, and T. Högström, “Navigation aided image processing in UAV surveillance: Preliminary results and design of an airborne experimental system,” *Journal of Robotic Systems*, vol. 21, no. 2, pp. 63–72, Feb. 2004.

[51] Q. Zhu, K. Cheng, and H. Zhang, “SSD tracking using dynamic template and log-polar transformation,” in *Multimedia and Expo, 2004. ICME ’04. 2004 IEEE International Conference on*, vol. 1, 2004, pp. 723–726 Vol.1.

[52] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *Proceedings of the 7th international joint conference on Artificial intelligence-Volume 2*, 1981, pp. 674–679.

[53] D. Comaniciu, V. Ramesh, and P. Meer, “Kernel-Based object tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–575, 2003.

[54] T. Kanade and C. Tomasi, “Detection and tracking of point features,” Carnegie Mellon University, Tech. Rep. CMU–CS-91-132, April 1991.

[55] D. G. Lowe, “Object recognition from local Scale-Invariant features,” in *Computer Vision, IEEE International Conference on*, vol. 2. Los Alamitos, CA, USA: IEEE Computer Society, 1999, p. 1150.

[56] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide-baseline stereo from maximally stable extremal regions,” *Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, Sep. 2004.

[57] M. Donoser and H. Bischof, “Efficient maximally stable extremal region (MSER) tracking,” in *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 1. Los Alamitos, CA, USA: IEEE Computer Society, 2006, pp. 553–560.

[58] P. Viola and M. Jones, “Robust real-time object detection,” *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2002.

[59] H. Grabner, J. Sochman, H. Bischof, and J. Matas, “Training sequential on-line boosting classifier for visual tracking,” in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, 2008, pp. 1–4.

[60] F. Chaumette and S. Hutchinson, “Visual servo control. i. basic approaches,” *Robotics & Automation Magazine, IEEE*, vol. 13, no. 4, pp. 82–90, 2006.

[61] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design*, 2nd ed. Wiley, Nov. 2005.

[62] G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital Control of Dynamic Systems*, 3rd ed. Prentice Hall, Dec. 1997.

[63] K. Deguchi, “Optimal motion control for image-based visual servoing by decoupling translation and rotation,” in *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, vol. 2, 1998, pp. 705–711.

[64] E. Malis, F. Chaumette, and S. Boudet, "2-1/2 d visual servoing," *Robotics and Automation, IEEE Transactions on*, vol. 15, pp. 238–250, 1999.

[65] P. Corke and S. Hutchinson, "A new partitioned approach to image-based visual servo control," *Robotics and Automation, IEEE Transactions on*, vol. 17, pp. 507–515, 2001.

[66] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: A unifying framework," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221–255, 2004.

[67] J. Clegg, "A nonlinear integrator for servomechanisms," *Trans. AIEE*, vol. 77, no. Part II, pp. 41–42, 1958.

[68] O. Beker, C. V. Hollot, Y. Chait, and H. Han, "Fundamental properties of reset control systems," *Automatica*, vol. 40, no. 6, pp. 905–915, Jun. 2004.

[69] Y. Chait and C. V. Hollot, "On Horowitz's contributions to reset control," *International Journal of Robust and Nonlinear Control*, vol. 12, no. 4, pp. 335–355, 2002.

[70] D. Wu, G. Guo, and Y. Wang, "Reset Integral-Derivative control for HDD servo systems," *Control Systems Technology, IEEE Transactions on*, vol. 15, no. 1, pp. 161–167, 2007.

[71] Y. Zheng, Y. Chait, C. V. Hollot, M. Steinbuch, and M. Norg, "Experimental demonstration of reset control design," *Control Engineering Practice*, vol. 8, no. 2, pp. 113–120, Feb. 2000.

[72] Y. Guo, Y. Wang, L. Xie, and J. Zheng, "Stability analysis and design of reset systems: Theory and an application," *Automatica*, vol. 45, no. 2, pp. 492–497, Feb. 2009.

[73] T. Sim, G. Hong, and K. Lim, "Multirate predictor control scheme for visual servo control," *Control Theory and Applications, IEE Proceedings*, vol. 149, no. 2, pp. 117–124, 2002.

[74] R. Ergon, "Modified Smith-predictor multirate control utilizing secondary process measurements," *Modeling, identification and control*, vol. 28, no. 1, pp. 15–18, 2007.

[75] H. Xie, L. Sun, W. Rong, and X. Yuan, "Visual servoing with modified Smith predictor for micromanipulation tasks," in *Mechatronics and Automation, 2005 IEEE International Conference*, vol. 1, 2005, pp. 71–76 Vol. 1.

[76] T. Chen and B. A. Francis, *Optimal Sampled-Data Control Systems.* Springer, Mar. 1996.

[77] M. Nemani, T. Tsao, and S. Hutchinson, "Multi-Rate analysis and design of visual feedback digital Servo-Control system," *Journal of Dynamic Systems, Measurement, and Control*, vol. 116, no. 1, pp. 45–55, Mar. 1994.

[78] J. Kuipers, *Quaternions and rotation sequences: a primer with applications to orbits, aerospace, and virtual reality.* Princeton Univ Pr, 2002.

[79] A. Pascoal, I. Kaminer, and P. Oliveira, “Navigation system design using time-varying complementary filters,” *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 36, no. 4, pp. 1099–1114, 2000.

[80] R. Mahony, T. Hamel, and J. Pflimlin, “Nonlinear complementary filters on the special orthogonal group,” *Automatic Control, IEEE Transactions on*, vol. 53, no. 5, pp. 1203–1218, 2008.

[81] F. Lewis, X. Lihua, and D. Popa, *Optimal and robust estimation.* CRC Press, 2007.

[82] D. Simon, *Optimal state estimation.* John Wiley & Sons, Inc., 2006.

[83] J. Farrell, *Aided Navigation: GPS with High Rate Sensors.* McGraw-Hill Professional, 2008.

[84] R. Beard, “State Estimation for Micro Air Vehicles,” *Studies in Computational Intelligence*, vol. 70, pp. 173–199, 2007.

[85] T. Hamel and R. Mahony, “Attitude estimation on SO (3) based on direct inertial measurements,” in *International Conference on Robotics and Automation, ICRA2006*, 2006.

[86] E. Bachmann, I. Duman, U. Usta, R. McGhee, X. Yun, and M. Zyda, “Orientation tracking for humans and robots using inertial sensors,” in *International Symposium on Computational Intelligence in Robotics & Automation (CIRA 99)*, 1999, pp. 187–194.

[87] J. Marins, X. Yun, E. Bachmann, R. McGhee, and M. Zyda, “An extended Kalman filter for quaternion-based orientation estimation using MARG sensors,” in *Proceedings of the 2001 IEEE/RSJ, International Conference on Intelligent Robots and Systems, Maui, Hawaii*, 2001.

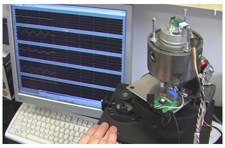
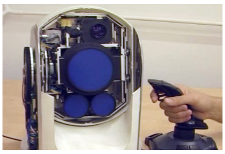
[88] T. Harada, T. Mori, and T. Sato, “Development of a tiny orientation estimation device to operate under motion and magnetic disturbance,” *The International Journal of Robotics Research*, vol. 26, no. 6, pp. 547–559, 2007.

[89] A. Kim and M. Golnaraghi, “A quaternion-based orientation estimation algorithm using an inertial measurement unit,” in *Position Location and Navigation Symposium*, 2004.

[90] S. Kumar and T. Jann, “Estimation of attitudes from a low-cost miniaturized inertial platform using kalman filter-based sensor fusion algorithm,” in *SADHANA - Academy Proceedings in Engineering Sciences*, vol. 29. Indian Academy of Sciences, 2004, pp. 217–235.

[91] W. Press *et al.*, *Numerical Recipes in C++*, 2nd ed. Cambridge university Press, 2002.

Videos attached on CD-ROM

Thumbnail	File name	Description & www link
  INERTIALLY STABILIZED CAMERA car tracking test on a helicopter	`tracking_demo.avi`	Tracking demo with S250 platfrom from MI-17 helicopter test. Green rectangle represents the target to be tracked. The aim of the control system is to ensure that the target is in the center of the image. `http://www.youtube.com/watch?v=bxidJ_pLJ88`
	`H240-LOS.avi`	Line-of-sight stabilization experiment with the H240 double-gimbal platform. Video combines the outer view of the platform being moved in hands while showing the video output from the camera. `http://www.youtube.com/watch?v=Wt4wWgB1YNc`
	`heli_rotate.avi`	Line-of-sight experiment with tracking of the standing target (a cart on a runway) using S250 four-joint platform during flight experiment in MI-17 helicopter. `http://www.youtube.com/watch?v=0EfsIJbEzx4`
	`dual-stage.avi`	Line-of-sight stabilization using the single-axis dual-stage benchmark model. Video shows the top part of the model with the voice coil motor floating as there is a base movement disturbance created by hand. `http://youtu.be/F5N3WkDDRZM`
	`S250.avi`	Joystick manipulation and Line-of-sight stabilization using S250 platform – laboratory experiment. During manipulation by joystick it is observable that inner and outer stages move with respect to each other.

	`CTPort2012.mov`	One episode of popular czech TV show PORT broadcasted by Czech television. Episode was devoted to inertially stabilized platform S250. [czech language]
		`http://www.ceskatelevize.cz/porady/10121359557-port/`
		`772-pripad-kamerove-hlavice/`

Vita

Martin Řezáč was born in 1984 in Přerov, Czech republic. He received his Ing. degree (equivalent to M.Sc.) in cybernetics and measurement with major in control engineering (summa cum laude) at Faculty of Electrical Engineering, Czech Technical University in Prague, Prague, Czech Republic, in 2008. He was one of the two key developers of the control systems for the stabilized platforms developed at CTU. During his doctoral studies he also spent six months in Eindhoven University of Technology, The Netherlands, thanks to Erasmus Socrates program. Currently (as of 2013) he is employed as a research engineer in Porsche Engineering Services company.

Address: Department of Control Engineering, Faculty of Electrical Engineering
Czech Technical University
Karlovo náměstí 13/E, 12135, Prague
Tel: +420 733114523
E-mail: rezac.martin@centrum.cz
Web: `http://cz.linkedin.com/pub/martin-rezac/32/204/848`

This dissertation was typeset with LaTeX 2_ε[1].

The acknowledgement for the design of the cover of this thesis belongs to Ing. arch. Jan Řezáč.

[1] LaTeX 2_ε is an extension of LaTeX. LaTeX is a collection of macros for TeX. TeX is a trademark of the American Mathematical Society.