Chapter 27
An Experimental Test Setup for Advanced Estimation and Control of an Airborne Wind Energy System

Kurt Geebelen, Milan Vukov, Andrew Wagner, Hammad Ahmad, Mario Zanon, Sebastien Gros, Dirk Vandepitte, Jan Swevers, Moritz Diehl

Abstract This chapter gives a detailed description of a test setup developed at KU Leuven for the launch and recovery of unpropelled tethered airplanes. The airplanes are launched by bringing them up to flying speed while attached by a tether to the end of a rotating arm. In the development of the setup, particular care was taken to allow experimental validation of advanced estimation and control techniques such as moving horizon estimation and model predictive control. A detailed overview of the hardware, sensors and software used on this setup is given in this chapter. The applied estimation and control techniques are outlined in this chapter as well, and an analysis of the closed loop performance is given.

27.1 Introduction

This chapter gives a detailed description of a test setup developed at KU Leuven. The setup can be used for the launch and recovery of unpropelled tethered airplanes. It has been constructed for two purposes: the experimental validation of advanced estimation and control techniques developed in the Highwind project, and to demonstrate the feasibility of a novel technique for launching and landing unpropelled tethered airplanes. In contrast to airplanes with on-board power generation, that use on-board propellers and generators to extract wind energy, no propellers are

Kurt Geebelen (✉) · Milan Vukov · Mario Zanon · Sébastien Gros · Andrew Wagner · Moritz Diehl
KU Leuven, Department of Electrical Engineering, Kasteelpark Arenberg 10, 3001 Leuven, Belgium, e-mail: kurt.geebelen@mech.kuleuven.be

Dirk Vandepitte · Jan Swevers
KU Leuven, Department of Mechanical Engineering, Celestijnenlaan 300C, 3001 Leuven, Belgium.

Hammad Ahmad
Mobile Marine Robotics Research Center, University of Limerick, Ireland.

needed on-board for a pumping power generation cycle, where the airplane drives a generator on the ground. This reduces the overall weight of the airborne system, but requires a different startup methodology than when a powered Vertical Take-Off and Landing (VTOL) approach is used, for example, by the company Makani Power [10]. In a rotational start procedure, the airplane is brought up to speed by an arm rotating around a central axis. Once the airplane has gained enough speed, the tether can be unrolled, allowing the airplane to gain altitude. The developed setup also allows us to perform experiments at high velocity in a limited space. This is a major benefit when it comes to testing the control system, since it can be done in a more controlled environment. The outline of this chapter is as follows: Sect. 27.2 describes the experimental test setup in detail, Sect. 27.3 describes the control architecture that is used to perform closed loop experiments, Sect. 27.4 gives an analysis of the closed loop experiments, and Sect. 27.5 formulates the conclusions.

27.2 Experimental setup

This section gives a detailed overview of the test setup developed at the KU Leuven. Sect. 27.2.1 describes the hardware components of the setup, Sect. 27.2.2 describes the sensors that are available on the setup and Sect. 27.2.3 describes the software that interfaces the hardware, sensors and algorithms.

27.2.1 Hardware

The idea behind the setup in terms of airborne wind energy is to perform a rotational start and landing of a tethered airplane. It will, however, also serve as a platform to experimentally validate advanced estimation and control algorithms such as Moving Horizon Estimation and Model Predictive Control, which are outlined in Chap. 12. Because of the limited space required to perform the initial phase of the rotation start, the setup can be placed indoors, which allows testing the system before moving to a more unpredictable outdoors environment. It also allows experiments to be performed all year round, independently of the weather conditions. The available indoors space is 6 by 6 m, and is enclosed by nets for safety reasons. This space limits the arm length to 1 m. The carousel is designed with a provision to increase the arm length and hence increase the towing capability of the carousel. The height of the setup is chosen to be 2.5 m, allowing some ground clearance for the airplane. The test-setup is shown in Fig. 27.1. It is designed for high stiffness, such that the forces of the airplane have minimal influence on the structure of the carousel. The carousel can rotate at a maximum speed of 60 rotations per minute with the airplane attached to it. A winch with a power of 400 W is placed on the carousel to control the tether length.

The airplane used for the experiments is an Ariane P5, shown in Fig. 27.1, which is typically used for F5D class radio controlled model aircraft pylon racing. These airplanes have to take sharp turns at high velocities, and are therefore built from carbon fiber, giving them high strength and stiffness. The Ariane P5 has a wingspan of 1 m, a surface area of 0.1 m^2 and a weight of 0.6 kg when fully equipped with all sensors, actuators and the micro-controller. It has ailerons and elevators to control lateral and longitudinal dynamics, but lacks a rudder. To increase the control authority of the airplane, the ailerons and elevators are enlarged with carbon plates. This is needed because the tether is attached to the fuselage 2.5 cm under the center of mass, resulting in a restoring torque when the airplane rolls or pitches. The original control surfaces cannot provide enough torque to overcome this effect at short tether lengths.

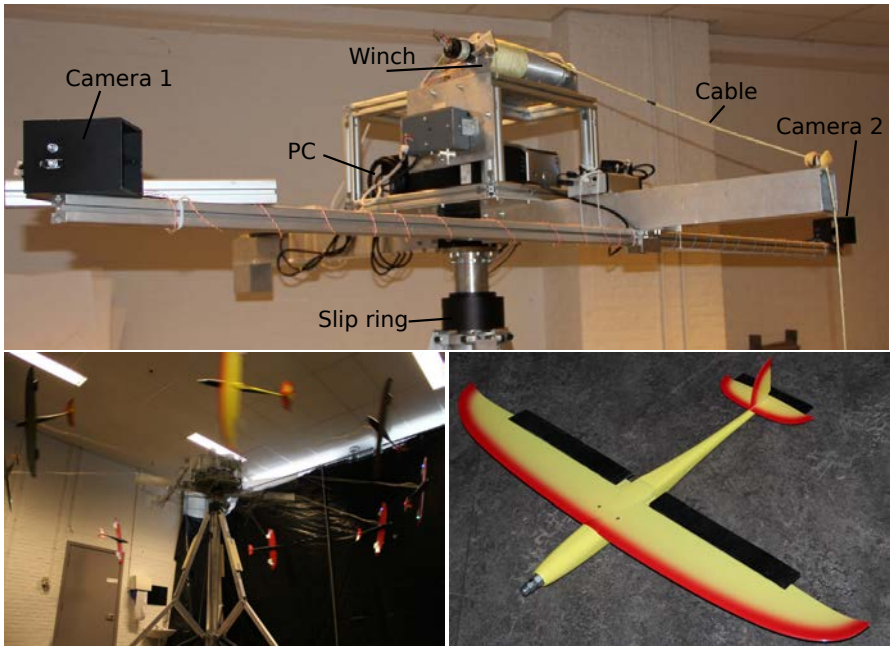

Fig. 27.1 **Top:** Close-up of top of carousel. **Bottom-Left:** KU Leuven test setup. **Bottom-Right:** Ariane P5.

The tether used in this AWE setup contains 3 pairs of insulated copper wires. Two pairs are used for the communication between the micro-controller inside the airplane and the ground computer. The third pair is used to provide 12 V DC at maximum 5 A. The copper pairs are enclosed in a sheath of braided Kevlar that takes the mechanical tension. The outer diameter of the cable is 3.6 mm. Due to space limitations inside the plane and the need to deal with voltage drop in the tether, a custom on-board power distribution circuit is used to power the on-board instrumentation. Four linear regulators are used for stepping down the voltage from 12 V to 6 V and

5 V respectively. A bank of capacitors with a total capacitance of 1600 μF is used to handle current load spikes when all servos are actuated concurrently (see Fig. 27.2).

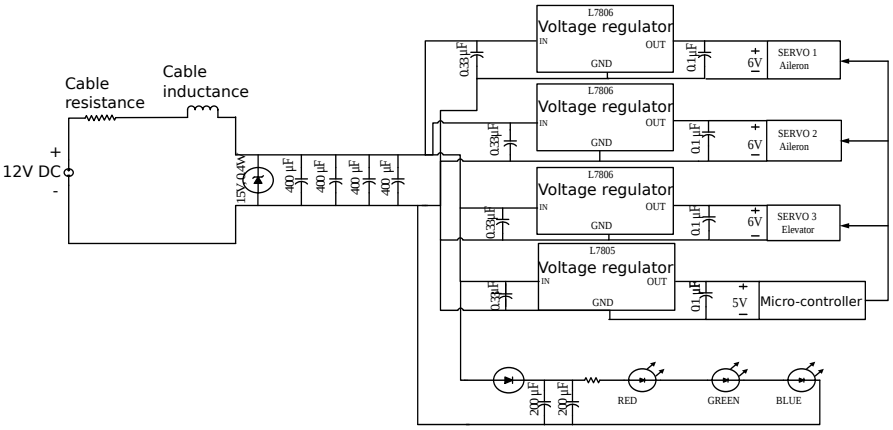

Fig. 27.2 Cable interaction and on-board power electronics

27.2.2 Sensors

The test setup is equipped with several sensors providing information on the position and orientation of the airplane, as well as on the angle of the carousel. The sensors available on the test setup include an encoder for measuring the carousel angle, a stereo vision system and an Inertial Measurement Unit (IMU). It will be the task of the Moving Horizon Estimator (MHE), outlined in Sect. 27.3.2, to fuse these measurements to form a state estimate. Therefore, particular care was taken to synchronize the different sensors. This makes the task of the estimator easier, since it does not have to cope with measurements taken on an irregular grid. Fig. 27.3 gives an overview of the instrumentation of the setup.

The **stereo vision system** consists of two Point Grey Flea3 cameras with a resolution of 1600 by 1200 pixels that can take images at a maximum sampling frequency of 15 Hz. The cameras are mounted on the carousel and observe three markers (red, green and blue LEDs) mounted on the airplane. This provides information to estimate the position and orientation of the airplane.

The **six degrees of freedom IMU** (Analog Devices ADIS16367) measures linear acceleration and angular velocities of the airplane in all directions. The maximum sampling frequency of the IMU is 800 Hz, which is fast enough to capture the major mechanical modes of the tethered aircraft.

The bulk of the processing power is provided by a PC mounted on the carousel that has an Intel Core i7 860 2.8 GHz quad-core processor, 6 GB RAM memory and

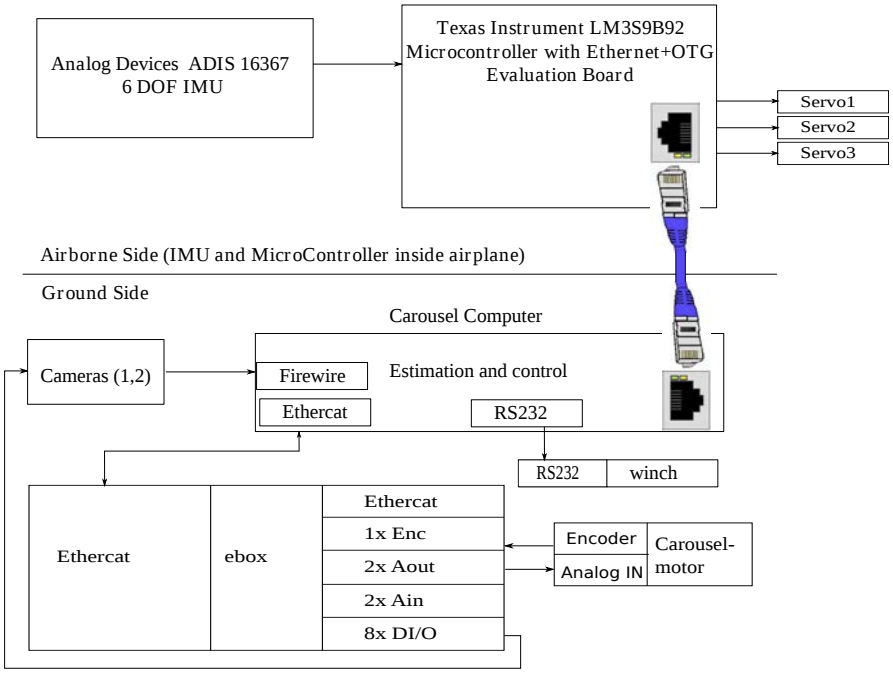

Fig. 27.3 Schematic for instrumentation of the setup

is running Ubuntu Linux on a Xenomai kernel. The data acquisition system consists of a micro-controller (Texas Instruments LM3S9B92) mounted on the plane that communicates with the PC via Ethernet, and an E-box [2] that interfaces with the carousel motor drives, angular encoder, and the camera triggers.

27.2.3 Software layout

Another important part of the test set-up is the software architecture used to control it. The chosen 'Open Robot Control Software' (Orocos) Toolchain [1] is an open source software framework for real-time control of robotic and mechatronic systems. The key feature of Orocos is the Real-Time Toolkit, a C++ programming framework that supports hard real-time data-flow programming. This makes it easier for programmers to write hard real-time programs with correct, deterministic timing. Furthermore, a variety of drivers for common sensors and actuators are available that are pre-packaged as Orocos components. Every sensor, actuator, and algorithm in our system is wrapped in an Orocos component. Each component typically has multiple inputs and outputs, and the components are then linked together at runtime,

forming a closed loop system. Figure 27.4 shows the layout of all Orocos components in the current closed loop system.

The **Camera** components are responsible for triggering the cameras to take an image. At the same time, the **IMU** on the airplane and the **encoder** on the carousel are triggered to take a measurement. Once the camera images are transferred to the computer, they are processed by the **LEDTracker**, that finds the positions of all LEDs in both cameras. The **IMU Buffer** component is responsible for averaging the IMU measurements over 1 period and outputting the average at the same sampling rate the cameras are running. The **Pose from markers** component computes the position and orientation (pose) of the airplane directly from the marker positions. It is used for initialization of the state estimator. The **MHE** then takes in all measurements and computes the state estimate. This state estimate is then passed to the **Nonlinear Model Predictive Controller (NMPC)**, which also gets the reference that we want to track from the **Trajectory generator**. The NMPC component then computes the control output and passes it to the **airplane's** control surfaces. The control output is also fed back to the MHE component. The **carousel** is currently not controlled, but is running at a constant velocity.

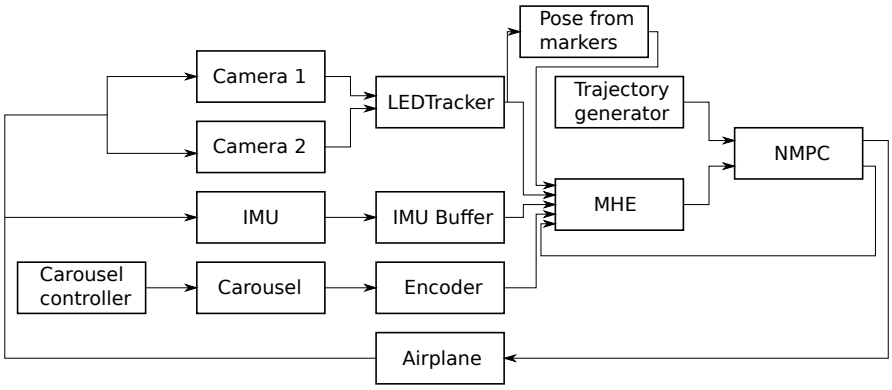

Fig. 27.4 Software overview

27.3 Control architecture

The goal of the experimental setup is to perform closed loop experiments using Moving Horizon Estimation (MHE) and Nonlinear Model Predictive Control (NMPC). This section describes the setup of the MHE and NMPC . We start with a short description of the model used in both the estimator and the controller.

27.3.1 Model equations

The state vector $\mathbf{x}$ of the model consists of the following components: $\{p, \dot{p}, R, \omega, \delta, \dot{\delta}, \delta_{\text{ail}}, \delta_{\text{elev}}\}$. The position p and velocity $\dot{p}$ of the airplane is expressed through a set of Cartesian coordinates $\{x, y, z\}$ and $\{\dot{x}, \dot{y}, \dot{z}\}$ relative to a rotating reference frame attached to the carousel arm tip. The orientation of the airplane reference frame w.r.t. the rotating reference frame is denoted by the rotation matrix R. The angular velocity of the airplane is denoted by ω. The angle and angular velocity of the carousel arm are denoted by δ and $\dot{\delta}$ respectively. The angles of the ailerons and elevators are denoted by δ_{ail} and δ_{elev} respectively. The tether length is constant in the experiments, and therefore is not part of the state vector.

The control input is given by $\mathbf{u} = [\ddot{\delta}, \dot{\delta}_{\text{ail}}, \dot{\delta}_{\text{elev}}]$: the angular acceleration of the carousel and the angular velocities of the control surfaces. By controlling the angular velocity of the control surfaces and not their angles directly, the dynamics of the control surfaces can be incorporated by putting bounds on these angular velocities. In case the angles would be controlled directly, the controller could apply discrete jumps in these angles, which in practice is impossible because of the dynamics of the control surfaces. The dynamic model of the plane is derived as an index-3 DAE. Using index reduction techniques, it is reduced to an index-1 DAE, together with some consistency conditions that need to be imposed at a certain time when simulating the model. If they are imposed at one point, they are preserved when integrating the model equations. A detailed description of the model can be found in [5] and in Chap. 10.

27.3.2 Moving Horizon Estimation

The Kalman filter is the most commonly used tool for state estimation and sensor fusion, and it is optimal in a least squares sense for linear systems with Gaussian noise. In case of non-linear dynamics, an extended or unscented Kalman filter can be used, but there are no optimality results for most nonlinear systems. Another approach comes in the form of Moving Horizon Estimation. A MHE estimates the system state by solving a nonlinear fitting problem based on a window of past measurements in real-time, and is described in Chap. 12. In our setup, the following optimization problem is solved at each sampling time:

$$\begin{aligned} \underset{\mathbf{x}_k, \mathbf{u}_k}{\text{minimize}} \quad & \sum_{k=-N}^{0} \| \mathbf{y}_k - h(\mathbf{x}_k) \|^2_{R^{-1}} + \sum_{k=-N}^{-1} \| \mathbf{u}_k^{\text{sent}} - \mathbf{u}_k \|^2_{Q^{-1}} \\ \text{subject to} \quad & \mathbf{x}_{k+1} = \phi(\mathbf{x}_k, \mathbf{u}_k),\ k = -N, \ldots, -1 \\ & c(\mathbf{x}_0) = 0 \end{aligned} \tag{27.1}$$

where $\mathbf{x}_k$ is the system state, $\mathbf{y}_k$ represents the measurements, $\mathbf{u}_k^{\text{sent}}$ are the control inputs that were sent to the system and $\mathbf{u}_k$ are the estimates of the MHE for these

control inputs. Q and R are the covariance matrices of the model and measurement noise respectively, N is the considered time horizon, f is the model of the system, h is the measurement function and c are the consistency conditions of the problem that are imposed at the current time.

The difference between the reference control inputs $\mathbf{u}_k^{\text{sent}}$ and the control estimates $\mathbf{u}_k$ can be viewed as model uncertainty. This term accounts for variations of the state that were not predicted by the model and reference inputs, typically called the process noise. It also accounts for the fact that the actual position of the actuators might not be precisely known, because of the unknown performance of the low-level controllers that control the position of the actuators. The noise on the measurements is equal to the difference between the actual measurements and these predicted by the state estimate. The problem thus minimizes a weighted sum of the process noise and the measured versus the estimated output of the system over the considered time horizon.

When a new measurement arrives, the horizon is shifted by one sampling instant and the problem is reformulated and solved. The new problem is initialized with the solution of the previous problem. Initialization of the state on the last time instant of the time horizon is done by forward integration of the model equations.

27.3.2.1 Measurements

The complete measurement vector is given by $\{\delta, \mathbf{a}_{\text{IMU}}, \omega_{\text{IMU}}, \mathbf{m}\}$. Angle δ is the measurement of the carousel angle, $\mathbf{a}_{\text{IMU}}$ and ω_{IMU} are the measurements of the acceleration and angular velocity of the airplane, and $\mathbf{m}$ are the positions of the three markers in the images of both cameras. All measurements are taken at the same time. The camera measurements come with a delay of about 150 ms. Half of this is due to the time needed to transfer the picture from the cameras to the PC, the other half is due to computation time needed to extract the marker positions from the camera images. Hence there is are no marker measurements for the last two nodes in the estimation horizon. The estimation horizon used in this estimator is 1 s. Since the measurements are taken at 10 Hz, this results in 10 estimation intervals. In total, the entire horizon thus comprises 185 measurements: 11 from the encoder, 66 from the IMU and 108 from the marker measurements. Added to this are the signals that were sent to the angles of the control surfaces (ailerons and elevator). They are not directly measured, but angle references are given to the servo motors' internal controllers. The angular velocities of the control surfaces that are given by the controller are also given as a measurement to the MHE.

The measurement function for the stereo vision system is based on a pinhole camera model, given by

$$\begin{bmatrix} us \\ vs \\ s \end{bmatrix}_{ij} = \left[\mathbf{P}_{\text{Cam}_i}\right] \left[\mathbf{R}^{\text{R}}_{\text{Cam}_i}\right] (\mathbf{p}_{\text{Cam}_i} + \mathbf{p}_j^R), \tag{27.2}$$

where u and v are the pixel coordinates of the marker, s is the homogeneous scaling factor, $\mathbf{P}_{\mathrm{Cam}_i}$ is the matrix of intrinsic parameters of camera i that depends on the focal length and the principal point of the camera, $\mathbf{R}^{\mathrm{R}}_{\mathrm{Cam}_i}$ is the rotation matrix representing the orientation of the reference frame in the camera frame, $\mathbf{p}_{\mathrm{Cam}_i}$ is the position of the camera frame in the reference frame and $\mathbf{p}_j^R$ is the position of marker j in the reference frame. The relationship between the position of marker i and the position and orientation of the airplane in the reference frame is given by:

$$\mathbf{p}_j^R = \mathbf{p} + \mathbf{R}\mathbf{p}_j^B \tag{27.3}$$

where $\mathbf{p}_j^B$ is the position of marker j in the body frame. Equations 27.2 and 27.3 give the relation between the position and orientation of the airplane and position of a marker in a camera.

The measurement function for the acceleration measurements of the IMU comes from the acceleration of the airplane in the carousel frame, which is given by the model equations. We transform it to the airplane frame, and add gravity. The measurements of the angular velocities provided by the IMU are direct measurements of ω, accounted for the orientation of the IMU inside the airplane.

This moving horizon estimator is implemented in the ACADO Toolkit [8] and C-code tailored for this specific problem was exported with the ACADO Code Generation tool [4]. The weighting matrices are fed online to the estimator. This way, certain measurements can be given a lower or zero weight in case they are missing. This sometimes happens for the stereo vision measurements when a marker is hidden from a camera by the tether. On the last two sampling times, there is also no measurement from the stereo vision system, because the transfer and marker detection time takes 2 sampling periods. By putting the weight for these measurements to zero, this time delay in the measurements can be tackled in an elegant manner.

27.3.3 Model Predictive Control

Nonlinear Model Predictive Control is an ideal framework when dealing with the control of nonlinear, constrained systems, and has been previously applied to airborne wind energy systems in e.g. [3, 6, 7, 9]. At each sampling interval we look for the control action that optimizes a certain objective function, subject to a combination of (nonlinear) dynamical, input, and state constraints. The objective can be designed for tracking a certain reference trajectory, but may also be a performance measure that has to be optimized.

In our setup, the objective is to track a pre-computed reference state trajectory. The following optimization problem is solved at each sampling time:

$$
\begin{aligned}
\underset{\mathbf{x}_k,\mathbf{u}_k}{\text{minimize}} \quad & \sum_{k=0}^{N-1} \| \mathbf{x}_k - \mathbf{x}_k^{\text{ref}} \|_V^2 + \sum_{k=0}^{N-1} \| \mathbf{u}_k - \mathbf{u}_k^{\text{ref}} \|_W^2 + \| \mathbf{x}_N - \mathbf{x}_N^{\text{ref}} \|_S^2 \\
\text{subject to} \quad & \mathbf{x}_{k+1} = \phi(\mathbf{x}_k, \mathbf{u}_k),\ k = 0, \ldots, N-1 \\
& \underline{\delta}_{\text{ail}} \leq \delta_{\text{ail}_k} \leq \bar{\delta}_{\text{ail}},\ k = 0, \ldots, N \\
& \underline{\delta}_{\text{elev}} \leq \delta_{\text{elev}_k} \leq \bar{\delta}_{\text{elev}},\ k = 0, \ldots, N \\
& \underline{\dot{\delta}}_{\text{ail}} \leq \dot{\delta}_{\text{ail}_k} \leq \bar{\dot{\delta}}_{\text{ail}},\ k = 0, \ldots, N-1 \\
& \underline{\dot{\delta}}_{\text{elev}} \leq \dot{\delta}_{\text{elev}_k} \leq \bar{\dot{\delta}}_{\text{elev}},\ k = 0, \ldots, N-1 \\
& \ddot{\delta}_k = 0,\ k = 0, \ldots, N-1 \\
& \mathbf{x}_0 = \mathbf{x}_0^{\text{est}}
\end{aligned}
\tag{27.4}
$$

where $\mathbf{x}^{\text{ref}}$ and $\mathbf{u}^{\text{ref}}$ are the reference trajectories for the state and control input respectively, V, W and S are weighting matrices, $\underline{\delta}_{\text{ail}}$, $\overline{\delta_{\text{ail}}}$, $\underline{\delta}_{\text{elev}}$, $\overline{\delta_{\text{elev}}}$, $\underline{\mathbf{u}}$ and $\bar{\mathbf{u}}$ are lower and upper limits on the control surface angles and control inputs. The angular velocity of the carousel is not controlled and is constant. $\mathbf{x}_0^{\text{est}}$ is the state estimate provided by the MHE. The consistency conditions are not part of the constraints, because the state estimate given by the MHE already satisfies these conditions.

The components of V that relate to δ_{ail} and δ_{elev} are set to zero, so that the angle of the control surfaces is not penalized. The weight on $\dot{\delta}_{\text{ail}}$ and $\dot{\delta}_{\text{elev}}$ ensures that the variation of the control input is not too large, which benefits the life span of the actuators. The terminal cost matrix S is computed by solving the Riccati equation for the system linearized around the reference state at the end of the prediction horizon. The horizon of the controller was chosen to be 1 s, with a sampling period of 0.1 s. Like the MHE, this controller was implemented in the ACADO Toolkit and specialized C-code was exported using the ACADO Code Generation tool.

27.4 Closed loop experiments

Several closed-loop experiments were performed on this test setup. The results shown here come from an experiment at a constant carousel velocity of 60 rpm that had a duration of 2.6 hours, during which the setpoint for the controller was continuously changed between two values. Here, a "setpoint" is a reference where all states aside from the carousel angle δ are constant. It would hold one setpoint for 6 s and then transition to the other setpoint. The transition from one setpoint to the other is done by a linearly interpolated ramp in the height state with a duration of 0.5 s. The setpoints are computed by computing the equilibrium state the airplane should have to fly at a certain height. These were chosen here to be -0.118 m and -0.06 m below the arm level. The lower setpoint height is the height the airplane would have when all control surfaces are neutral.

Figure 27.5 shows the closed-loop tracking performance for the roll angle, angular velocity around the z-axis and for the height z. The roll is defined here to be

0° when the wing is vertical. Note that we do not provide a setpoint for the roll directly, but it is derived from the components of the rotation matrix, which is part of the state vector. From the figure, it is clear that there is an oscillation that is not controlled. This oscillation is due the turbulence created by the fact that we are constantly flying in our own wake. Since the airplane does not have a rudder, there also is low yaw damping, which also makes the control of these oscillations hard or even impossible. Note that the current experiments are not aiming at controlling these oscillations, but rather on achieving stable closed loop MHE and NMPC results. The figures show that, except for the oscillations described before, the tracking behavior for the roll and angular velocity is reasonable. The tracking behavior for the height is worse. This makes sense, since the height is only indirectly controlled via the orientation.

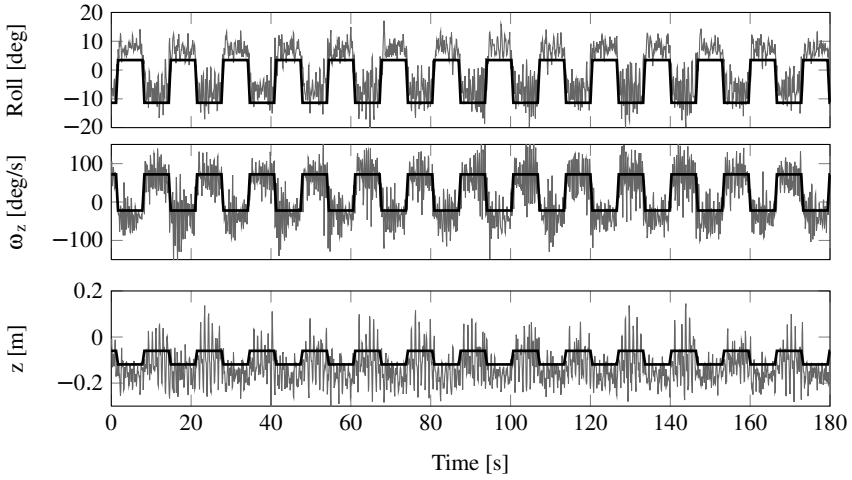

Fig. 27.5 Closed loop tracking behavior for roll, angular velocity around the z-axis and height. Reference in black, estimated variables in grey.

Table 27.1 show the low execution time of the auto-generated C-code. At each sampling time, one Real-Time Iteration (RTI) is done for both the MHE and the NMPC. Chap. 12 explains the RTI-scheme in more detail. The execution is thus split in two parts: a preparation phase that can be done before the measurements (for MHE) or the state estimate (for NMPC) have arrived, and a feedback phase that is done as soon as the measurements or state estimate arrives. The average execution time is 4.51 ms for the MHE and 4.06 ms for the NMPC. Note that, because no inequality bounds were hit in these experiments for both the MHE and the NMPC, the execution time is constant.The average feedback time is 0.75 ms and 0.50 ms for the MHE and NMPC respectively. On average it thus takes 1.39 ms to provide the feedback control signal. Execution times are measured with OROCOS timer

		Average
MHE	Preparation phase	3.76 ms
	Estimation phase	0.75 ms
	Overall execution time	4.51 ms
MPC	Preparation phase	3.56 ms
	Feedback phase	0.50 ms
	Overall execution time	4.06 ms

Table 27.1 Execution times of the MHE and NMPC

services. Those services internally use the Linux function clock_gettime(), which provides resolution in the nanosecond range.

27.5 Conclusion and future work

This chapter presented experimental verification of optimal control based state estimation and control of a nonlinear system with fast dynamics. The MHE is able to fuse measurements from a variety of sensors, and provides state estimates even when only a subset of the measurements is available. The NMPC based controller succeeded in tracking a reference trajectory, while respecting dynamic constraints as well as actuator constraints. Future work will focus on improving MHE so that it can make use of the high frequency IMU measurements [11], as well as performing experiments with varying tether length. A test setup for outdoor experiments is also under development. Compared to the current setup, it will have a larger arm of 2 m and be mounted on a trailer for mobility. It will be equipped with a 10 kW winch and a 4 kW carousel driving motor.

Acknowledgements This research was supported by Research Council KUL: PFV/10/002 Optimization in Engineering Center OPTEC, GOA/10/09 MaNet and GOA/10/11 Global real- time optimal control of autonomous robots and mechatronic systems. Flemish Government: IOF / KP / SCORES4CHEM, FWO: PhD/postdoc grants and projects: G.0320.08 (convex MPC), G.0377.09 (Mechatronics MPC); IWT: PhD Grants, projects: SBO LeCoPro; Belgian Federal Science Policy Office: IUAP P7 (DYSCO, Dynamical systems, control and optimization, 2012-2017); EU: FP7-EMBOCON (ICT-248940), FP7-SADCO (MC ITN-264735), ERC ST HIGHWIND (259 166), Eurostars SMART, ACCM.

References

1. Bruyninckx, H.: Open robot control software: the OROCOS project. In: Proceedings of the 2001 IEEE International Conference on Robotics and Automation (ICRA), Vol. 3, pp. 2523–2528, Seoul, Korea, 21–26 May 2001. doi: 10.1109/ROBOT.2001.933002

2. Eindhoven University of Technology: E-box project page. http://cstwiki.wtb.tue.nl/index.php?title=E-box. Accessed 17 July 2013
3. Ferreau, H. J.: Model predictive control algorithms for applications with millisecond timescales. Ph.D. Thesis, KU Leuven, 2011. https://lirias.kuleuven.be/bitstream/123456789/312147/1/phd_ferreau.pdf
4. Ferreau, H. J., Kraus, T., Vukov, M., Saeys, W., Diehl, M.: High-speed moving horizon estimation based on automatic code generation. In: Proceedings of the 51st IEEE Annual Conference on Decision and Control, pp. 687–692, Maui, HI, USA, 10–13 Dec 2012. doi: 10.1109/CDC.2012.6426428
5. Gros, S., Zanon, M., Diehl, M.: Orbit Control for a Power Generating Airfoil Based on Non- linear MPC. In: Proceedings of the 2012 American Control Conference, pp. 137–142, Montréal, Canada, 27–29 June 2012. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6315367
6. Gros, S., Zanon, M., Vukov, M., Diehl, M.: Nonlinear MPC and MHE for Mechanical Multi-Body Systems with Application to Fast Tethered Airplanes. In: Proceedings of the 4th IFAC Nonlinear Model Predictive Control Conference, pp. 86–93, Leeuwenhorst, Netherlands, 23–27 Aug 2012. doi: 10.3182/20120823-5-NL-3013.00061
7. Houska, B.: Robustness and Stability Optimization of Open-Loop Controlled Power Generating Kites. M.Sc.Thesis, Ruprecht-Karls-Universität, Heidelberg, 2007. http://www.kuleuven.be/optec/files/Houska2007a.pdf
8. Houska, B., Ferreau, H. J., Diehl, M.: ACADO Toolkit: An Open Source Framework for Automatic Control and Dynamic Optimization. Optimal Control Applications and Methods **32**(3), 298–312 (2011). doi: 10.1002/oca.939
9. Ilzhöfer, A., Houska, B., Diehl, M.: Nonlinear MPC of kites under varying wind conditions for a new class of large-scale wind power generators. International Journal of Robust and Nonlinear Control **17**(17), 1590–1599 (2007). doi: 10.1002/rnc.1210
10. Makani Power. http://www.makanipower.com/why-airborne-wind/. Accessed 11 July 2012
11. Quirynen, R., Vukov, M., Diehl, M.: Auto Generation of Implicit Integrators for Embedded NMPC with Microsecond Sampling Times. In: Proceedings of the 4th IFAC Nonlinear Model Predictive Control Conference, pp. 175–180, Leeuwenhorst, Netherlands, 23–27 Aug 2012. doi: 10.3182/20120823-5-NL-3013.00013