

Sensors and Navigation Algorithms for Flight Control of Tethered Kites

Michael Erhard¹ and Hans Strauch²

Abstract— We present the sensor setup and the basic navigation algorithm used for the flight control of the SkySails towing kite system. Starting with brief summaries on system setup and equations of motion of the tethered kite system, we subsequently give an overview of the sensor setup, present the navigation task and discuss challenges which have to be mastered. In the second part we introduce in detail the inertial navigation algorithm which has been used for operational flights for years. The functional capability of this algorithm is illustrated by experimental flight data. Finally we suggest a modification of the algorithms as further development step in order to overcome certain limitations.

I. INTRODUCTION

The SkySails system is a towing kite system which allows modern cargo ships to use the wind as source of power in order to save fuel and therefore to save costs and reduce emissions [1]. The SkySails company has been founded in 2001 and as main business offers wind propulsion systems for ships. Starting the development with kites of 6–10 m² size the latest generation of products, with a nominal size of 320 m², can replace up to 2 MW of the main engine’s propulsion power. Besides marine applications there are strongly increasing activities in using tethered kites and rigid wings for generating power from high-altitude wind [2], [3]. Therefore the design of control systems for tethered kites has become a growing field of experimental [4], [5] and theoretical [6], [7], [8], [9] research efforts. These control designs demand stable and reliable state estimates under changing wind conditions in order to perform their complex task in a satisfying and robust way.

In this paper we focus on the sensor setup and on the estimation algorithms used for our flight controller. The most important task is the estimation of the kite orientation. In contrast to many applications in the field of inertial navigation we do not have a sensor which provides absolute attitude information. The task of attitude estimation, based on acceleration and rotation rate sensors alone, poses a special challenge for our navigation algorithm.

Many classical inertial navigation algorithms are based on Kalman filtering [10], [11], [12] and extensions to this theory [13]. Our approach is related to complimentary filtering [14], [15] which will be discussed in the last section. We would like to emphasize that the aim of this paper is to illustrate the principle ideas and the challenges to be tackled due

We acknowledge funding from the Federal Ministries BMWI and BMBF, LIFE III of the European Commission, City of Hamburg/BWA and Innovationsstiftung Hamburg.

¹M. Erhard is with SkySails GmbH, Veritaskai 3, D-21079 Hamburg, Germany, e-mail: michael.erhard@skysails.de, http://www.skysails.de.

²H. Strauch is a consultant to SkySails.

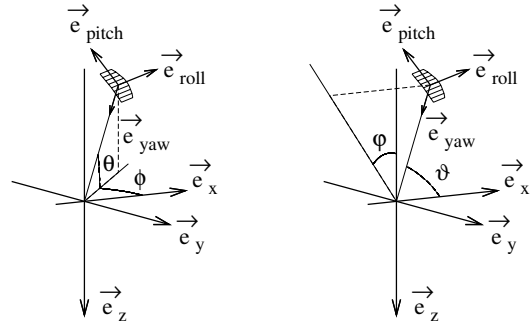


Fig. 1. Definition of coordinates for the considered system. The right-handed coordinate system is defined by the basis vectors $\vec{e}_x, \vec{e}_y, \vec{e}_z$ with \vec{e}_x in wind direction and \vec{e}_z pointing downwards with respect to gravity. The kite position is parameterized by introducing two different sets of spherical coordinates $\{\phi, \theta\}$ (left figure) and $\{\varphi, \vartheta\}$ (right figure). For a more precise definition see (2) and (3) respectively. The kite axes are labeled as roll \vec{e}_{roll} , pitch \vec{e}_{pitch} and yaw \vec{e}_{yaw} . This corresponds to the definition usually used in aerospace applications with roll axis parallel to forward and yaw axis parallel to down directions respectively. Note that for the usual flight situation the yaw vector \vec{e}_{yaw} is defined by the position of the kite assuming it is more or less constrained to the origin by a rigid rod. Thus orientation of the kite is represented by the single angle ψ_g and ψ respectively.

to system dynamics and environmental disturbances specific for our combination of sensors and towing kites. Thus the presentation also reflects the development history and does not intend to be a rigorous, ready-to-use construction manual for a general type of filter algorithm.

The paper is organized as follows: We start with short summaries of the overall system setup, the selected coordinates and the system dynamics. We then describe our sensor setup and present the attitude estimation algorithm in detail. Subsequently we explain the wind referencing setup and justify the navigation function by experimental flight data. We complete the article with a discussion of further development steps extending the previous algorithms.

II. SYSTEM SETUP AND COORDINATES

In the following we will make use of two different coordinate systems for the navigation and control algorithm, respectively. The control design is based on a coordinate system with the wind direction as symmetry axis, while the attitude estimation algorithm is referenced to the gravity (down) direction. The choice of these different coordinate systems reflect the difference in the two design tasks (see [16] for further discussion especially of the controller related aspects).

The two coordinate systems are introduced in Fig. 1. For a line length L the state of the kite is defined by the set of the three angles $\{\phi, \theta, \psi_g\}$ and $\{\varphi, \vartheta, \psi\}$ respectively. With

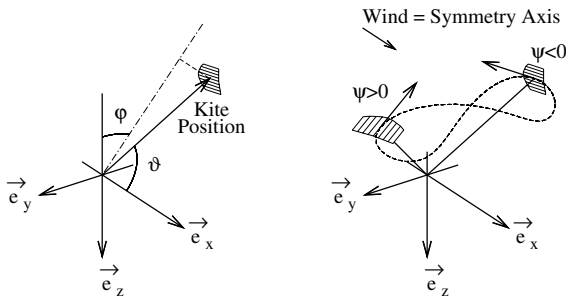


Fig. 2. Principle of system dynamics: the kite position is given by ϑ, ϕ (left figure) while the yaw angle ψ denotes the orientation with respect to the wind. The quantity ψ is crucial for controlling the flight direction of the kite, shown for the figure-eight pattern (right figure).

respect to the basis vectors $\vec{e}_x, \vec{e}_y, \vec{e}_z$ the kite position \vec{x} is determined by the set $\{\phi, \theta\}$ and $\{\varphi, \vartheta\}$ as:

$$\vec{x} = L \begin{pmatrix} \cos \phi \cos \theta \\ \sin \phi \cos \theta \\ -\sin \theta \end{pmatrix} = L \begin{pmatrix} \cos \vartheta \\ \sin \varphi \sin \vartheta \\ -\cos \varphi \sin \vartheta \end{pmatrix}. \quad (1)$$

The kite axes are denoted as \vec{e}_{roll} (roll or longitudinal), \vec{e}_{pitch} (pitch) and \vec{e}_{yaw} (yaw).

For a description based on rotation matrices one would start with a kite at position $L\vec{e}_x$, with roll-axis in negative z -direction $\vec{e}_{\text{roll}} = -\vec{e}_z$, and then apply the following rotations: first $-\psi_g$ about x , then θ about y and finally ϕ about z . This transformation reads:

$$R = R_z(\phi)R_y(\theta)R_x(-\psi_g). \quad (2)$$

The quantity ψ_g represents the angle between the \vec{e}_{pitch} -axis and the horizon or the angle between the \vec{e}_{roll} -axis and the 'upward' ($-\vec{e}_z$)-direction.

In the second coordinate system $\{\varphi, \vartheta, \psi\}$ the rotations are first $-\psi$ about x , then ϑ about y and finally φ about z . This transformation reads:

$$R = R_x(\varphi)R_y(\vartheta)R_z(-\psi). \quad (3)$$

Here one could interpret the angle ψ as orientation of the kite longitudinal axis with reference to the wind. For a given kite position \vec{x} (parameterized by φ and ϑ) the reference orientation $\psi = 0$ corresponds to the minimum of the scalar product $(\vec{e}_{\text{roll}}, \vec{e}_x)$ obtained when turning the kite fixated at this position \vec{x} around its yaw axis \vec{e}_{yaw} . A nonzero value of ψ represents a kite orientation obtained by a rotation of ψ about the yaw axis \vec{e}_{yaw} starting at this reference orientation.

Next we provide the transformation between the two coordinate systems $\{\phi, \theta, \psi_g\} \Rightarrow \{\varphi, \vartheta, \psi\}$:

$$\varphi = \arctan(\sin \phi \cos \theta, \sin \theta) \quad (4)$$

$$\vartheta = \arccos(\cos \phi \cos \theta) \quad (5)$$

$$\psi = \psi_g + \arctan(\sin \phi, \cos \phi \sin \theta). \quad (6)$$

III. DYNAMICS AND EQUATIONS OF MOTION

For the description of the dynamics we refer to Fig. 2, illustrating a dynamically flying kite in a figure-eight pattern. The position of the kite is parameterized by the previously

TABLE I
OVERVIEW OF SENSORS ABOARD THE CONTROL POD (UPPER ROWS)
AND THE SHIP (LOWER ROWS).

Sensor	Quantity	Description
Inertial Measurement Unit (IMU)	$\vec{\omega}_s$	Turn rates of the control pod below the kite
	\vec{a}_s	Accelerations of the control pod below the kite
Impeller Anemometer	v_a	Air path speed in \vec{e}_{roll} -direction
Strain Gauge Pod	F	Force measurement towing line
Barometer	h	Barometric height of the control pod
Tow Point	ϕ_s, θ_s	Angles of towing rope with respect to the ship (compare Fig. 1 left).
Ship Anemometer	v_w, ϕ_w	Apparent wind speed and direction aboard the ship.
Ship IMU		Roll and pitch angles of the ship.
Line length	L	Rotary encoder measurement of unwound towing line length

introduced coordinates φ, ϑ while the flight direction is determined by ψ . Before explaining the dynamics we give a summary of the equations of motion:

$$\dot{\psi} = g v_a \delta + \dot{\varphi} \cos \vartheta \quad (7)$$

$$\dot{\vartheta} = \frac{v_a}{L} \left(\cos \psi - \frac{\tan \vartheta}{E} \right) \quad (8)$$

$$\dot{\varphi} = -\frac{v_a}{L \sin \vartheta} \sin \psi. \quad (9)$$

The basic kite response is given mainly by the first term of (7) while $\dot{\varphi} \cos \vartheta$ is a correction due to the motion on the sphere [16]. A steering deflection δ results in a turn rate about yaw axis, scaled by g and air path speed v_a . The integrated quantity ψ represents the kite orientation with respect to the wind and thus controls the flight direction (compare (8) and (9)). E denotes the glide ratio of the kite. For an angle $\psi = 0$ the system stays more or less stationary with the kite roll axis aligned against the wind direction. In order to obtain the depicted dynamical 'figure eight' pattern one has to command a certain $\psi_s > 0$, resulting in a kite motion into negative \vec{e}_y -direction. After some time the commanded ψ_s is changed to a negative value leading to a curve and subsequently to a motion into positive \vec{e}_y -direction.

The altitude ϑ of the flight trajectory can be controlled by the amplitude of $|\psi_s|$. We would like to stop at this point and not to proceed into a detailed derivation. We just conclude that ψ is the crucial quantity for the flight controller. A detailed justification of the equations of motion and the control system architecture can be found in [16].

IV. SENSOR SETUP

An overview of the sensor setup is given in Table I. The data acquisition is performed by distributed computers running and merging the data at a main sample rate of 10 Hz.

The main sensor is the inertial measurement unit (IMU) in the control pod, which is situated under the kite including three perpendicular arranged gyroscopes and accelerometers

based on the solid state microelectromechanical systems (MEMS) technology. For some years we used the Crista OEM sensor head from Cloudcap Technology Inc. [17] which combines sensors from Analog Devices Inc. with an A/D-converter and includes a factory calibrated lookup table for gains and offsets over the operational temperature range. Later we switched to the ADIS 16355 device from Analog Devices Inc. [18] which provides a complete IMU with temperature calibrated outputs on the digital interface.

We would like to emphasize at this point that calibration and offset compensation over the operational temperature range are crucial for the performance of the navigation algorithm. In order to improve the acceleration offsets of the ADIS devices we determine the offset values on a water leveled reference plate before installation of the sensor units. Both IMUs have a measuring range of ± 300 deg/s (gyros) and ± 100 m/s² (accelerometers). The typical observed bias values (1σ) for both IMUs are in accordance with the data sheets and approximately 0.2 deg/s for the gyros and 0.2 m/s² for the accelerometers.

The IMU values are sampled at a rate of 200 Hz. 20 values are averaged for the turn rates $\vec{\omega}_s$ and for the accelerations \vec{a}_s which are then provided to the main 10 Hz cycle.

The air path speed of the kite in the range of 5–50 m/s is determined by a frequency measurement of pulses generated by an impeller anemometer.

For completeness we note that there are also a strain gauge and a barometer altimeter aboard the control pod. They are used for redundancy and guidance purposes. However, these topics will not be elaborated in this paper.

The towpoint is the deflection point of the towing rope aboard the ship. Two angular sensors determine the mechanically sensed direction of the towing line with respect to the ship as azimuth and elevation angles. Another IMU device (XSens MTi) is installed on the ship measuring the roll and pitch angles due to the wave induced motion of the ship.

The azimuth and elevation angles are merged with the roll and pitch angles in order to get the azimuth ϕ_s and elevation θ_s values. This corresponds in good approximation to an inertial reference and provides reasonable suppression of sea state influences which could easily cause ship roll amplitudes of ± 10 deg in operating conditions. We would like to point out that above transformation is correct only if the ships' center of angular motion coincides with the tow point, which is usually not the case. Yet, it is an acceptable approximation as the main dynamics takes place with respect to the roll axis.

A further sensor is the anemometer aboard the ship measuring wind speed and direction of the apparent wind. Although we will argue later that the wind conditions at flight altitude might be quite different, this sensor is nonetheless an important reference input for the estimation algorithms especially during the launch of the system.

The dynamics is influenced by the line length which is measured by an multiturn rotary encoder attached to the drum of the towing rope winch.

For our prototype systems we also used two GPS receivers

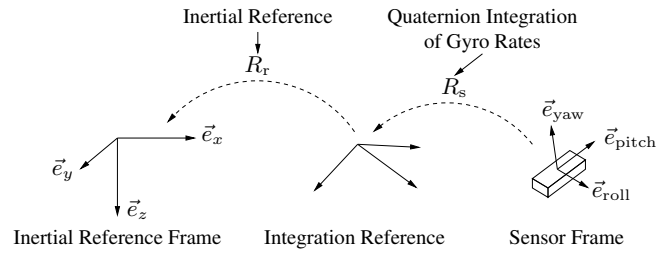


Fig. 3. The yaw estimator's state is represented by the two rotations R_s and R_r . Rotation R_s is computed by quaternion integration of the three turn rates and depicts the fast dynamics of the flying system. The rotation R_r references the system to the inertial frame thus capturing a potential slow drift due to sensor errors.

in the control pod and aboard the ship for calculating the difference between positions. Although we obtained valuable data for research and development purposes we discarded the use of GPS for operational flights as the position data suffer from dropouts due to rapid changes of antenna orientation during dynamic flights.

V. YAW ANGLE ESTIMATOR

In this section we discuss the yaw angle estimator (YAE) which determines ψ_g . This quantity can be considered as the attitude angle relative to the horizon. It is important to emphasize that we have no direct measurement of this angle, e.g. based on optical or microwave technology. Such a system would be expensive and difficult to operate at sea. Therefore we must estimate the orientation based on the rotation rates $\vec{\omega}_s$ and the acceleration values \vec{a}_s alone.

The orientation can be determined by integration of the turn rates but has to be referenced to the horizon. In order to obtain the direction of gravity we carry out the following steps: first the measured accelerations are transformed to the integration reference frame and then the obtained acceleration vector is averaged. This coordinate system represents, on average, an inertial frame: As we deal with a tethered system we expect that due to the averaging all dynamic accelerations cancel out over time, because otherwise the system 'would fly away' in contradiction to the constraint by the tether. Thus the average value represents an estimate for the gravity vector. However, extra precautions have to be taken in order to deal with sensor offsets, which may lead to a slow drift of the coordinate system.

The main idea of the algorithm is illustrated in Fig. 3: the turn rates $\vec{\omega}_s$ are integrated and then used to build an orientation matrix R_s , which represents the orientation of the IMU sensor with respect to the integration reference. This reference already features the property of an inertial system but drifts due to the measurement errors and sensor offsets with respect to the inertial reference frame. By evaluation of accelerations and extraction of the apparent gravity direction a rotation matrix R_r is computed by averaging. Thus R_r relates the, possibly drifting, integration system to the inertial reference frame. The final navigation angles are calculated on the basis $R = R_r \cdot R_s$ which represents the orientation of the IMU sensor with respect to the inertial reference system.

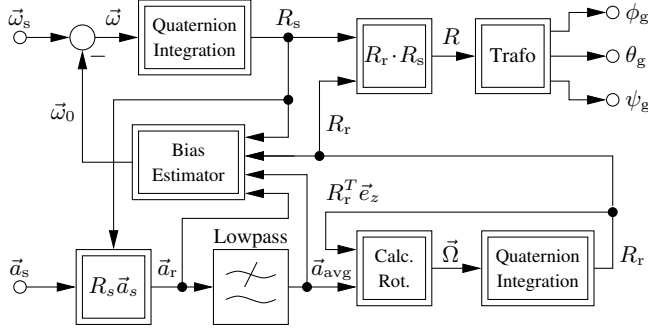


Fig. 4. Overview of the yaw angle estimator algorithm which computes the orientation estimate $\{\phi_g, \theta_g, \psi_g\}$ using turn rates $\vec{\omega}_s$ and accelerations \vec{a}_s .

We now will go into the details of the estimator algorithm. The respective components and their interactions are shown in Fig. 4. The first block to be mentioned is the integration of the turn rate vector $\vec{\omega} = \vec{\omega}_s - \vec{\omega}_0$ to R_s . The vector $\vec{\omega}_0$ denotes the sensor bias estimates. Details of the quaternion propagation can be found e.g. [19]. The rotation matrix R_s describes the fast dynamics of the systems.

This matrix R_s is now used to transform the measured accelerations \vec{a}_s into the integration reference by

$$\vec{a}_r = R_s \vec{a}_s. \quad (10)$$

This vector is subsequently filtered by a lowpass: $\vec{a}_{avg} = LP[\vec{a}_r]$, where $LP[]$ represents three independent filter operations on the vector components using a second order Butterworth lowpass with a cutoff frequency of 0.01 Hz. We would like to emphasize that (10) and the subsequent filter realize the central idea introduced in the beginning of this section: The component wise averaged value \vec{a}_{avg} thus represents an estimate for the gravity vector.

The next task is to compute the matrix R_r such that the normalized acceleration estimate $\vec{a}_0 = \vec{a}_{avg}/|\vec{a}_{avg}|$ is transformed into the $(-\vec{e}_z)$ -direction which is the measured acceleration signal of the gravitational force pointing into 'down'-direction:

$$R_r \vec{a}_0 = -\vec{e}_z. \quad (11)$$

The straight forward solution of the R_r propagation task could be to determine the drift of this vector between time steps $n \rightarrow (n+1)$ by $\Delta \vec{a} = \vec{a}_0(n+1) - \vec{a}_0(n)$ and propagate rotation R_r accordingly. Since numerical errors would accumulate a better solution is the introduction of a feedback loop and basing the propagation upon the deviation

$$\Delta \vec{a} = \vec{a}_0 - R_r^T \cdot (-\vec{e}_z). \quad (12)$$

We then have to find a rotation $\vec{\Omega}$ which compensates for this deviation by turning \vec{a}_0 accordingly:

$$\Delta \vec{a} = -\vec{\Omega} \times \vec{a}_0. \quad (13)$$

Inversion of this equation with respect to $\vec{\Omega}$ is not unique. If we require $\vec{\Omega} \perp \vec{a}_0$, which is equivalent to the minimal rotation fulfilling (13), some elementary vector algebra yields:

$$\vec{\Omega} = -\vec{a}_0 \times \Delta \vec{a}. \quad (14)$$

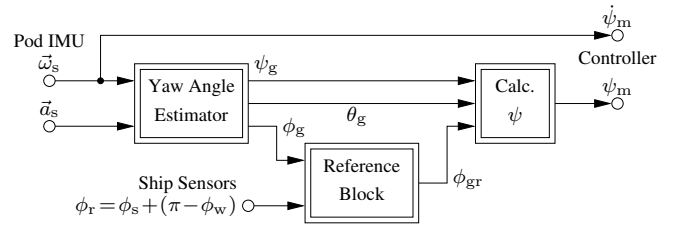


Fig. 5. Setup of the navigation algorithm which references the ϕ_g output of the YAE and computes the orientation ψ_m with respect to the wind for the flight controller.

This rotation $\vec{\Omega}$ is used for the propagation of R_r via quaternion algebra. The navigation angles $\{\phi_g, \theta_g, \psi_g\}$ are calculated out of $R = R_r \cdot R_s$ by using a similar sequence of rotations as given in (2).

For perfect sensors we would have finished the task at this point, but our gyroscopes bring along bias values on the turn rates which would accumulate to angle errors. In order to minimize these angle errors, it is crucial to compensate for these offsets before the quaternion integration (compare Fig. 4). Explaining the estimation of the offset vector $\vec{\omega}_0$ we start with the error due to dragging which is defined as:

$$\Delta \vec{a}_r = \vec{a}_r - \vec{a}_{avg}. \quad (15)$$

Now a rotation $\vec{\Omega}_r$ has to be found with the property

$$\Delta \vec{a}_r = \vec{\Omega}_r \times \vec{a}_r. \quad (16)$$

It is worth mentioning that this is a rather intuitive approach as during curve flights $|\vec{a}_r| \gg |\vec{a}_{avg}|$. Therefore the relation to a rotation seems to be weak. But as $\vec{\Omega}_r$ is strongly filtered the approach holds and worked in practice. An important issue of (16) is that \vec{a}_r is the reference for $\Delta \vec{a}_r$ and not \vec{a}_{avg} in order to get the smoother behavior. With the same type of reasoning as for (14) we get

$$\vec{\Omega}_r = \frac{1}{|\vec{a}_r|^2} (\vec{a}_r \times \Delta \vec{a}_r). \quad (17)$$

For estimation of the offset rate $\vec{\omega}_0$ the transformed rotation $R_s^T \vec{\Omega}_r$ is integrated with a gain of $\gamma = 0.003$ 1/s as follows:

$$\vec{\omega}_0(n+1) = \vec{\omega}_0(n) + \Delta t \gamma R_s^T \vec{\Omega}_r. \quad (18)$$

We are aware that this offset estimation algorithm is more an intuitive than mathematically rigorous approach. In our description we further omitted an additional damping term, regarding the rotational degree of freedom with respect to \vec{e}_z , which we have also implemented in order to further robustify the filtering scheme. Explaining these technical details cannot be done within the limits of this paper.

VI. WIND REFERENCING

As already introduced, the controller input value ψ_m represents the orientation with respect to the wind. The general arrangement for its computation is shown in Fig. 5. The IMU turn rates $\vec{\omega}_s$ and accelerations \vec{a}_s are fed into the yaw angle estimator (YAE) which determines estimates for the angles θ_g and ψ_g with respect to gravity. As this gravity reference

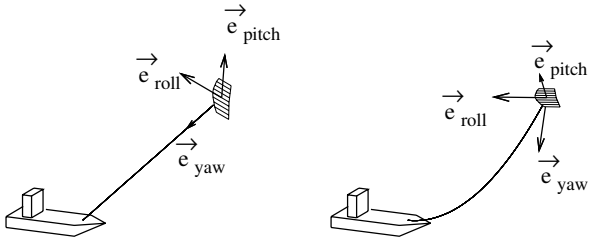


Fig. 6. Two different dynamic regimes have to be covered by the navigation algorithm. In the left figure the regular constrained flight situation is shown. The towing line force orients \vec{e}_{yaw} in line direction. For the free flight situation in the right figure the orientation of the kite has to be considered independently from the position.

is a single vector only, the ϕ_g output represents the dynamics with respect to the vertical \vec{e}_z -axis but lacks an absolute reference and therefore shows an offset drift due to sensor noise. As a consequence the 'reference block' is introduced in order to reference this value to $\phi_r = \phi_s + (\pi - \phi_w)$, which is the horizontal angle of the towing line to the wind direction, calculated from towpoint angle ϕ_s and wind direction ϕ_w aboard the ship. Without going into details this block performs a complementary filtering similar to [14] with highpass behavior with respect to ϕ_g combined with an offset generated by lowpass filtering of ϕ_r .

The computation of ψ_m is based on ϕ_{gr} , θ_g and ψ_g using (6). The specific choice of these input quantities is explained with Fig. 6 illustrating the different flight situations. In the left figure the regular mode of operation is shown which is determined by a constrained dynamics. The \vec{e}_{yaw} -vector points into direction of the towing line. Both the IMU and the towpoint measurement values coincide $\phi_{gr} \approx \phi_s + (\pi - \phi_w)$, $\theta_g \approx \theta_s$ and thus can be used likewise in order to compute ψ_m for the control task.

A more challenging situation is the free flight situation shown in the right part of Fig. 6. Sudden wind fall offs, downwashes or wave induced motion of the ship may lead to flight situations where the distance between tow point and kite is shorter than the towing line length. These situations lead to a completely different dynamics of the flying system which now more or less behaves like a paraglider. Although in most cases these situations fortunately last only a few seconds, arrangements must be made that the input values for the flight controller stay within appropriate range.

The computation of ψ_m from the data set $\{\phi_{gr}, \theta_g, \psi_g\}$ is based on control pod sensors apart from the referencing of ϕ_{gr} to ship sensors. As this referencing takes place on a long-term timescale the short-term timescale navigation can be considered as a local navigation aboard the control pod which is presumably the best orientation estimate of the kite with respect to the wind even in extreme flight situations.

We conclude this section by noting that in order to get high-performance navigation information the determination of the wind direction is a crucial challenge as wind direction at flight altitude might differ from the wind direction at sea level up to some tenths of degrees. Therefore a wind offset estimate is computed which provides an angle correction

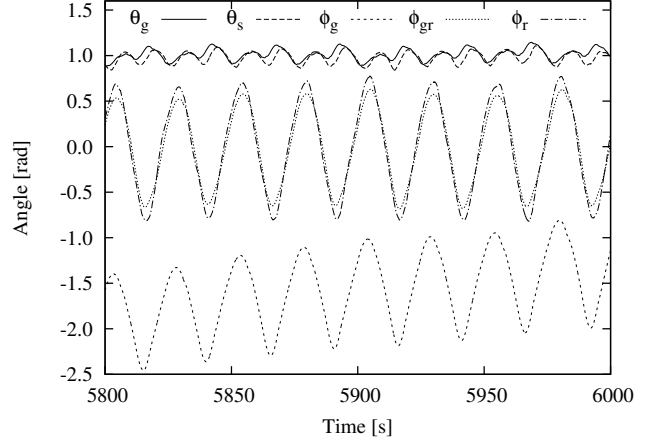


Fig. 7. Pattern-eight flight data comparing the YAE output ϕ_g to the reference value $\phi_r = \phi_s + (\pi - \phi_w)$ computed from ship sensors only. Except for a drifting offset these curves are in good agreement. This offset is compensated in the ϕ_{gr} value. For details see Section VI. Likewise the YAE output θ_g coincides with the towpoint value θ_s . Minor differences are due to additional excitation modes of the towing rope and control pod system not included in the simple tether assumption.

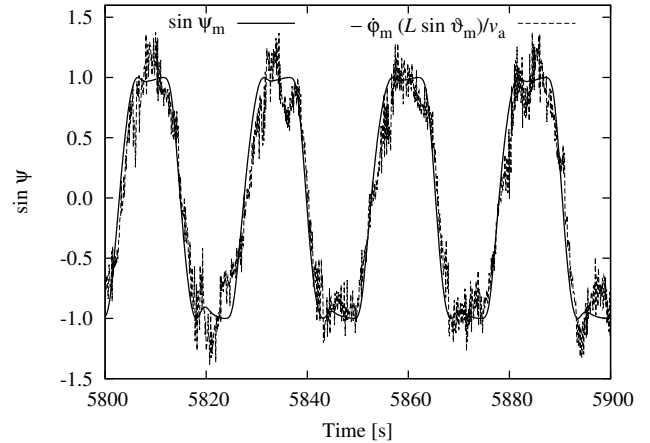


Fig. 8. Flight data comparing $\sin \psi_m$ to the corresponding term of the equation of motion (9). We give the unfiltered signal which is a bit noisy due to the derivation of φ_m with respect to time.

value to ϕ_r . Details of this algorithm go beyond the scope of this paper and will be published elsewhere.

VII. EXPERIMENTAL VALIDATION

In this section we compare the output angles of the YAE to other independent sensor measurements. As already discussed in Section VI during tethered dynamic flight the IMU and the towpoint measurement values coincide: ϕ_{gr} versus $\phi_s + (\pi - \phi_w)$ and θ_g versus θ_s . Corresponding measurement curves are shown in Fig. 7. The respective curves are in good agreement as explained in the figure caption. A verification of the ψ_m angle output of the navigation setup is shown in Fig. 8. The diagram plots the left-hand and right-hand sides of (9). The analogy of the curves shows the validity of this equation and thus the capability of the setup in providing a reasonable ψ_m value.

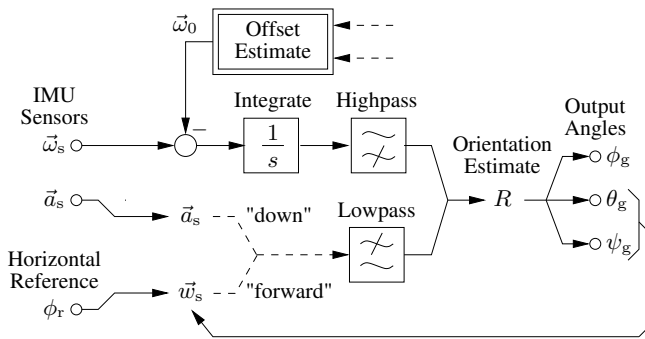


Fig. 9. Estimation algorithm combining YAE and wind referencing. The orientation input to the lowpass filter is based on the acceleration vector \vec{a}_s and the wind direction \vec{w}_s which is computed based on the horizontal reference ϕ_r and the estimator output angles θ_g and ψ_g . The function blocks are to represent the transfer dynamics rather than a concrete algorithm.

VIII. DISCUSSION AND FURTHER DEVELOPMENT

Although the presented YAE shows convincing results and has been used successfully for operational flights for years, the approach has certain shortcomings in estimating the offset rates $\vec{\omega}_0$. The fact that the referencing is with respect to the gravity axis only, may lead, in certain specific flight situations, to inaccurate offset estimations for sensor axes oriented to the \vec{e}_z -axis. This issue has also been reported for a Kalman filter approach on IMU data [11]. In order to overcome this situation we suggest a merging of YAE and wind referencing into one algorithm, thus processing direction of gravity and wind reference simultaneously.

An approach of a complementary filter processing accelerometer and magnetometer information is described in [14] which could be adopted in the following way: instead of the magnetometer we introduce a fictitious sensor value \vec{w}_s emulating a measurement of the horizontal reference wind direction. It is obtained by rotating the reference axis \vec{e}_x into the sensor frame by:

$$\vec{w}_s = R(\phi_r, \theta_g, \psi_g)^T \vec{e}_x. \quad (19)$$

Note that ψ_g and θ_g are filter outputs whereas ϕ_r is the horizontal referencing input. The principal data flow of such an algorithm is sketched in Fig. 9.

We would like to point out here that we consider the filtering of the acceleration input as critical: our YAE filter is based on the physics that the average of the acceleration vectors yields the direction of gravity for a tethered system. The common approach first calculates a direction out of the acceleration vector \vec{a}_s by normalization and uses $\vec{a}_s/|\vec{a}_s|$ as input value for the filter. This is also done for filter on the SO(3) described in [14]. As we have to deal with accelerations multiple of the gravity acceleration $|\vec{a}_s| \gg g$ during dynamical flight, it is questionable whether the error introduced by neglecting the $|\vec{a}_s|$ -amplitude is acceptable. Modifications and extensions to these filters may be necessary.

First validation tests of the combined algorithms with recorded flight data yielded estimated orientation angles

comparable to the output of the operational YAE and wind referencing setup. Meanwhile the estimated sensor biases seem to be more realistic than those of the YAE. However a critical sensitivity of the bias estimator on filter frequencies and flight situations has been observed, which necessitate further research and development efforts on these approaches.

Conclusively we would like to note that gains and filter frequencies of the presented algorithms were determined mainly by trial and error on recorded flight data sets. On the other hand a thorough Kalman filter design would be desirable and may result in some improvements of the estimator performance. However, such a Kalman theory based filter design for a tethered kite system turns out to be non-trivial as measurements are less corrupted by sensor noise than by disturbances (mainly wind), which can not be easily modeled within a statistically based framework.

REFERENCES

- [1] SkySails GmbH. [Online]. Available: <http://www.skysails.de>
- [2] "Book of abstracts," ISBN 978-94-6018-370-6, Airborne Windenergy Conference 2011. [Online]. Available: <http://www.awec2011.com>
- [3] L. Fagiano and M. Milanese, "Airborne wind energy: An overview," *American Control Conference (ACC)*, pp. 3132–3143, 2012.
- [4] B. Lansdorp, R. Ruitkamp, and W. Ockels, "Towards flight testing of remotely controlled surfkites for wind energy generation," *Proc. Modelling and Simulation Technologies Conf. AIAA-2007-6643*, Aug. 2007.
- [5] M. Canale, L. Fagiano, and M. Milanese, "High altitude wind energy generation using controlled power kites," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 2, pp. 279–293, 2010.
- [6] A. Ilzhöfer, B. Houska, and M. Diehl, "Nonlinear MPC of kites under varying wind conditions for a new class of large scale wind power generators," *Int. J. Robust Nonlinear Control*, vol. 17, no. 17, pp. 1590–1599, Nov. 2007.
- [7] P. Williams, B. Lansdorp, and W. Ockels, "Nonlinear control and estimation of a tethered kite in changing wind conditions," *AIAA Journal of Guidance, Control and Dynamics*, vol. 31, no. 3, pp. 793–799, 2008.
- [8] A. Furey and I. Harvey, "Evolution of neural networks for active control of tethered airfoils," *Proc. The 9th European Conference on Artificial Life*, pp. 746–756, 2007.
- [9] J. H. Baayen and W. J. Ockels, "Tracking control with adaption of kites," *IET Control Theory and Applications*, vol. 6, no. 2, pp. 182–191, 2012.
- [10] E. Foxlin, "Inertial head-tracker sensor fusion by a complementary separate-bias kalman filter," in *Proc. IEEE Virtual Reality Annual International Symposium (VRAIS'96)*, 1996, pp. 185–194.
- [11] H. Luinge and P. Veltink, "Measuring orientation of human body segments using miniature gyroscopes and accelerometers," *Med. Biol. Eng. Comput.*, vol. 43, pp. 273–282, 2005.
- [12] A. M. Sabatini, "Quaternion-based extended kalman filter for determining orientation by inertial and magnetic sensing," *IEEE Trans. Biomed. Eng.*, vol. 53, no. 7, pp. 1346–1356, 2006.
- [13] J. L. Crassidis, F. L. Markley, and Y. Cheng, "A survey of nonlinear attitude estimation methods," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 1, pp. 12–28, 2007.
- [14] R. Mahony, T. Hamel, and J.-M. Pflimlin, "Non-linear complementary filters on the special orthogonal group," *IEEE Trans. Autom. Control*, vol. 53, no. 5, pp. 1203–1217, 2008.
- [15] K. J. Jensen, "Generalized nonlinear complementary attitude filter," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 5, pp. 1588–1592, 2011.
- [16] M. Erhard and H. Strauch, "Control of towing kites for seagoing vessels," *IEEE Trans. Control Syst. Technol.*, 2012, DOI 10.1109/TCST.2012.2221093.
- [17] *Crista Sensor Head Interface Control Document*, Cloud Cap Technology Inc., May 2004.
- [18] *ADIS16350/ADIS16355 Datasheet, Rev. A*, Analog Devices Inc.
- [19] J. B. Kuipers, *Quaternions and Rotation Sequences*. Princeton University Press, 2002.