



ON-LINE INERTIA MEASUREMENT OF UNMANNED AERIAL VEHICLES USING ON-BOARD SENSORS AND BIFILAR PENDULUM

Matija Krznar^{1,*}, Denis Kotarski², Petar Piljek³ and Danijel Pavković³

¹Sv. Križ Začretje, Croatia

²Karlovac University of Applied Sciences
Karlovac, Croatia

³University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture
Zagreb, Croatia

DOI: 10.7906/indecs.16.1.12
Regular article

Received: 13 December 2017.
Accepted: 26 February 2018.

ABSTRACT

Identification of a dynamical model and its parameters is one of the fundamental problems in the field of robotics and system dynamics modelling. For the general situation of an object motion with six degrees of freedom (6-DOF), such as in the case of the Unmanned Aerial Vehicle (UAV), the key physical parameters are vehicle mass and moment of inertia. Even though UAV mass and its geometry/topology are easily obtainable, it is difficult to identify the inertia tensor considering that it is not measurable by static tests. This article presents a simple and effective method for on-line estimation of a rigid-body inertia based on a two-wire pendulum and an on-board integrated sensor system. Herein, the test subject (i.e. UAV) is suspended by two thin parallel wires in such a way to form a bifilar torsional pendulum about the vertical axis. Using on-board sensors from the UAV flight controller (FC) unit, the pendulum oscillations are recorded and processed in order to obtain trend-free and noise-free signals used in the final inertia estimation phase. The proposed identification algorithm is verified experimentally for two typical cases of suspended objects related to the UAV control box and a full UAV configuration.

KEYWORDS

UAV, on-line moment of inertia identification, bifilar pendulum, IMU sensors

CLASSIFICATION

ACM: D.1.1.

JEL: Z00

*Corresponding author, *η*: matija.krznar@gmail.com; +385 91 2282200;
Aleja L. Vukelića 4, Sv. Križ Začretje, HR – 49 223, Croatia

INTRODUCTION

For the purpose of UAV control systems development, knowledge about mathematical model of the system is required. In order to attain the requisite precision of such system model, all important parameters are needed to be known with sufficient accuracy [1, 2]. Since moment of inertia cannot be estimated through static tests, its identification could pose certain difficulties. For geometrically simple objects, inertia can be derived analytically or obtained from a model in CAD software. For more complex bodies such as the UAVs, equations tend to get more complex, which makes the calculation of inertia quite impractical, so it would be more convenient to estimate it from dynamical tests. Various approaches in [3-7] introduce identification of inertia based on measurements of the oscillation period of a bifilar pendulum. Although all these methods enable very precise results, they are based on offline analysis of measured data.

This article proposes the implementation of an on-line inertia estimation procedure using Kalman filter techniques and angle estimate from the inertial measurement unit (IMU) integrated inside the UAV flight controller (FC). In order to eliminate unwanted signal drift and noise from measurements, a recursive differentiating filter is used, cascaded to the low-pass filter implemented in the form of a scalar Kalman filter. Such processed signal is used to determine the period of a pendulum-suspended body oscillations, through detection and measurement of angle rise condition. As a final result, estimation result is based on analytical relationships of a pendulum oscillation period and inertia.

For the purpose of experimental identification of the UAV inertia, test setup is developed and assembled from two vertical beams forming the base of the pendulum, one horizontal beam and the test body (UAV) suspended via a two thin wires. When the test body is displaced from the initial condition and set free, a restoring torque twists it back toward the equilibrium position, and periodic motion is initiated, characterized by specific oscillation frequency.

The article is organized as follows. At first, a pendulum dynamics model considering the test body is introduced. Following by the design of an inertia estimation algorithm that includes the body deflection angle measurement, sensor drift compensation (detrending), a low-pass filtering to remove the sensor noise, and a pendulum oscillation period detection. Final section gives results of experimental verification for the proposed moment of inertia identification procedure and the concluding remarks.

PENDULUM MODEL

Dynamic modelling based on equations of motion is used to relate the measurable parameters of the pendulum's configuration to the inertia of the object suspended from the pendulum (see e.g. [3-7]).

For modelling purposes, some non-dominant physical factors are neglected as a trade-off between simplicity and estimation accuracy. The following assumptions are introduced: pendulum is centred around UAV Center of Gravity (CoG), wires are ideally parallel, and angular displacement of the wires from the vertical axis is negligible, wires are considered stiff and with negligible mass, displacement angle is kept small, there are no unbalanced vertical forces and rotation of pendulum is considered only about the vertical axis.

Figure 1a shows the following defined geometric constrains: distance between suspension wires, $d = 2r$; length of wires, h ; twist angle around vertical axis, α ; tilt angle due rotation, β .

From geometry it is possible to derive a kinematics relation by means of a simple trigonometric relationship. Further simplification is introduced by constraining oscillation

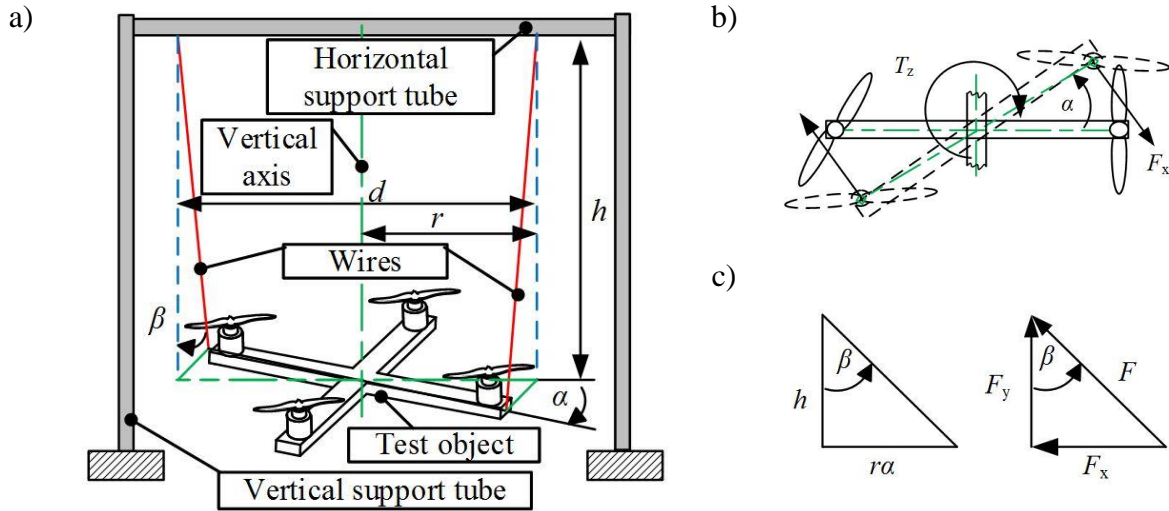


Figure 1. Bifilar pendulum: a) principal schematic, b) UAV deflections from normal and c) deflection angle representation related to body displacement and acting forces.

amplitude to small angles wherein $\sin \beta \approx \beta$. Thus, angle of twist α can be related to rotation angle β via distance $r\alpha$ obtained from arc length formula:

$$\sin \beta = r\alpha / h, \quad \beta = r\alpha / h. \quad (1)$$

Tension force in wires is derived from object weight. At stationary conditions (equilibrium), net torque T_z (Fig. 1) about vertical axis is zero and each wire holds half of object's weight.

$$F = \frac{1}{2}mg, \quad F_y = \frac{1}{2}mg \sin \beta, \quad F \approx F_y. \quad (2)$$

When the object is rotated around vertical axis by a small angle, a horizontal reaction occurs due to the tension in the wires, given by the following relationships:

$$F_x = \tan \beta F_y, \quad F_x = \beta F_y. \quad (3)$$

From Figure 1b, reaction torque is induced by horizontal component of the tension. Since each string produces equal forces, and thus total torque equals $2F_x r$, where:

$$F_x = \frac{1}{2}mg \frac{r}{h} \alpha, \quad T_z = \frac{mg r^2}{h} \alpha. \quad (4)$$

Using Newton-Euler formulation, with $T_z = I\ddot{\alpha}$, equations of motion are derived [6, 7].

$$I\ddot{\alpha} + \frac{mgd^2}{4h} \alpha = 0. \quad (5)$$

By solving this equation analytically, the solution for undamped oscillation case can be found in the standard oscillatory form:

$$\alpha(t) = c_1 \sin(kt) + c_2 \cos(kt), \quad k = \sqrt{\frac{mgd^2}{4hl}}. \quad (6)$$

That oscillatory motion is characterized by the following period and frequency:

$$T = \frac{2\pi}{\omega} = \frac{2\pi}{k}, \quad f = \frac{1}{2\pi} k. \quad (7)$$

INERTIA ESTIMATION

The inertia estimation algorithm is implemented by using a Matlab Embedded Coder™, wherein estimator components are written as functions and are verified by computer simulations utilising Matlab/Simulink environment. By using Pixhawk Development Support, appropriate firmware for the flight controller (FC) is generated and uploaded. The angle information is obtained by attitude estimator embedded within the UAV [8, 9].

Figure 2 shows the individual components of the inertia estimation algorithm.

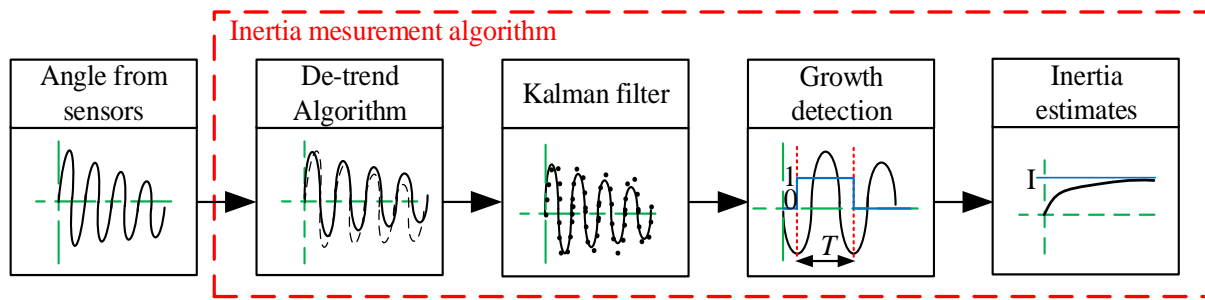


Figure 2. Inertia estimator algorithm schematic representation.

MEASURING ANGLE OF OSCILLATION

By utilising the integrated inertial MEMS sensors connected to the microcomputer, it is possible to determine vehicle attitude in the form of corresponding Euler angles. Moreover, by using the Estimation and Control Library (ECL) from Pixhawk PX4 flight controller, the angle is obtained by sensor fusion of magnetometer, gyroscope and accelerometer [8, 9]. Thus, obtained data is fed to the external computer through serial port by using External mode option for running the UAV model [9]. For that purpose, a custom firmware containing estimator application is generated and uploaded by using Matlab/Simulink Support library for Pixhawk Development [9].

INERTIAL SENSOR DRIFT COMPENSATION

As discussed above, the oscillation angle is obtained from the IMU sensors integrated within the flight controller unit. These sensors provide fast and precise response, but may suffer from accumulated error, i.e. they may tend to ‘drift’. Due to aforementioned integrator properties of sensor dynamics, a constant error in acceleration measurement results in a linear (ramp) error in velocity and a quadratic (parabolic) error growth in terms of the UAV position. In order to remove (filter out) these drift effects, and, thus, to increase the robustness of inertia estimation, a recursive drift removal filter is introduced [10]:

$$y(k) = (1 - a_f)[x(k) - x(k - 1)] + a_f y(k - 1), \quad (8)$$

where y is the filter output, x is its input and a_f is the filter tuning parameter typically chosen between 0,9 and 1. The corresponding discrete-time (z -domain) transfer function of the detrending filter is given as:

$$H(z) = (1 - a_f) \frac{1 - z^{-1}}{1 - a_f z^{-1}}. \quad (9)$$

Figure 3a shows the pseudo code of the proposed detrending filter. Its effectiveness in terms of drift removal for typical IMU sensor measurements is illustrated by the results in Figure 3b. In particular, the deflection angle measurement linear trend superimposed to the actual pendulum oscillations is effectively removed, thus revealing the actual pendulum oscillatory movement in terms of displacement angle.

DISCRETE-TIME SCALAR KALMAN FILTER

To filter out the measurement noise, a discrete-time scalar type Kalman filter (KF) is implemented. For the case of single-dimensional signals, such as the angle measurement, scalar Kalman filter may be convenient since it provides an optimal estimate of the state variable under assumption that measurement noise variance and state variable perturbations are known in advance [11]. The scalar KF is the special case of the more general matrix wherein,

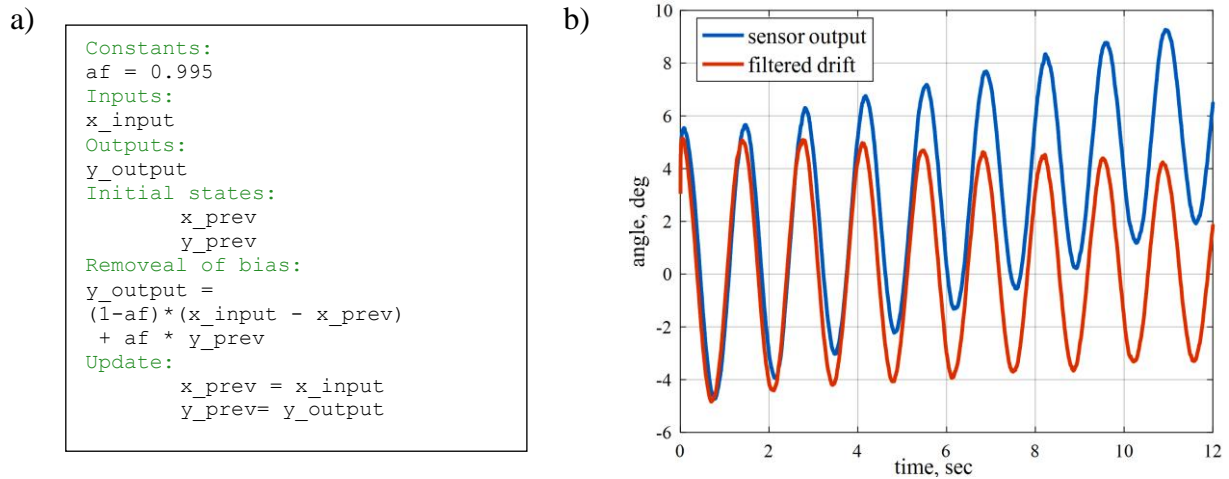


Figure 3. a) Pseudo code of offset removal function, b) offset removal function response.

wherein the model parameters and measurements are scalar-valued [12, 13]. The filter operates by interchanging two characteristic phases: “predict” and “update”. The predict phase uses the state estimate from the previous time step to produce an estimate of the state at the current time step (a-priori estimate). In the update phase, the current a-priori prediction is combined with current observation in order to refine the state estimate. This corrected estimate is termed a-posterior state estimate [11, 14].

The internal state of the scalar Kalman filter is modelled by a random-walk process model [11]:

$$x_k = x_{k-1} + v_{k-1}, \quad (10)$$

which includes a Gaussian perturbation (noise) v , characterized by perturbation variance Q . The measurement noise is assumed to be Gaussian, and its variance R is derived from sensor technical specifications. The final scalar Kalman filter implementation is given by the following set of equations.

Predict phase:

$$x_k = x_{k-1}, P_k = P_{k-1} + Q. \quad (11)$$

Update phase:

$$K = \frac{P_{k-1}}{P_{k-1} + R}, x_{k-1} = x_k + K(u - x_k), P_{k-1} = (1 - K)P_k. \quad (12)$$

Since the measurement noise variance R is assumed to be known in advance, the state perturbation variance Q represents the scalar Kalman filter tuning parameter. Its choice is usually a trade-off between parameter variations tracking ability and attenuation of noise.

Scalar Kalman filter implementation pseudo code is shown in Figure 4a. The filter response to an oscillatory signal with noise characterized by variance $R = 1$ is shown in Figure 4b, for the case of the Kalman filter tuning parameter $Q = 10^{-3}$. The responses in Figure 4b indicate that the random-walk model-based scalar Kalman filter may be characterized by a certain response delay, but the general oscillatory behaviour is captured well.

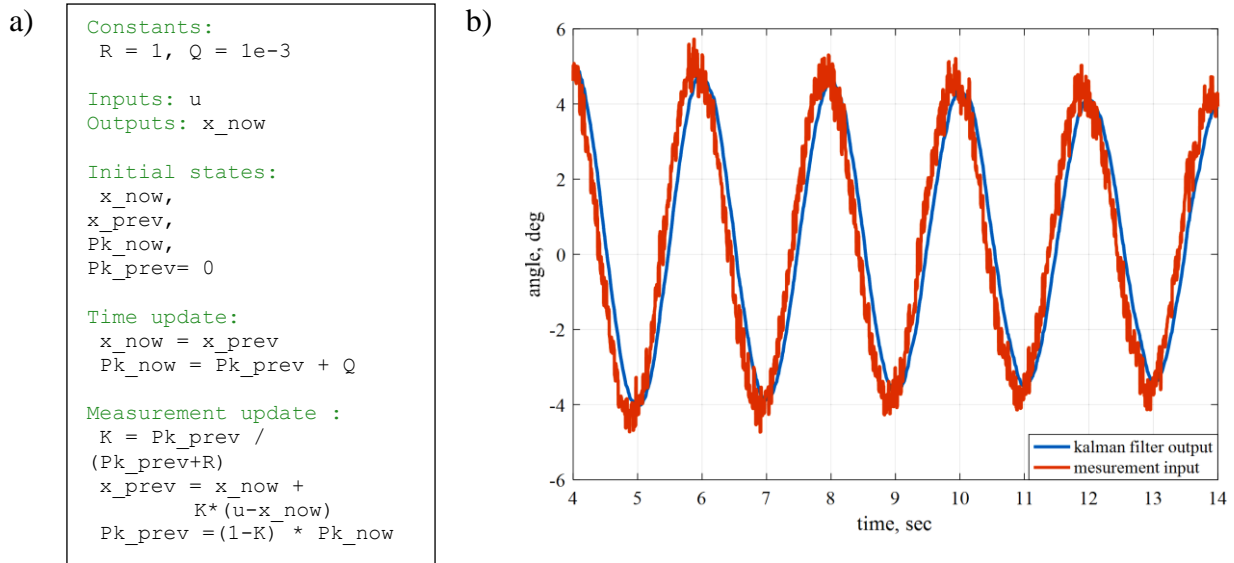


Figure 4. a) Function pseudo code, b) filter response.

OSCILLATION PERIOD ESTIMATION

Inertia estimation algorithm is based on the detection of oscillation rise and fall state with respect to time, and related oscillation period estimation. Angle evolution in time-domain can be separated into following stages: rise, fall, local maximum and local minimum. To detect if signal is in the state of increase (rise) or decrease (fall), a simple test algorithm is proposed. Input takes n samples from the inertial sensors and stores them in an array. When rise condition is detected a function initiates the counter that counts a number of samples until angle decrease condition is detected, wherein the counting is halted and the resulting number of counts is proportional to the oscillation period. By multiplying the counter final value and sample rate algorithm, the duration of rise condition is computed. If the last point is equal or higher than mean of first and second point, the angle has been in the state of growth, and the algorithm output is 1, otherwise the output is 0.

The proposed algorithm is given by pseudo code in Figure 5a, while the illustration of signal growth detection vs. oscillation plot is shown in Figure 5b. In this particular case, the angle growth detection initiates a counter when the input signal hits a drop point. The value of the counter multiplied by the sampling time provides an estimate of the oscillation period. Inertia is obtained based on the initial system parameters of the pendulum and the detected period of the oscillation. In order to obtain smooth (noise-free) estimation, the estimator output (inertia estimate) is further filtered by means of a narrow-bandwidth low-pass filter.

INERTIA ESTIMATOR

From previously derived pendulum equations (4)-(6), expression for the moment of inertia in terms of oscillation period (or frequency) [3-8] can be obtained as:

$$I = \frac{mgd^2}{4h} \frac{T^2}{4\pi^2} = \frac{mgd^2 T^2}{16h\pi^2} = \frac{mgd^2}{16h\pi^2 f^2}. \quad (13)$$

A two second delay is introduced before each inertia value is recorded, in order to eliminate initial oscillations in the estimate, that may otherwise lead to error in estimation. For every calculated inertia, data is averaged and final inertia estimation is acquired. Figure 8 show pseudo-code implementation and the results of period detection and average inertia calculation.

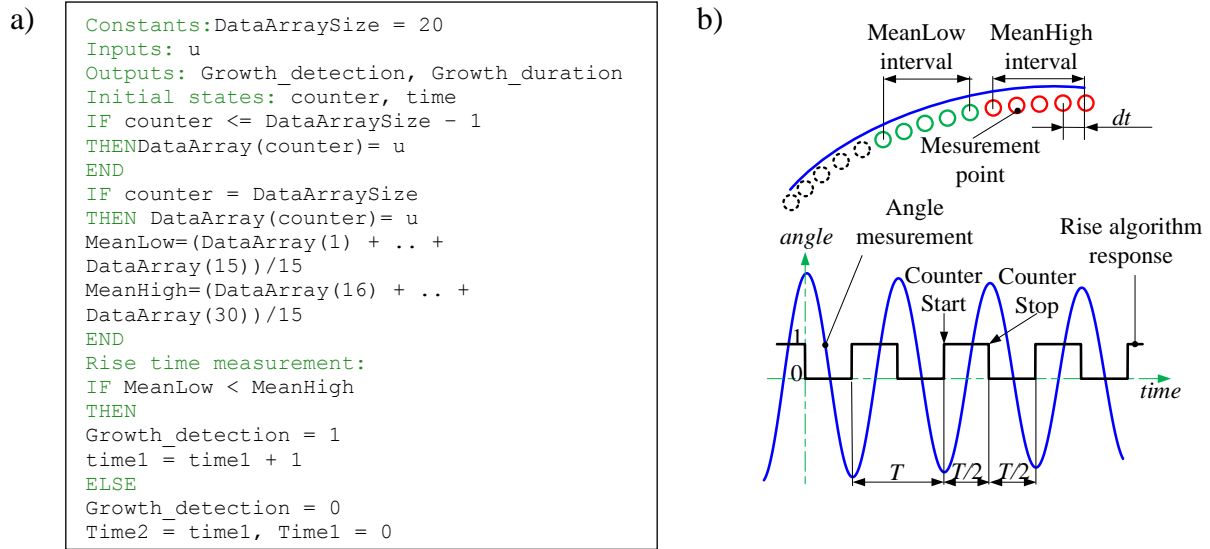


Figure 5. a) Algorithm pseudo code, b) response and period detection.

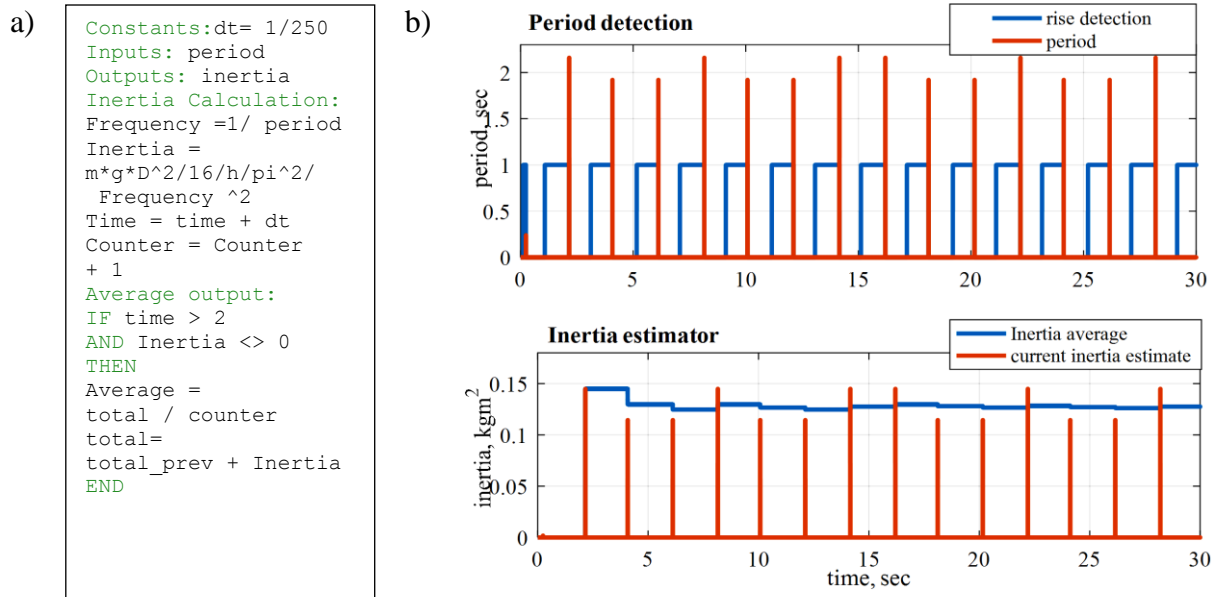


Figure 6. a) Algorithm pseudo code, b) response of Kalman filter, period detection and inertia estimation.

MEASUREMENTS AND RESULTS

For the validation of the proposed inertia estimation method, a moment of inertia is calculated analytically. Measurements are taken for two test case scenarios including the control unit box (Fig. 7a), and the overall UAV system (Fig. 7b).

In the first test case, the box, inertia depends on width a , length b , height c , and the mass m with respect to the perpendicular axis to the rod, which passes through the middle of the rod (Fig. 7a) [15]. The flight computer and the box are centred with respect to the rotational axis, and, hence, the total inertia of the system about this rotational axis is calculated as:

$$I_{\text{box}} + I_{\text{flight computer}} = \frac{1}{12} m_b (a_b^2 + b_b^2) + \frac{1}{12} m_{fc} (a_{fc}^2 + b_{fc}^2). \quad (14)$$

Figure 8 shows the principal schematics of the experimental setup for the case of control unit box and the UAV suspended by the pendulum test wires. In the former case (Fig. 8a), the suspension point is at the edge of the box, while in the latter case (Fig. 8b), the overall UAV is suspended at the axles of two opposing quad rotor propellers.

The quad rotor inertia model consists of the geometric bodies such as the cross-beam, boxes and the cylindrical elements corresponding to motors and propellers (propulsion). Since the quad rotor UAV mechanical structure shows a high degree of symmetry, body inertia tensor can be reduced to a diagonal matrix [16]. Further approximation is to define the following parts: one cross-beam structure, modelled as two solid narrow cylinders fastened in the middle forming a cross, one electronics box and one FC, modelled as a rectangular parallelepiped, four motors, modelled as solid cylinders, four propellers, modelled as the flat cylinders (i.e. inertia of the rotating blade). The inertia of individual elements is listed in Table 1.

The total UAV inertia (Fig. 8b) is obtained by combining the inertia contributions of each body: frame inertia, electronics, flight computer, motors and propellers. Using parallel axis theorem and equations from literature [16], the following final set of equations is obtained for inertia values:

$$I_{xx} = \sum I_{\text{body}, x}, I_{yy} = \sum I_{\text{body}, y}, I_{zz} = \sum I_{\text{body}, z} \quad (15)$$

The setup is shown in Figure 9, wherein additional stability of the frame is assured by tripod elements at the ground level. Inertia is measured about vertical (yaw) axis. Undertaken measurements and the final results are shown in Table 2. The procedure is similar for other axes of motion, i.e. the body is rotated according to referenced axis of measurement. Figures 10 and 11 show the real-time inertia estimator plot for test cases a) and b). In first test case a logged signals stored on SD card are shown, and for the second test case, a real-time scope output using external mode is shown. Such live monitoring occasionally suffers from signal loss due to real-time execution of code. Because of limited memory external mode model is run as background task, but execution of code is not halted [9].

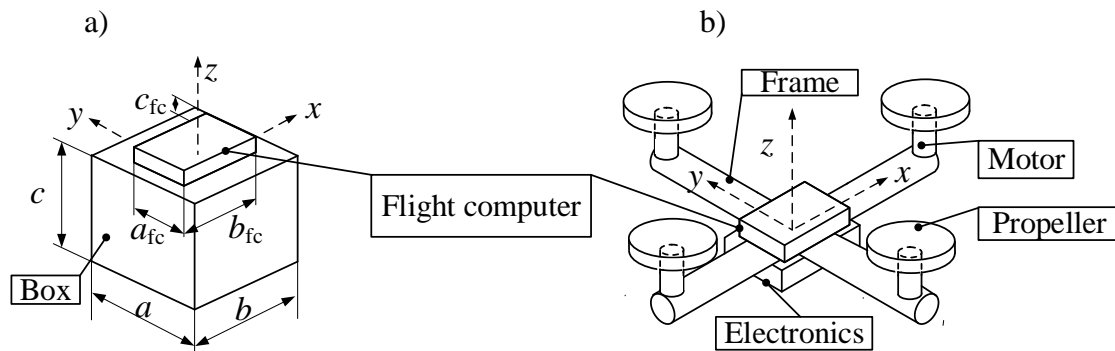


Figure 7. Inertia calculation for: a) box and b) UAV.

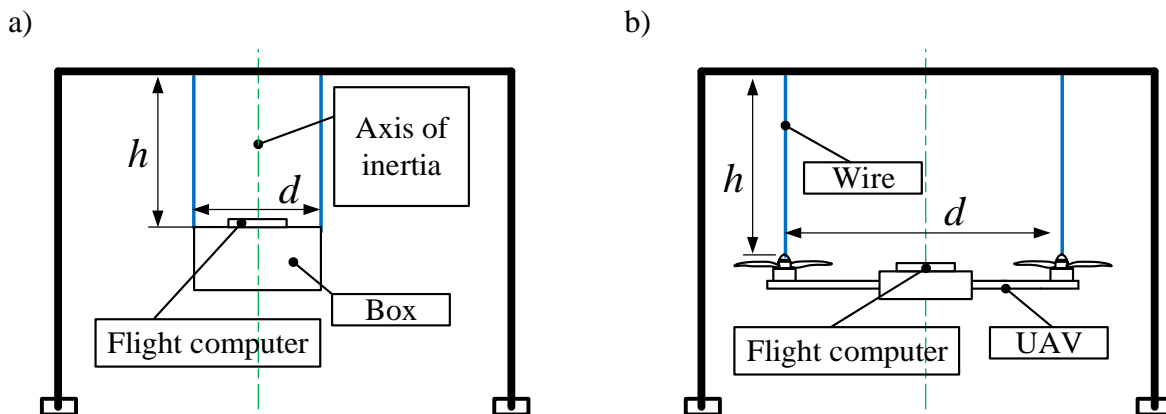


Figure 8. Test cases for inertia measurement: a) plate/box, b) UAV.

Table 1. Inertia calculation for test cases a) and b).

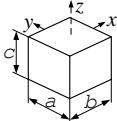
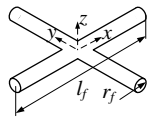
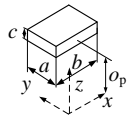
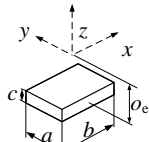
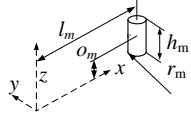
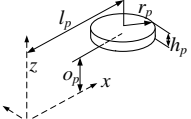
Part name	Formula	Geometry	Dimensions	Calculated inertia, kg·m ²
Box	$I_x = \frac{m_b(b^2+c^2)}{12}$ $I_y = \frac{m_b(c^2+a^2)}{12}$ $I_z = \frac{m_b(a^2+b^2)}{12}$		$m_b = 376 \text{ g}$ $a = 50 \text{ mm}$ $b = 50 \text{ mm}$ $c = 50 \text{ mm}$	$I_x = 1.5667 \cdot 10^{-4}$ $I_y = 1.5667 \cdot 10^{-4}$ $I_z = 1.5667 \cdot 10^{-4}$
Frame	$I_x = I_y = m_f \left(\frac{r_f^2}{4} + \frac{l_f^2}{12} \right) + \frac{1}{2} m_f r_f^2$ $I_z = m_f \left(\frac{r_f^2}{4} + \frac{l_f^2}{12} \right) + m_f \left(\frac{r_f^2}{4} + \frac{l_f^2}{12} \right)$		$m_f = 454 \text{ g}$ $r_f = 14 \text{ mm}$ $l_f = 520 \text{ mm}$	$I_x = 0.0102968$ $I_y = 0.0102969$ $I_z = 0.0205047$
Flight computer	$I_{fcx} = m_{fc} \left(\frac{a^2}{12} + \frac{c^2}{12} + o_p \right)$ $I_{fcy} = m_{fc} \left(\frac{b^2}{12} + \frac{c^2}{12} + o_p \right)$ $I_{fcz} = m_{fc} \left(\frac{b^2}{12} + \frac{a^2}{12} \right)$		$m_{fc} = 38 \text{ g}$ $a = 55 \text{ mm}$ $b = 80 \text{ mm}$ $c = 18 \text{ mm}$ $o_p = 45 \text{ mm}$	$I_x = 8.755 \cdot 10^{-5}$ $I_y = 1.008 \cdot 10^{-4}$ $I_z = 3.245 \cdot 10^{-5}$
Electronics	$I_{ex} = m_e \left(\frac{a^2}{12} + \frac{c^2}{12} + o_e \right)$ $I_{ey} = m_e \left(\frac{b^2}{12} + \frac{c^2}{12} + o_e \right)$ $I_{ez} = m_e \left(\frac{b^2}{12} + \frac{a^2}{12} \right)$		$m_e = 380 \text{ g}$ $a = 200 \text{ mm}$ $b = 120 \text{ mm}$ $c = 35 \text{ mm}$ $o_e = 45 \text{ mm}$	$I_x = 0.00804685 \text{ kgm}^2$ $I_y = 0.00839545 \text{ kgm}^2$ $I_z = 8.5075 \cdot 10^{-4} \text{ kgm}^2$
Motors	$I_{mx} = m_m \left(\frac{r_m^2}{4} + \frac{l_m^2}{12} + o_m \right),$ $I_{my} = m_m \left(\frac{r_m^2}{4} + \frac{h_m^2}{12} + l_m + o_m \right)$ $I_{mz} = m_m \left(\frac{r_m^2}{2} + l_m^2 \right)$		$m_m = 55 \text{ g}$ $r_m = 14 \text{ mm}$ $l_m = 241 \text{ mm}$ $h_m = 39.7 \text{ mm}$ $o_m = 45 \text{ mm}$	$I_x = 1.2129 \cdot 10^{-4} \text{ kgm}^2$ $I_y = 0.00331575 \text{ kgm}^2$ $I_z = 0.00319984 \text{ kgm}^2$
Propellers	$I_{px} = m_p \left(\frac{r_p^2}{6} + \frac{h_p^2}{12} + o_p \right)$ $I_{py} = m_p \left(\frac{r_p^2}{6} + \frac{h_p^2}{12} + l_p + o_p \right)$ $I_{pz} = m_p \left(\frac{r_p^2}{3} + l_p^2 \right)$		$m_p = 2 \text{ g}$ $r_p = 127 \text{ mm}$ $l_p = 241 \text{ mm}$ $h_p = 4 \text{ mm}$ $o_p = 50 \text{ mm}$	$I_x = 1.1428 \cdot 10^{-5} \text{ kgm}^2$ $I_y = 1.2758 \cdot 10^{-4} \text{ kgm}^2$ $I_z = 1.2691 \cdot 10^{-4} \text{ kgm}^2$

Table 2. Results of measurements.

Test case number and subject	Estimated inertia, kg·m ²	Calculated inertia, kg·m ²	Difference, estimated vs. calculated, %
1. Box, z-axis	1,5407 10 ⁻⁴	1,5667 10 ⁻⁴	1,66
2. Box, z-axis	1,4734 10 ⁻⁴	1,5667 10 ⁻⁴	5,96
3. Box, z-axis	1,5584 10 ⁻⁴	1,5667 10 ⁻⁴	0,53
Average:	1,5242 10 ⁻⁴	1,5667 10 ⁻⁴	2,70
1. UAV, z-axis	0,0216	0,0316	31,65
2 UAV, z-axis	0,0285	0,0316	9,81
3. UAV, z-axis	0,0273	0,0316	13,61
4. UAV, z-axis	0,0281	0,0316	11,08
5. UAV, z-axis	0,0287	0,0316	9,18
Average:	0,0269	0,0316	14,87

The inertia estimates based on the pendulum period estimation show good consistency and low sensitivity to measurement noise, i.e. smooth steady-state estimates of moment of inertia are obtained by the proposed identification method.



Figure 9. Photograph of the UAV inertia measurement setup.

Box inertia about the particular axis of rotation is verified by means of a single experiment, due to high confidence in the relatively simple analytical expressions for box inertia (Table 1). On the other hand, for more complex body shapes, repeated tests would be worthwhile in order to establish the consistency of the experimentally obtained results, and, thus, to confirm the results of rather complex analytical calculations.

CONCLUSIONS

This article shows the hardware implementation of a simple and effective method for on-line estimation of a solid-body moment of inertia, with particular application to the UAVs. Using inexpensive (and readily available) MEMS sensors and interface with computer, this method is portable to many hardware platforms. Moreover, by using automated on-line inertia estimation procedure, the effect of human error is excluded from the results.

Experimental results were presented for a two test cases, a homogeneous box and a 500 mm wide experimental UAV. Analysis of calculated and measured inertia was given for the purpose of verification of results. Developed method of inertia estimation has been experimentally verified, and it has been characterized by simplicity of implementation, adequate accuracy, high repeatability and low costs in terms of the hardware requirements. User interaction is limited only in the starting phase of measurement by imposing initial conditions (twist angle of 5° - 10°).

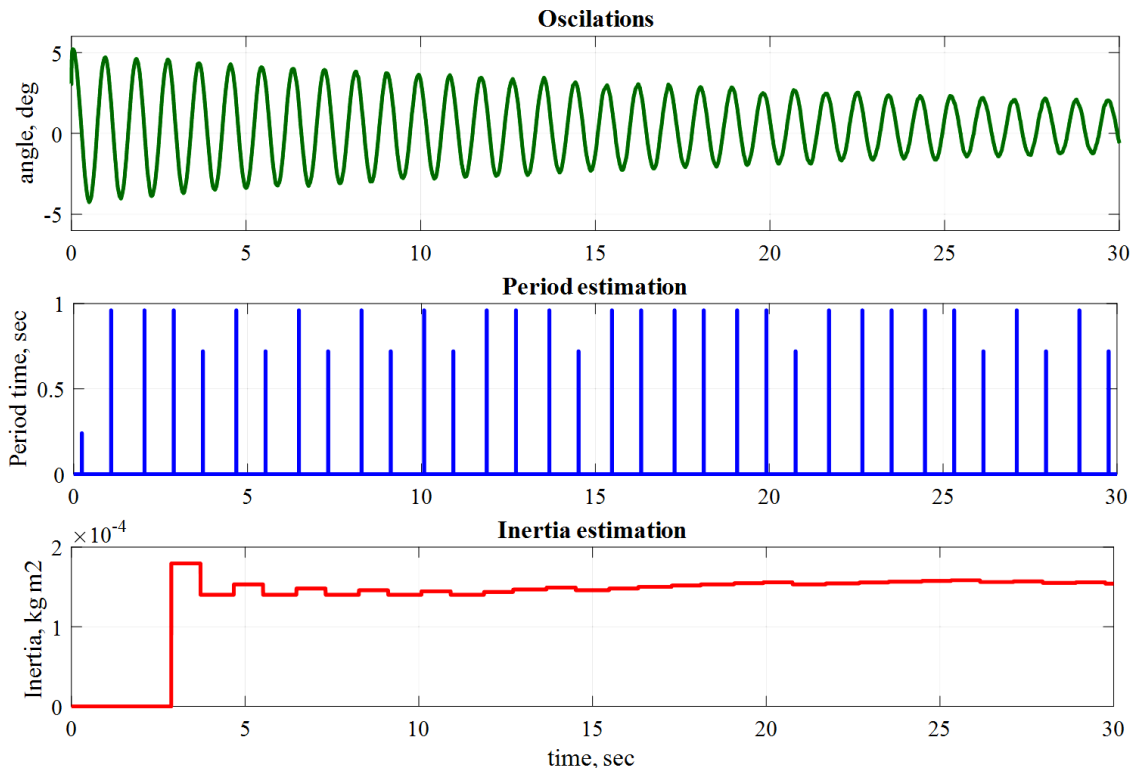


Figure 10. First test case, box – oscillations of body, period and inertia estimator response.

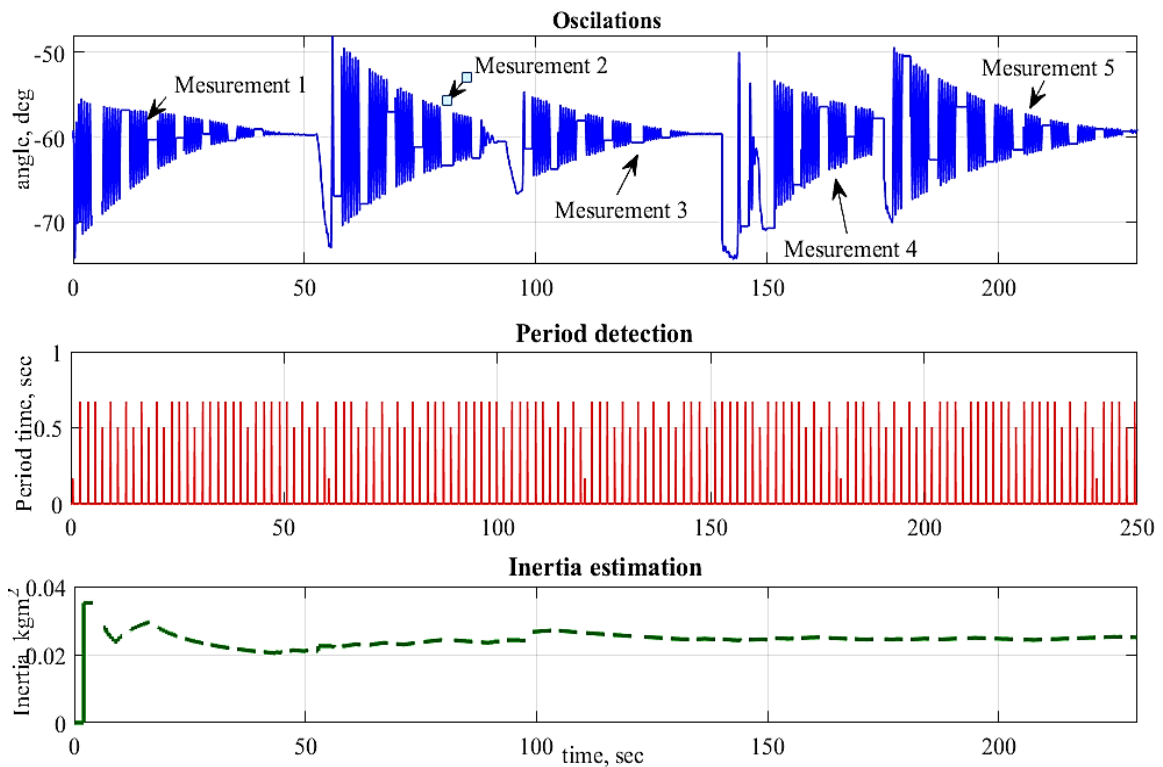


Figure 11. Second test case, UAV – oscillations of body, period and inertia estimator response.

APPENDIX

Table 3. Abbreviations used in the text.

Notion	Abbr.	Notion	Abbr.
angular acceleration	$\ddot{\alpha}$	micro-electro-mechanical system	MEMS
box geometric measures	a, b, c	moment of inertia	I
box mass	m_b	moment of inertia of box	I_{box}
centre of gravity	CoG	moment of inertia of flight computer	I_{fc}
distance between suspension wires	$d=2r$	net torque	T_z
estimation and control library	ECL	output	y
filter tuning parameter	a_f	pendulum geometric constant	k
flight computer mass	m_{fc}	period	T
flight controller (computer)	FC	perturbation (gaussian noise)	v
force in x or y direction	F_x, F_y	six degrees of freedom	6-DOF
frequency	f	state perturbation variance	Q
inertial measurements unit	IMU	tilt angle due rotation	β
input	x	twist angle around vertical axis	α
length of wires	h	unmanned aerial vehicle	UAV
measurement noise variance	R	UAV principal moments of inertia	I_{xx}, I_{yy}, I_{zz}

REFERENCES

- [1] Kotarski, D.; BeniĆ, Z. and Krznar, M.: *Control Design for Unmanned Aerial Vehicles with Four Rotors*. Interdisciplinary Description of Complex Systems **14**(2), 236-245, 2016, <http://dx.doi.org/10.7906/indecs.14.2.12>,
- [2] BeniĆ, Z.; Piljek, P., and Kotarski, D.: *Mathematical modelling of unmanned aerial vehicles with four rotors*. Interdisciplinary Description of Complex Systems **14**(1), 88-100, 2016, <http://dx.doi.org/10.7906/indecs.14.1.9>,
- [3] Schedlinski, C. and Link, M.: *A survey of current inertia parameter identification methods*. Mechanical Systems and Signal Processing **15**(1), 189-211, 2001, <http://dx.doi.org/10.1006/mssp.2000.1345>,
- [4] Mustapa, Z.; Saat, S.; Husin, S.H. and Abas, N.: *Altitude controller design for multi-copter UAV*. Proceedings of 2014 International Conference on Computer, Communications, and Control Technology (I4CT). Kedah, pp.382-387, 2014, <http://dx.doi.org/10.1109/I4CT.2014.6914210>,
- [5] McClelland, W.A.: *Inertia measurement and dynamic stability analysis of a radio-controlled joined-wing aircraft*. Air Force Institute of Technology, Ohio, 2016,
- [6] Jardin, M. and Mueller, E.R.: *Optimized measurements of unmanned-air-vehicle mass moment of inertia with a bifilar pendulum*. Journal of Aircraft **46**(3), 763-775, 2009, <http://dx.doi.org/10.2514/1.34015>,
- [7] Bauchau, O.A.; Rodriguez, J. and Chen, S.-Y.: *Modeling the Bifilar Pendulum Using Nonlinear, Flexible Multibody Dynamics*. Journal of the American Helicopter Society **48**(1), 53-62, 2003, <http://dx.doi.org/10.4050/JAHS.48.5>,
- [8] Feng, L. and Fangchao, Q.: *Research on the Hardware Structure Characteristics and EKF Filtering Algorithm of the Autopilot PIXHAWK*. Proceedings of 6th International Conference on Instrumentation & Measurement, Computer, Communication and Control, Jul 21-23, 2016. Harbin, pp.228-231, 2016, <http://dx.doi.org/10.1109/IMCCC.2016.128>,

- [9] –: Pixhawk PX4 Simulink/Matlab Embedded Coder Support: User Manual 2017.
<https://www.mathworks.com/hardware-support/pixhawk.html>,
- [10] Yates, R. and Lyons, R.: *DC Blocker Algorithms*.
IEEE Signal Processing Magazine **25**(2), 132-134, 2008,
<http://dx.doi.org/10.1109/MSP.2007.914713>,
- [11] Grewal, M.S. and Andrews, A.P.: *Kalman Filtering – Theory and Practice with MATLAB*.
John Wiley and Sons, New York, 2001,
- [12] Pavković, D.; Deur, J. and Lisac, A.: *A Torque Estimator-based Control Strategy for Oil-Well Drill-string Torsional Vibrations Active Damping Including an Auto-tuning Algorithm*.
Control Engineering Practice **19**(8), 836-850, 2011,
<http://dx.doi.org/10.1016/j.conengprac.2011.04.012>,
- [13] Pavković, D.; Lobrović, M.; Hrgetić, M.; Komljenović, A. and Smetko, V.: *Battery Current and Voltage Control System Design with Charging Application*.
Proceedings of 2014 IEEE Conference on Control Applications, Oct 8-10, 2014. Antibes, pp.1133-1138, 2014,
<http://dx.doi.org/10.1109/CCA.2014.6981481>,
- [14] Catlin, D.E.: *Estimation, Control, and the Discrete Kalman Filter*.
Springer-Verlag, New York, 1989,
- [15] Featherstone, R.: *Rigid body dynamics algorithms*.
Springer, New York, 2008,
- [16] Tommaso, B: *Modelling, identification and control of a quadrotor helicopter*. M.Sc Thesis.
Lund University, Lund, 2008.