Efficient BLDC controller for critical UAV applications

Pavel Kirienko,[*] Dmitry Ramensky, Anton Krol
Zubax Robotics
Domodedovskaya 27, Moscow, Russia 115582

ABSTRACT

In this work we developed a BLDC controller (ESC) for electrically propulsed UAV, that resolves two major issues of the current state-of-the-art approaches: efficiency and reliability. The first issue was addressed through implementation of the field-oriented control (FOC) method. The second issue was resolved by means of reliable doubly-redundant CAN bus communication interface and heuristic self-diagnostic algorithm. The firmware developed over the course of this project is available under the BSD license and is known as PX4ESC [1].

1 INTRODUCTION

1.1 *Problem statement*

Most modern unmanned aerial vehicles (UAV) leverage electric propulsion. While superior to many alternatives in terms of efficiency and maintenance costs [2], electrically propelled vehicles suffer from low energy density of currently avaiable batteries. This precludes the use of UAV in applications that require longer flight times or significant payloads, unless measures are taken to maximize the efficiency of the propulsion system. Hence, the first issue that we address is efficiency.

The second issue stems from the fact that a BLDC motor controller of an electrically propelled aircraft is a mission-critical component, failure of which is likely to be fatal for the vehicle. Despite lack of proper research to back up the following claim, our experience in the industry shows that failures in the powertrain may be responsible for majority of accidents involving light UAV. Hence, improving predictability and reliability of BLDC controllers should have significant effect on the overall safety of UAV operations.

1.2 *Our approach*

This work is devoted to design of a BLDC controller that addresses the two major issues described in the previous section.

We address the first issue - efficiency - through implementation of a highly efficient motor control solution based on the field-oriented control (FOC) technique.

The second issue - reliability and predictability - is addressed by means of a heuristic self-diagnostic algorithm, continuous status feedback, and use of doubly redundant CAN bus as a communication medium.

1.3 *Structure of the paper*

In the next section of the paper we describe our approach to implementation of an efficient motor control algorithm. The following part is dedicated to the software architercure and the reliable communication interface. We finalize the work with a brief description of the prototype hardware developed over the course of this project.

2 FIELD-ORIENTED CONTROL

2.1 *Control principle*

Field oriented control (FOC) provides smooth motion at low speeds, efficient operation at high speeds, and high dynamic characteristics [3]. The structure of the sensorless FOC algoritm is shown on the figure 1. Regulation is performed on the rotating dq axis. The current component is regulated to the reference value provided by the speed controller, while the direct current component is set to zero. The outputs of the current controllers, representing the voltage references, are then impressed to the motor using the SVPWM technique, once the inverse transformation from the rotating to the fixed stator axis is performed (Clarke and Park transforms). An outer speed control loop completes the scheme. All of the controllers used are PID controllers.

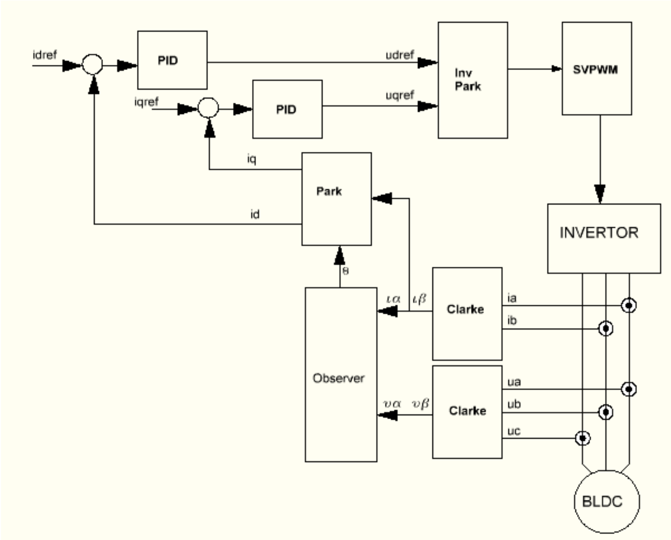

Fig. 1: FOC scheme.

---

[*] pavel.kirienko@zubax.com

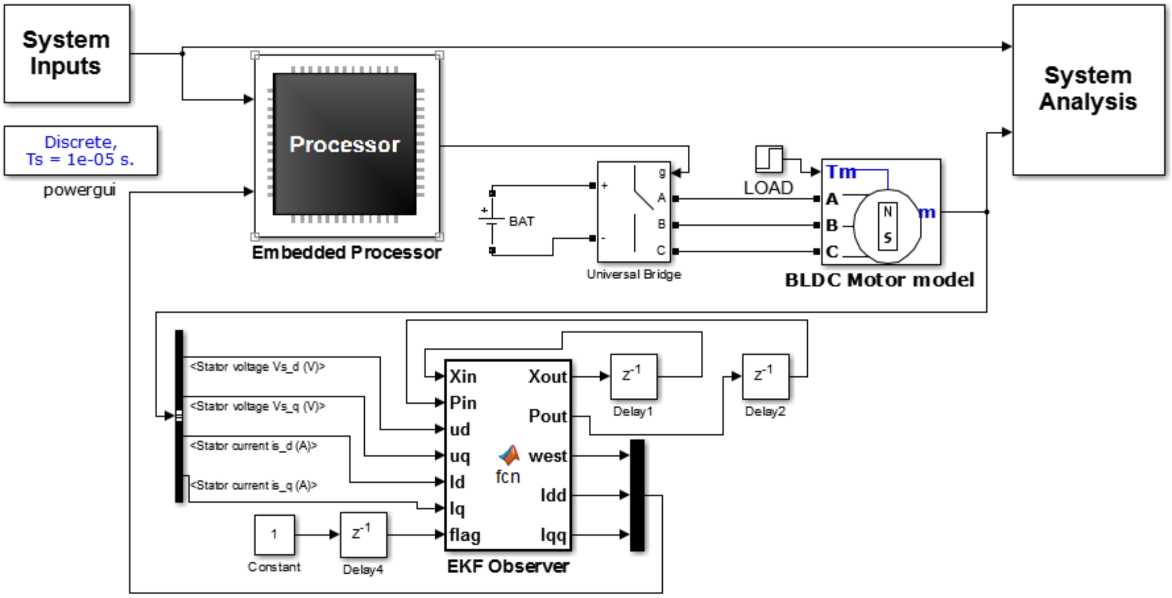

Fig. 2: Matlab/Simulink STM32 Mat-target test model.

2.2 *Modeling*

Most of the modern UAV motors lack sensors due to size and weight constraints, which leads one to resort to a sensorless solution. The most intricate and important part of a sensorless FOC BLDC controller is the state observer. Many research works have been done in speed and position estimation using sliding mode, high frequency signal injection method, fuzzy control, back EMF and voltage period measurement [4], and the extendended Kalman filter (EKF) [5]. EKF is the most attractive and popular of those listed, as it delivers rapid, precise, and accurate estimation [6]. Hence, we chose the EKF-based approach.

We chose Matlab/Simulink for modeling, as it is one of the most powerful instruments for modeling and design of BLDC controllers. EKF code is implemented in the Matlab/Simulink model (figure 2) as the S-Function block. Simulations performed on this model (figures 3, 4) and experiments on the real hardware demonstate good motor speed and rotor position tracking accuracy by the EKF observer.

Matlab/Simulink is a good instrument for rapid prototyping and HIL system building. In this project, an STM32 embedded target for MATLAB and Simulink is used to configure peripherial modules of the STM32 processor and perform processor in the loop simulations.

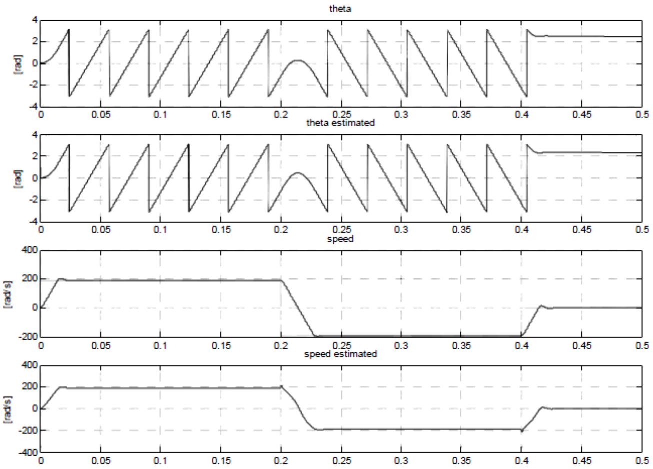

Fig. 3: FOC controller and observer performance: motor angle and speed states during speed reference changing.

3 SOFTWARE ARCHITECTURE

3.1 *General structure*

The embedded software is based on the NuttX RTOS. The choice of RTOS is justified by its permissive license, POSIX compatibility and support for nested interrupts. The latter trait is paramount for hard real-time processing performed by the FOC module, as interrupt nesting allows it to preempt execution of interrupt handlers of the RTOS kernel itself, which

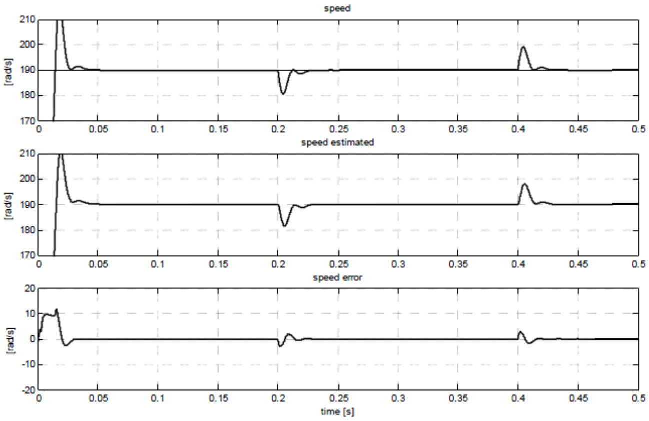

Fig. 4: FOC controller and observer performance: motor angle and speed states during load variation.

ensures lack of interference between RTOS and the hard real-time FOC processing.

We use libuavcan [7] to support the UAVCAN protocol (this is covered in the following subsection).

3.2 Control and feedback interface

We leverage UAVCAN [8] as a communication interface, which is a lightweight protocol designed for reliable communication in aerospace and robotic applications via CAN bus. UAVCAN natively supports redundant physical interfaces, which suits the reliability requirements. Also, UAVCAN implements higher-level management features, such as node reconfiguration and remote firmware upgrade. The following list briefly summarizes the features that are supported by our design through leveraging of UAVCAN; more detailed description follows:

- Control input.
- Status feedback.
- Remote reconfiguration.
- Remote firmware upgrade.
- Plug-and-play capability.

3.2.1 Control input

UAVCAN allows to transfer multiple ESC setpoints in a single CAN frame, which means that baseline latency is defined by the CAN frame transfer time. Worst case transfer time can be derived from maximum frame length, which is [9]: 128-bit long sequence plus at most 21 stuff bits, so the worst case frame length would be 149 bits. For a 1 Mbps CAN bus baud rate, this yields 149 microsecond latency. This shows that the chosen communication medium is superior in terms of latency to the popular analog interface that uses pulses of varying width between 1 and 2 milliseconds to encode control input.

3.2.2 Status feedback

UAVCAN reduces node health status to a scale of three values - **OK**, **Warning**, and **Critical**. Such reduction allows other nodes to easily derive overall system health status, and as a consequence it allows the flight controller, or the operator, to decide whether it is safe to continue the mission. We implemented a heuristic self-diagnostic algorithm that maps the internal states of the ESC to the health scale as follows:

- If the motor control algorithm fails (e.g. if the motor is blocked or damaged), **Critical** status is assumed.
- If the power stage temperature exceeds the first limit, **Warning** status is assumed.
- If the power stage temperature exceeds the second limit, **Critical** status is assumed.
- Otherwise, status **OK** is assumed.

Where the first and second temperature limits are set to 60 and 80°C, respectively.

Aside from the health scale, the full states are also published via standard messages defined by UAVCAN.

3.2.3 Remote firmware upgrade

The project utilizes the PX4 [10] UAVCAN bootloader for remote firmware upgrade over UAVCAN.

3.2.4 Plug-and-play capability

UAVCAN specification defines ways to connect nodes to the vehicle bus without prior configuration. Particularly, the following features are defined:

- Automatic CAN bus baud rate detection (autobauding).
- Automatic allocation of UAVCAN node ID.
- Automated detection of instance ID via feedback from the user.

The latter feature is demonstrated on these videos, where motor ID and direction of rotation are assigned to the controllers via direct user's input:

- `https://youtu.be/c9O0R9YJuOg`
- `https://youtu.be/6zGYhqyN6Uw`

4 HARDWARE ARCHITECTURE

The system is designed to be easily portable across different hardware implementations, as long as they all share the following traits and core parts:

- MCU STMicroelectronics STM32F446.
- FET driver IC TI DRV8302.
- MCU pinout is identical to the one used in the reference design.

Such portability ensures that the same design can be tailored to different application requirements through simple redesign of the power stage hardware. Since costs of hardware development are negligible compared to the costs of development of the embedded software, portability enables the project to easily answer the needs of wide variety of applications.

Prototype hardware is shown on the figure 5.

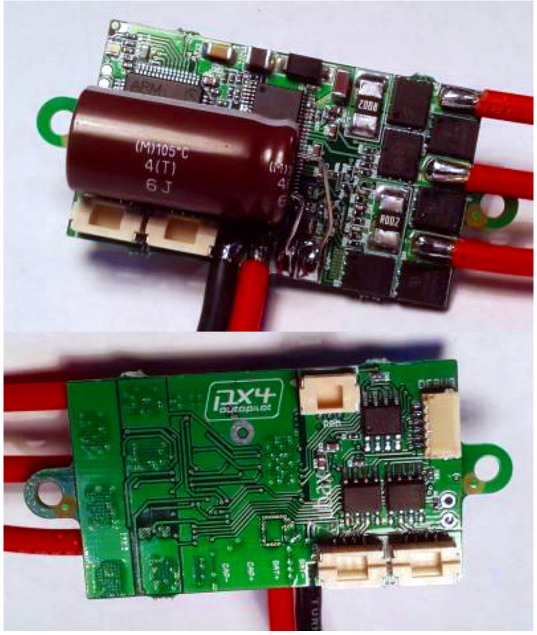

Fig. 5: Pixhawk ESC prototype that is compatible with PX4ESC firmware.

5 CONCLUSION

Over the course of this project, we designed a high-end BLDC controller for electrically propelled aerial vehicles. The main characteristics of this design are high efficiency and suitability for mission-critical applications.

The firmware is released at `http://github.com/PX4/px4esc` under the three-clause BSD license. The reference hardware sources will be available shortly at `https://pixhawk.org/modules/pixhawk_esc`.

REFERENCES

[1] PX4ESC - advanced multi-platform three phase brushless motor controller firmware. `https://pixhawk.org/firmware/px4esc`.

[2] J. Meyer D. L. Gabriel and F. Du Plessis. Brushless DC motor characterisation and selection for a fixed wing UAV. AFRICON, 2011.

[3] Copley Controls Corp. What is Field Oriented Control and what good is it? `http://www.maccon.de/fileadmin/FTPROOT/Field-Oriented-Control.pdf`.

[4] H. L. Chan and K. T. Woo. Closed loop speed control of miniature brushless DC motors. *Journal of Automation and Control Engineering*, 3(4), 2015.

[5] N. Mohan R. Dhaouadi and L. Norum. Design and implementation of an extended Kalman filter for the state estimation of a permanent magnet synchronous motor. *IEEE Transactions on Power Electronics*, 6:491–49, 1991.

[6] D. Janiszewski. Extended Kalman filter based speed sensorless PMSM control with load reconstruction. pages pp. 1465–1468. IECON, 2006.

[7] Libuavcan - reference implementation of UAVCAN for embedded systems and Linux. `https://github.com/UAVCAN/libuavcan`, 2014.

[8] Pavel Kirienko. UAVCAN - a lightweight protocol designed for reliable communication in aerospace and robotic applications via CAN bus. `http://uavcan.org`, 2014.

[9] Alan Burns Ken Tindell. Guaranteeing message latencies on control area network (CAN).

[10] PX4 - open-source high-end autopilot. `https://px4.io`.

[11] Dennis Nolah. Sensorless six-step BLDC commutation. `http://www.st.com/st-web-ui/static/active/en/resource/technical/document/application_note/DM00072008.pdf`, 2013.