LinkBoard: Advanced Flight Control System for Micro Unmanned Aerial Vehicles

Mariusz Wzorek, Piotr Rudol, Gianpaolo Conte, Patrick Doherty
Department of Computer and Information Science
Linköping University Linköping, Sweden
email: mariusz.wzorek@liu.se

***Abstract*—This paper presents the design and development of the LinkBoard, an advanced flight control system for micro Unmanned Aerial Vehicles (UAVs). Both hardware and software architectures are presented. The LinkBoard includes four processing units and a full inertial measurement unit. In the basic configuration, the software architecture includes a fully configurable set of control modes and sensor fusion algorithms for autonomous UAV operation. The system proposed allows for easy integration with new platforms, additional external sensors and a flexibility to trade off computational power, weight and power consumption. Due to the available onboard computational power, it has been used for computationally demanding applications such as the implementation of an autonomous indoor vision-based navigation system with all computations performed onboard. The autopilot has been manufactured and deployed on multiple UAVs. Examples of UAV systems built with the LinkBoard and their applications are presented, as well as an in-flight experimental performance evaluation of a newly developed attitude estimation filter.**

I. Introduction

The technologies associated with UAVs have been under research and development during several decades. As a result, the advantages of using these platforms have become apparent for many application areas. The tasks they can perform or assist with can be found in fields such as search and rescue, physical structure inspections, remote sensing, entertainment and research, to name just a few. Small-sized UAVs have many advantages due to the ease of operation, relatively low price and safety of operation. In order to maximize their usefulness, however, these UAVs have to operate with a level of autonomy which is adjustable to a specific task they perform.

Flight control systems (i.e. autopilots) play a crucial role in supporting this autonomy. Larger platforms with larger payloads have the capacity to carry and use advanced computer systems with considerable computational capabilities. However, in small UAV designs custom-made autopilots have to be used due to limited payload and computational power capacities. The idea behind the LinkBoard is to design and develop a small flight control system which is capable of running basic control modes and sensor fusion but also the advanced algorithms and functionalities needed when operating autonomously. This is achieved by maximizing the available computational power and minimizing the physical dimensions and weight. The system is designed to be highly configurable, flexible and easy to extend with new sensors, algorithms, platforms and applications.

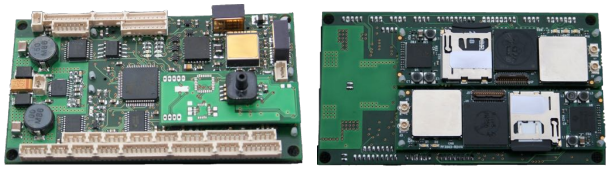

Fig. 1. Photo of front (left) and back (right) of the LinkBoard.

In recent years, a number of open-source and commercial autopilots for small UAVs have been developed. The PX4/Pixhawk [1] flight control system developed at ETH Zurich is one of the most popular. It uses one main Microcontroller Unit (MCU) integrated with an Inertial Measurement Unit (IMU) and implements sensor fusion and basic control modes. The autopilot supports a set of UAV platform configurations which are configurable through a Graphical User Interface (GUI). A limited number of parameters can be set using the interface. Additional, non-supported configurations have to be added through a firmware update.

The Paparazzi [2] is an open-source autopilot project started in 2003 at the French Civil Aviation University (ENAC) with focus on fixed-wing UAV control and later extended to support multirotor configurations. Through the years multiple hardware solutions have been developed with a recent generation based on a single MCU and a Gumstix Overo board (Lisa/L, www.gumstix.com) with an external IMU. The autopilot configuration is done through a GUI and allows for building custom firmware depending on the platform configuration. It weighs 30g without the IMU and the Gumstix board and has total dimensions of 50mm x 90mm.

Other autopilot designs based on a single MCU and an integrated IMU are less relevant for comparison due to limited computational resources. These include the Autoquad [3], the LibrePilot (previously OpenPilot) [4] and APM2.8 from the ArduPilot [5] project.

The LinkBoard in comparison integrates two MCUs, two Gumstix Overo boards and an IMU offering high computational power in a small and lightweight package. Depending on the application, the autopilot hardware can easily be reconfigured by adding or removing the two Gumstix boards, making it lighter and smaller or more powerful. Additionally, the onboard software allows for full configuration without the need to build or change the firmware. This includes not only basic

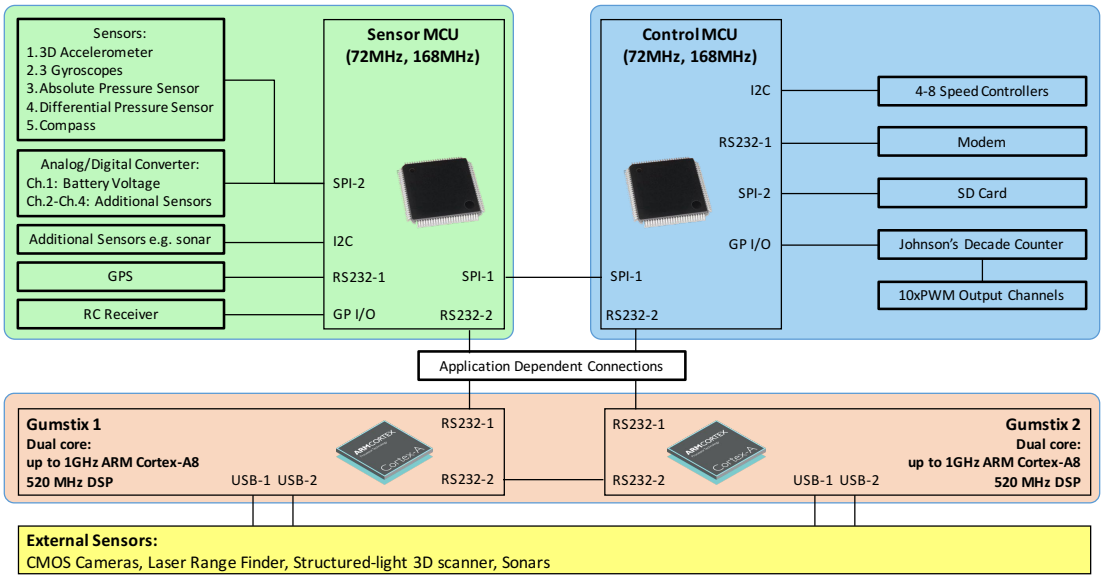

Fig. 2. The LinkBoard hardware architecture.

configuration parameters (e.g. control gains, filter parameters) but also the control system and its structure. The LinkBoard software allows for defining multiple sets of control systems based on a cascade of fully parametrized PID controllers.

The remainder of the paper is structured as follows. First, the hardware components and design of the LinkBoard are described in Sec. II, followed by the description of the software architecture in Sec. III. Application examples and experimental validation results are provided in Sec. IV. Finally, conclusions are provided in Sec. V.

II. The LinkBoard Hardware Architecture

The LinkBoard (Fig. 1) has a modular design that allows for adjusting the required computational power depending on mission requirements. In the full configuration, the LinkBoard weighs 30g, has very low power consumption and has a footprint smaller than a credit card (45mm x 80mm). The system is based on two ARM-Cortex microcontrollers which implement the core sensor fusion and flight functionalities and optionally, two Gumstix Overo boards for user software modules. The LinkBoard includes a three-axis accelerometer, three rate gyroscopes, and absolute as well as differential pressure sensors for estimation of the altitude and the air speed, respectively. The LinkBoard features a number of interfaces which allow for easy extension and integration of additional equipment. It supports various external devices (both hardware and software) such as a laser range finder, analogue and digital cameras on a gimbal, a GPS receiver, and a magnetometer.

The LinkBoard PCB design has been optimized for size and to minimize EMI interference. The board has six layers with three inner layers dedicated for ground, 3.3V and 5V signals. Fig. 2 presents the LinkBoard hardware architecture. More details are provided in the following subsections.

A. Computational Units

The core of the LinkBoard consists of two interconnected ARM Cortex Microcontroller Units (MCUs). In the basic version two STM32F103 MCUs from STMicroelectronics (www.st.com) are used. The STM32F103 (Cortex-M3) runs at 72MHz clock frequency which results in 1.25DMIPS/MHz. The MCU has a 32-bit RISC core with high-speed embedded memories (512kB of flash memory and 64kB of SRAM).

Alternatively, the LinkBoard can use STM32F415 running at 168MHz. The STM32F415 is based on a Cortex-M4 core which features a single precision Floating Point Unit (FPU), 1MB of flash memory and 192kB of SRAM.

One of the MCUs (Sensor) is used for sampling data from sensors and running the sensor fusion algorithms. The second MCU (Control) runs basic flight control modes, interfaces for communicating with ground control station and optional Gumstix boards as well as sends commands to actuators (e.g. servos, speed controllers). The Sensor and Control MCUs are interconnected with an SPI bus and exchange the necessary data at a 500Hz update rate.

Additionally, a 64Mb external flash memory is integrated on the LinkBoard (M25P64) and connected to the Control MCU. The memory is used to store software configuration active during the flight. More details are provided in Section III.

The two Cortex MCUs provide all the necessary functionalities for achieving autonomous flight and support many kinds of applications on their own. This allows for configuring the board to use minimal power. Additional user software modules implementing high-level functionalities can be executed on two Gumstix Overo boards running a Linux operating system. The LinkBoard supports headers for direct connection of the two Gumstix at the back side of the board. All necessary voltages and headers are integrated on the LinkBoard (plug and play).

B. Power Management

Small UAV platforms typically use a single Lithium-Polymer battery supply for the electrical motors and onboard systems, which often require different input voltages. In order to minimize the number of power converters and cables required, the LinkBoard was designed to run on a single power supply (from battery) and additionally provide the necessary voltages to its external sensors and actuators (e.g. servos). The integrated MCUs, sensors and I/O communication ports on the LinkBoard require the following voltage levels: 1.8V (I/O), 2.5V (gyroscope and Analog-to-Digital - ADC reference), 3.3V (sensors, MCUs) and 5V (I/O, ADC). In order to satisfy those requirements a two-stage power management system has been designed.

In the first stage a switching power supply LM26400Y from Texas Instruments (www.ti.com) is used to convert a single power input (7V to 20V) to 3.3V and 5V supplies with 2A of current each. The LM26400Y device is a dual PWM peak-current mode buck regulator with two integrated power MOSFET switches and in the LinkBoard a high 90% efficiency is achieved. The converted power is sufficient for powering the whole system and supplying currents (at 3.3V and 5V) to additional external devices, such as external sensors or a Pan-Tilt-Roll mechanism.

The reference voltage (2.5V) used by gyroscopes and ADC sensors is converted in the second stage from 3.3V by a REF3325 low-dropout converter. The I/O voltage (1.8V) used for RS232 communication between Gumstix Overo boards and external devices is done using Bidirectional Logic Level Translators (ADG3308) from Analog Devices (www.analog.com).

C. Integrated sensors

The design decision has been made to include a set of integrated sensors necessary for basic flight control modes directly on the LinkBoard. Additional sensors for implementing more advanced autonomous flight capabilities should be integrated externally. This approach has two advantages. First, some of the sensors have to be mounted externally, e.g. GPS module with integrated antenna, sonar or magnetometer to minimize magnetic field interference. Second, it is easier to upgrade external sensor modules without redesigning the core of the system. The integrated sensors connected to the Sensor MCU through the SPI bus include:

- 3-axis accelerometer - a MEMS linear accelerometer (LIS3LV02DQ) from STMicroelectronics has been used. The sensor has a $\pm$2G/$\pm$6G selectable range.
- 3 rate gyroscopes - high performance MEMS gyroscopes (ADXRS453) from Analog Devices have been integrated. The sensors support $\pm$300°/sec angular rate sensing. Additional features include ultrahigh vibration rejection (0.01°/sec/g), excellent null bias stability (16°/hour) and internal temperature compensation.
- Analog-to-Digital (ADC) converter - a 4 channel 12-bit ADC (AD7924) from Analog Devices has been chosen. The converter features a fast throughput rate of 1 MSPS at low power consumption (6mW). One of the sensor inputs is used to sample the battery voltage and the others can be used for external sensors (e.g. sonar).
- Absolute pressure sensor - a piezoresistive transducer (MPXH6115A) from NXP Semiconductors (www.nxp.com) has been integrated. The sensor has a range of 15 to 115 kPa with 1.5% maximum error and is temperature compensated (-40°C to +125°C). The sensor output is provided as an analog voltage. For easy and reliable integration the sensor is mounted on a small external board which includes a dedicated ADC converter for the analog sensor output. The board is then mounted directly on top of the LinkBoard through stiff headers and the SPI bus is used to sample the converted data.

D. External sensors

The LinkBoard supports various types of power supplies and communication interfaces as well as software drivers that enable application based reconfigurability, ease of use and fast platform integration. Examples of the external sensors integrated so far include the following:

- u-blox 7 GPS module with integrated antenna from u-blox (www.u-blox.com).
- 3-axis magnetometer (MicroMag 3) from PNI Sensor Corporation (www.pnicorp.com).
- Ultrasonic sensors from MaxBotix (www.maxbotix.com) for close to ground altitude estimation.
- CMOS color and monochrome cameras (FireFly MV) from Point Grey (www.ptgrey.com).
- Laser range finders (URG-04LX and UTM-30LX) from Hokuyo (www.hokuyo-aut.jp).
- A structured-light 3D scanner (Xtion Pro) from Asus (www.asus.com).

E. Other interfaces

The LinkBoard supports communication with a Ground Control Station (GCS) as well as other external devices through RS232 ports. As a standard, the communication with GCS is realized using a radio modem (PRM112, 2.4GHz) from Laird Technologies (www.lairdtech.com). Other radio modems with an RS232 port and 3.3V or 5V supply can be used.

Control of the platform actuators and its payload is supported on the LinkBoard (Control MCU) via I2C and PWM interfaces. In case of quadrotor platforms, a common interface to speed controllers is an I2C bus. The LinkBoard also supports up to 10 PWM outputs for platforms which require standard servo control signals. To minimize the usage of generic I/O ports on the Control MCU, a Johnson Decade Counter is used. PWM output can also be used for controlling the platforms payload. An example of such an application is the control of a Pan-Tilt-Roll mechanism for cameras. Six generic I/O ports (three for each MCU) are available as well. A standard RC receiver for the backup pilot can be used as the LinkBoard (Sensor MCU) supports up to 7 PWM input channels. An SD card interface has been integrated to allow fast logging of onboard data.

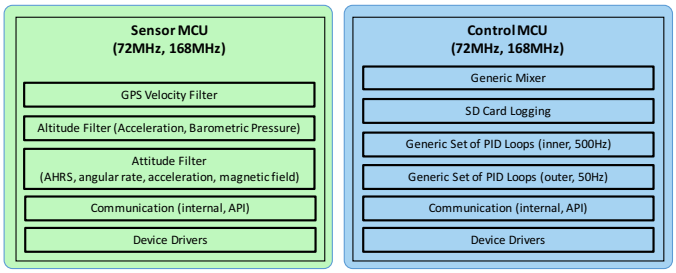

Fig. 3. Main software components of the LinkBoard for Control and Sensor MCUs.

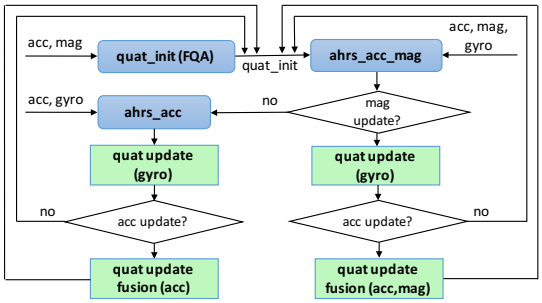

Fig. 4. The Attitude and Heading Reference System (AHRS) filter structure.

III. The LinkBoard software architecture

The design of the software is modular and highly configurable with APIs allowing for integrating the LinkBoard with various platforms as well as adapting it to different applications. All software components are generic and their configuration can be uploaded, updated on the fly and saved onboard. There is no need for recompilation of the firmware when using different types of UAV platform configurations.

A. Sensor MCU

The task of the Sensor MCU is collecting data from sensors (cf. hardware schematic in Fig. 2), executing sensor fusion algorithms and delivering the data to the Control MCU as well as external processors. The main functional modules are listed in Fig. 3.

A number of sensor fusion algorithms are implemented on the Sensor MCU. These include an Attitude and Heading Reference System (AHRS) filter, altitude filter and GPS velocity filter. The altitude filter combines barometric pressure with accelerations using a complementary filter to compute altitude and its first derivative (velocity). High frequency accelerometer data is combined with low frequency barometric pressure readings to produce estimates at the rate of 500Hz. Similarly, the GPS velocity complementary filter combines accelerations with GPS velocities to estimate velocity at the rate of 500Hz.

Estimating the attitude angles (pitch, roll and yaw) of the platform is an important task in a flight control system. The estimated angles are mainly used for the stabilization and control of the attitude dynamics. The sensor fusion algorithm implemented on the LinkBoard combines the information from an accelerometer, a magnetometer and an orthogonally mounted triad of gyroscopes. The estimation is independent from the GPS measurement which has two implications. First, the attitude angles are not affected by GPS outages, which makes it useful for indoor navigation operations. Second, the pitch and roll estimations depend on the sensed gravity field acceleration. The consequence of this fact is that in case of accelerated flight the estimated angles will be affected by a bias error. The AHRS filter structure is depicted in Fig. 4 and it is implemented using the quaternion attitude representation.

The initial quaternions are estimated using the Factored Quaternion Algorithm (FQA) [6] from the accelerometer and the magnetometer data. The FQA is a single frame deterministic method for solving Wahba's problem. After initialization, the quaternions are updated using the algorithm developed in [7]. The algorithm belongs to the class of complementary filtering methods. This type of method focuses on simple algorithms that are computationally lighter than conventional Kalman-based approaches. In addition, only one tuning parameter is required. The algorithm estimates the attitude in quaternions according to the following equation:

$$\boldsymbol{q}_{est,t} = \gamma_t \boldsymbol{q}_{\Delta,t} + (1-\gamma_t)\boldsymbol{q}_{\omega,t} \qquad 0 \leq \gamma_t \leq 1 \qquad (1)$$

where $\boldsymbol{q}_{\Delta,t}$ represents the quaternion update using accelerometer and magnetometer measurements. A gradient descent optimization algorithm is employed to compute $\boldsymbol{q}_{\Delta,t}$ (in the implementation only one iteration suffices). $\boldsymbol{q}_{\omega,t}$ represents the quaternion update using gyro measurements. The quantities γ_t and $(1-\gamma_t)$ are weights applied to the accelerometer/magnetometer contribution and to the gyroscopes' contribution respectively. The goal of the fusion algorithm is to provide an orientation estimate where $\boldsymbol{q}_{\omega,t}$ is used to filter out high frequency errors in $\boldsymbol{q}_{\Delta,t}$ and $\boldsymbol{q}_{\Delta,t}$ is used to compensate for integral drift in $\boldsymbol{q}_{\omega,t}$.

The algorithm depicted in Fig. 4 starts with the quaternion initialization (quat_init) using the FQA method. Afterwards, checks are performed whether new magnetometer and accelerometer data is available at the current time step. Since the gyroscopes used on the LinkBoard provide new data samples at the filter update rate (500Hz), the filter always updates quaternions based on angular rates. Fusion of accelerometer and/or magnetometer data is performed only when new samples are available (167Hz and 66Hz, respectively).

The results of the presented filters, as well as raw sensor data, are sent to the Control MCU at the full rate of 500Hz. Additionally, the data is accessible through an API. Selected variables can be sent through an RS232 connection to an external processor (e.g. Gumstix) which is useful in applications requiring raw data or additional processing to be performed. The rate is configurable and depends on the amount of data requested.

B. Control MCU

The main task of the Control MCU is implementing the control system, interfacing with the actuators (servo mo-

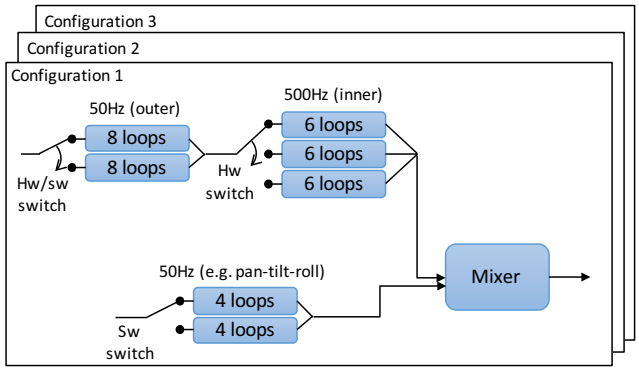

Fig. 5. LinkBoard Control MCU control structure reconfiguration.

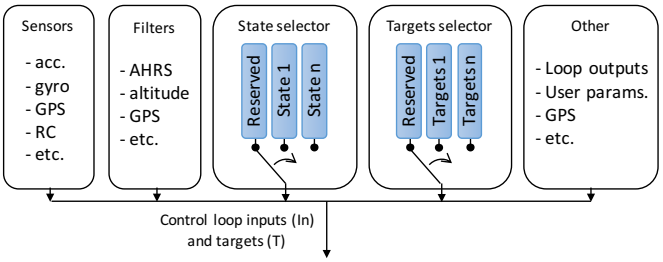

Fig. 6. Control loop inputs and targets.

tors and Electronic Speed Controllers - ESC), managing the onboard data logging on an SD card, and providing data to external processors including the telemetry data using a wireless modem.

One of the unique features of the LinkBoard software is the ability to fully specify a custom parametrized control system without the need for recompilation. Fig. 5 presents the reconfiguration schematic of the LinkBoard control system. A number of generic and configurable control loops are organized into three groups. First, one of three sets of six loops can be selected using a hardware switch associated with a switch on an RC transmitter of the backup pilot. These loops are calculated at 500Hz and are typically used to implement attitude stabilization (inner loops). The switch between the sets allows for choosing different flight modes, for example Manual and Autonomous Flight Modes. Second, one of two sets of eight control loops are available to implement the second layer of control and are typically used to implement outer control loops which are executed at a lower rate of 50Hz. Additionally, two sets of four control loops are available and are typically configured to control the Pan-Tilt-Roll unit of the camera.

The generic and fully configurable control loops follow the Proportional-Integral-Derivative (PID) general structure.

$$Output = sat^{+}_{-}(P + I + D + FF)K_{out} + O_{out} \quad (2)$$

$$P, I, D = sat^{+}_{-}((In + O_{in})S_{in} - (T + O_t)S_t)K \quad (3)$$

$$FF = sat^{+}_{-}((In + O_{in})S_{in}) \quad (4)$$

Each control loop has the structure presented in Eq. 2-4. The output of a loop is calculated by adding P, I, D and FF (Feed Forward) components, applying a gain K_{out} and offsetting the result by the value of O_{out}. The final result is then limited to the specified upper and lower limits (sat^{+}_{-}). P, I and D components are calculated by adding offsets O to the input and target variables (In and T, respectively) and scaling the result by S. The intermediate result is multiplied by respective gain K and separately limited. A similar operation is performed to calculate the value of the FF component, but the target part is not calculated. Each input and target variable can be specified to be transformed to the body coordinate frame (e.g. from GPS coordinates). Additionally, each of the P, I and D components can be made 360° continuous. The D component can be a classical derivative or can be specified to be treated as a proportional component, for example when the input and target variables are already first derivatives. Additionally, a low-pass filter can be applied to the D component output. The I component has an additional limiter which can be applied to the accumulator of the term.

Input and target variables (In and T) can be selected from a list (see Fig. 6) of available variables related to: sensors (e.g. accelerometer, gyroscopes, GPS, RC receiver), filters (attitude, altitude, velocity), output from another loop (cascading), UAV State and UAV Targets, and User Parameters. The three latter ones have special properties.

The UAV State input variables are related to the onboard handling of data associated with the platform's position, attitude and their first and second derivatives. There exists a number of state sets which can be injected and switched between at runtime (cf. Fig. 6). One of these sets is reserved for the state estimated onboard using the built-in filters. Other sets can be used to inject results (through an RS232 connection) of state estimation performed externally, for example, based on image processing or from a motion capture system. This approach allows for using the same control system structure with different state estimate sources. The UAV Targets variables are handled in a very similar fashion but relate to the target state. One set is reserved for the onboard higher-level control. It is responsible for implementing flight behaviors from take-off to landing which are configured using flight commands. These are then fed into a setpoint generation step which enforces limits on positions and velocities depending on a platform's properties or mission requirements. The details of the high-level flight behavior definition system are provided in [8].

General purpose User Parameters (40 variables of different byte sizes: float, 8, 16, and 32 bit integers) are available for applications which require additional non-standard inputs. These parameters are a way of future proofing the system to support applications which were not thought of during the development of the system.

Outputs of control loops can finally be mixed and assigned to servos or ESCs. Each of these outputs can be configured as a mix of inputs (control loop outputs) by adding or subtracting them to produce the final signal.

The ability to reconfigure the control loop structures and

Fig. 7. The LinkMAV coaxial platform (left). The LinkQuad platform with a structured-light 3D scanner and a Pan-Tilt color camera unit (middle). The newest generation of the LinkQuad platform with a laser range finder, an external GPS and a Pan-Tilt-Roll color camera unit (right).

specify their parameters with high granularity allows for adopting the control system to an application or platform at hand. For example, it is possible to inject signals to directly control the servos or ESCs effectively replacing the available control system. Similarly, it is possible to reuse the attitude stabilization but replace the position stabilization functionalities, or keep both and replace the flight behavior specification aspect of the LinkBoard software. Alternatively, it is straightforward to use the complete existing control system but provide a different state estimate to be used for controlling a platform. This makes the system very versatile and flexible.

C. Other considerations

The software on both Sensor and Control MCUs is divided into tasks which are executed as single threaded programs. A static schedule is used which takes into account the worst execution times of all tasks. This approach guarantees fully deterministic execution times for all the tasks.

The onboard data is logged at the rate of 500Hz on an SD card. This is essential for development as well as verification purposes. All input and target variables as well as outputs of the control loops are logged along with the data which allows to completely recreate the state of the system off-line. The same set of variables is available for streaming to the ground during flight in form of telemetry data. The data rate depends on the number of the selected variables.

The LinkBoard supports a piezoelectric buzzer used to indicate the state of the system. User configured sounds can be associated with events such as system initialization, flight mode switching, low battery warning etc.

The complete configuration of the system is saved to one of three configurations on the onboard flash memory. This allows for defining specific configuration for various applications e.g. indoor and outdoor flight, experimental configurations or for different platforms.

The LinkBoard is configured and operated though a custom user interface program called LinkGS. It is used for low-level configuration of the system. Additionally, APIs exist which allow for retrieving and injecting data as well as issuing commands to the system at runtime. On top of the API, a Robot Operating System (www.ros.org) wrapper exists which allows for integrating the system with the ROS ecosystem.

IV. Example Applications and Experimental Evaluation

The LinkBoard has been integrated and tested on a number of UAV platforms in various applications presented in multiple publications. Fig. 7 presents several example platform configurations with different sensor payloads.

The LinkMAV [9] coaxial UAV was used during the 3rd US-European Competition on Micro Air Vehicles (2007) and placed 3rd in the indoor challenge. Due to using the LinkBoard autopilot the flights were performed in a mix of assisted and autonomous modes.

The LinkQuad quadrotor platform with various sensor payloads was used in the development and deployment of applications described below.

The MAVwork [10] framework was deployed and used for non-GPS vision-based outdoor navigation applications [11] [12]. In [13] a vision-based pose estimation for autonomous indoor navigation was presented. An active vision framework has been developed and implemented on the LinkQuad platform [14]. The framework employs state-of-the-art object detection, object tracking, Bayesian filtering and AI-based methods. The experimental evaluation showed efficient detection and tracking of persons in real-time and was used for virtual leashing. In all three applications a color camera mounted on Pan-Tilt-Roll unit was used as the main sensor.

An evaluation of two reactive collision algorithms deployed on the LinkQuad has been presented in [15]. The algorithms proposed used a structured-light 3D scanner (Fig. 7, middle) for sense-and-avoid behavior.

A problem of area coverage with heterogeneous UAVs using scan patterns was addressed in [16]. The solution proposed divides the scanning task among UAVs and generates optimal trajectories while taking into account the differences in capabilities between platforms and sensors. The experimental evaluation included the LinkQuad platform equipped with a laser range finder and a color camera sensor.

A Machine Learning based navigation framework has been developed. The framework deals with avoiding dynamic obstacles in cluttered environments without prior coordination. The approach proposed combines policy search with model-predictive control and has been successfully tested in simulation and real-flight avoiding moving humans [17].

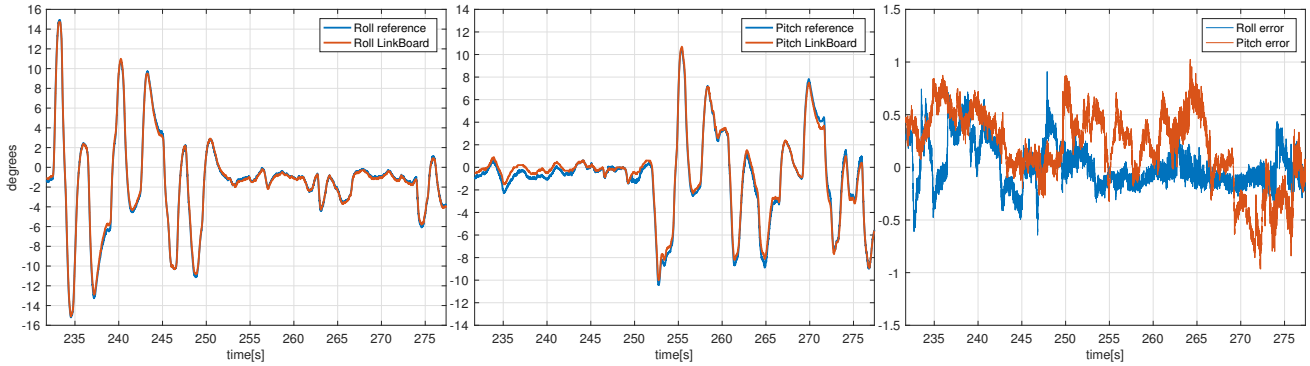

Fig. 8. The AHRS roll and pitch angle estimation results compared to the reference system.

The newly developed AHRS estimation filter presented in Sec. III-A has been evaluated in flight. The experiment was performed with an external tracking system from Vicon (www.vicon.com) as a reference. The flight was done in attitude stabilized mode to collect data for accuracy analysis. Fig. 8 presents onboard roll and pitch angle estimations from the AHRS filter in comparison to the Vicon reference and the resulting errors. In both cases the absolute estimation error was within $\pm 1^\circ$. The Root Mean Square Error (RMSE) for roll was 0.21° with maximum error of 0.91°. In pitch axis the RMSE was 0.39° with maximum error of 1.02°. Similar performance was confirmed for yaw angle estimation. Additionally, full 360° rotations in all axis were performed to validate continuity of the estimation as well as filter convergence. In summary, the AHRS filter evaluation has been shown to have more than satisfactory performance for accurate UAV attitude control.

V. Conclusions

In this paper we have presented the LinkBoard, an advanced flight control system for micro UAVs. Both hardware and software aspects of the LinkBoard design were presented. The hardware development was focused on minimizing its size and integrating an IMU and multiple microcontroller units to maximize the available computational power. The LinkBoard weighs 30g in full configuration and is smaller than the size of a credit card. Software developed onboard allows for full reconfiguration without the need to rebuild the firmware. This includes the control system structure and its parameters. These features make it a unique solution in comparison to other commercial and open-source systems. Multiple examples of systems developed with the LinkBoard and their applications were described and referenced. Additionally, a newly deployed AHRS complementary filter has been presented and evaluated in real flight. The result analysis showed high accuracy of the angle estimation as compared to the reference system.

Acknowledgment

This work is partially supported by the Swedish Research Council (VR) Linnaeus Center CADICS, the ELLIIT network organization for Information and Communication Technology, and the Swedish Foundation for Strategic Research (Smart Systems: RIT 15-0097).

References

[1] L. Meier, D. Honegger, and M. Pollefeys, "PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[2] P. Brisset, A. Drouin, M. Gorraz, P.-S. Huard, and J. Tyler, "The paparazzi solution," in *MAV 2006, 2nd US-European Competition and Workshop on Micro Air Vehicles*, 2006.

[3] "Autoqaud project." [Online]. Available: http://autoquad.org/about/

[4] "Librepilot project." [Online]. Available: https://www.librepilot.org

[5] "Ardupilot project." [Online]. Available: www.ardupilot.org/

[6] X. Yun, E. R. Bachmann, and R. B. McGhee, "A simplified quaternion-based algorithm for orientation estimation from earth gravity and magnetic field measurements," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 3, pp. 638–650, March 2008.

[7] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of imu and marg orientation using a gradient descent algorithm," in *IEEE International Conference on Rehabilitation Robotics*, 2011.

[8] P. Rudol and P. Doherty, "Versatile and expressive flight behaviour definition of vtol uavs for emergency assistance missions." in *International Symposium on Safety, Security and Rescue Robotics*, 2016.

[9] S. Duranti, G. Conte, D. Lundström, P. Rudol, M. Wzorek, and P. Doherty, "Linkmav, a prototype rotary wing micro aerial vehicle," in *17th IFAC Symposium on Automatic Control in Aerospace*, 2007.

[10] I. Mellado-Bataller, J. Pestana, M. A. Olivares-Mendez, P. Campoy, and L. Mejias, *MAVwork: A Framework for Unified Interfacing between Micro Aerial Vehicles and Visual Controllers*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 165–179.

[11] J. Pestana, I. Mellado-Bataller, J. L. Sanchez-Lopez, C. Fu, I. F. Mondragón, and P. Campoy, "A general purpose configurable controller for indoors and outdoors gps-denied navigation for multirotor unmanned aerial vehicles," *Journal of Intelligent & Robotic Systems*, vol. 73, 2014.

[12] I. Mellado-Bataller, "Autonomous linkquad quadcopter with computer vision." [Online]. Available: http://www.ignaciomellado.es/projects/mavwork-support-for-the-linkquad-quadcopter

[13] P. Rudol, M. Wzorek, and P. Doherty, "Vision-based pose estimation for autonomous indoor navigation of micro-scale unmanned aircraft systems," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

[14] M. Danelljan, F. S. Khan, M. Felsberg, K. Granström, F. Heintz, P. Rudol, M. Wzorek, J. Kvarnström, and P. Doherty, "A low-level active vision framework for collaborative unmanned aircraft systems," in *Lecture Notes in Computer Science*, vol. 8925, no. 8925, 2015.

[15] C. Berger, P. Rudol, M. Wzorek, and K. A., "Evaluation of reactive obstacle avoidance algorithms for a quadcopter." in *International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2016.

[16] C. Berger, M. Wzorek, J. Kvarnström, G. Conte, P. Doherty, and A. Eriksson, "Area coverage with heterogeneous uavs using scan patterns." in *International Symposium on Safety, Security and Rescue Robotics*, 2016.

[17] O. Andersson, M. Wzorek, P. Rudol, and P. Doherty, "Model-predictive control with stochastic collision avoidance using bayesian policy optimization," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.