

Reconnaissance micro UAV system

Petr Gabrlík
CEITEC

Central European Institute of Technology
Brno University of Technology
616 00 Brno, Czech Republic
Email: petr.gabrlík@ceitec.vutbr.cz

Vlastimil Kriz
CEITEC

Central European Institute of Technology
Brno University of Technology
616 00 Brno, Czech Republic
Email: vlastimil.kriz@ceitec.vutbr.cz

Ludek Zalud

Faculty of Electrical Engineering
and Communication
Brno University of Technology
616 00 Brno, Czech Republic
Email: zalud@feec.vutbr.cz

Abstract—This paper describes the design and implementation of the Uranus UAV. This quad-rotor flying robot was created to extend the abilities of the hitherto developed with airborne missions. The first part deals with the mathematical model of the robot. Next, the control system is designed, and the proposed hardware as well as the implemented software solution are presented. For integration into the robotic system, a new communication protocol was created and is described here too.

I. INTRODUCTION

In the research laboratories of CEITEC and BUT, the Cassandra robotic system has been developed [1]. This is a heterogeneous group of robots controlled mainly by telepresence from the operator base station. Within the described concept, each robot is assigned a specific task, such as reconnaissance, terrain mapping or contamination measurement. The intended purpose of the entire group is to perform these tasks in locations which are dangerous or inaccessible for humans. Until recently, this group included only ground robots. In order to enable the group to pursue the assigned tasks from the air or to reach locations unapproachable for ground robots, it was decided to extend the group with a flying robot.

To facilitate operation in a closed environment where airplanes can manoeuvre only with difficulty, an aircraft with the ability to hover was required. Because airships provide a poor payload/dimensions ratio, helicopters were taken into consideration; the first attempts were carried out with classical machines. However, such helicopters were not robust enough for the intended use in a difficult environment, mainly due to the fact that they contain lot of small moving parts which get easily damaged. Quadrotors (and multi-rotor copters in general) have proved to be the best platform for our purposes. The only moving parts on these devices are their rotors rigidly connected to the engines. Everything is controlled just by changing their rotor speeds. Yet this solution also exhibits a disadvantage, namely internal instability; thus, a control system needs to be implemented to provide the necessary operational ability [2].

Currently, quadrotors are being developed within a large number of projects, from hobby to professional ones. After investigating some of these projects, we considered the given status and decided to create our own solution. This approach offers a significant advantage: as the applied software is not license-restricted, we will have full control over the solution and will be able to modify our device if necessary. In this context, a specific airframe and control system were created. To design the control system, the behavior of the designed



Fig. 1. Uranus UAV during a flight

airframe with rotors was described and mathematical model assembled.

II. QUADROTOR MODEL

The mathematical model of the Uranus UAV comprises two parts. The first part handles the relation between the speed of the rotors and the forces and torques affecting the quadrotor rigid body. The second portion defines the dynamics of the rigid body, including transformation between frames and effect of gravity force. This separation allows us to design a controller with better dynamics, because the first part of the model can be easily linearized in the controller.

The forces and torques caused by propellers of the rotors and affecting the quadrotor rigid body are denoted as u_1 to u_4 . The meaning is as follows: u_1 - torque around the x axis caused by different thrust of rotors 2 and 4; u_2 - torque around the y axis caused by different thrust of rotors 1 and 3; u_3 - torque around the z axis caused by different reaction torques from rotors rotating in opposite directions; u_4 - force in the z axis caused by common thrust of all the rotors.

The relation between the thrust and propeller speed of a rotor is

$$F_T = k_T \cdot n^2, \quad (1)$$

and the relation between the reaction torque and propeller speed of a rotor is

$$M_R = k_M \cdot n^2. \quad (2)$$

Here, k_T and k_M are constants determined by the measurement. The first part of the model then could be described

as

$$\begin{aligned}
u_1 &= (-n_2^2 + n_4^2)k_T l \\
u_2 &= (n_1^2 - n_3^2)k_T l \\
u_3 &= (-n_1 + n_2 - n_3 + n_4)k_M \\
u_4 &= (-n_1^2 + n_2^2 - n_3^2 + n_4^2)k_T \quad (3)
\end{aligned}$$

l denotes the distance between the rotors and the center of gravity of a quadrotor.

The second part of the model can be further divided into four parts: Equations that describe rotational movement; equations of linear movement; equations describing transformation between frames; and equations expressing the effect of gravity force.

All variables are defined in 2 frames: robot frame and ground frame. The robot frame is bound to the UAV while the x axis faces forward, y faces right, and z faces down. The ground frame is bound with the ground and oriented north-east-down.

Rotational movement (robot frame):

$$\begin{aligned}
\dot{\omega}_x &= \frac{(I_{yy} - I_{zz})\omega_y\omega_z + u_1}{I_{xx}} \\
\dot{\omega}_y &= \frac{(I_{zz} - I_{xx})\omega_x\omega_z + u_2}{I_{yy}} \\
\dot{\omega}_z &= \frac{(I_{xx} - I_{yy})\omega_z\omega_y + u_3}{I_{zz}} \quad (4)
\end{aligned}$$

I_{xx} , I_{yy} and I_{zz} represent moments of inertia around a given axis.

Translation movement (robot frame):

$$\begin{aligned}
\dot{v}_x &= \frac{m\omega_y v_z + m\omega_z v_y + G_x}{m} \\
\dot{v}_y &= \frac{m\omega_x v_z + m\omega_z v_x + G_y}{m} \\
\dot{v}_z &= \frac{m\omega_x v_y + m\omega_y v_x + G_z - u_4}{m} \quad (5)
\end{aligned}$$

G_x , G_y and G_z are the gravity forces component in a given axis:

$$\begin{aligned}
G_x &= -mg \sin \theta \\
G_y &= mg \cos \theta \sin \phi \\
G_z &= mg \cos \theta \cos \phi \quad (6)
\end{aligned}$$

Transformation between the robot frame and ground

frame [3]:

$$\begin{aligned}
\dot{\phi} &= \omega_x + \omega_y \sin \phi \tan \theta + \omega_z \cos \theta \tan \theta \\
\dot{\theta} &= \omega_y \cos \theta - \omega_z \sin \phi \\
\dot{\psi} &= \omega_y \frac{\sin \phi}{\cos \theta} + \omega_z \frac{\cos \phi}{\cos \theta} \\
\dot{x} &= v_z(\sin \phi \sin \psi + \cos \phi \cos \psi \sin \theta) - \\
&\quad -v_y(\cos \phi \sin \psi - \cos \psi \sin \phi \sin \theta) + \\
&\quad +v_x \cos \theta \cos \psi \\
\dot{y} &= v_y(\cos \phi \cos \psi + \sin \phi \sin \theta \sin \psi) - \\
&\quad -v_z(\cos \psi \sin \phi - \cos \phi \sin \theta \sin \psi) + \\
&\quad +v_x \cos \theta \cos \psi \\
\dot{z} &= v_z \cos \phi \cos \theta - v_x \sin \theta + v_y \cos \theta \sin \phi \quad (7)
\end{aligned}$$

The parameters m, l, k_T, k_M of the models was measured and I_{xx}, I_{yy}, I_{zz} were computed. The values are as follows:

$$\begin{aligned}
m &= 1.05 \text{ kg} \\
l &= 0.35 \text{ m} \\
k_T &= 0.14 \cdot 10^{-6} \text{ N} \cdot \text{RPM}^{-2} \\
k_M &= 4.6 \cdot 10^{-9} \text{ Nm} \cdot \text{RPM}^{-2} \\
I_{xx} &= 8.6 \cdot 10^{-3} \text{ kg} \cdot \text{m}^2 \\
I_{yy} &= 8.6 \cdot 10^{-3} \text{ kg} \cdot \text{m}^2 \\
I_{zz} &= 12.0 \cdot 10^{-3} \text{ kg} \cdot \text{m}^2 \quad (8)
\end{aligned}$$

III. CONTROLLER DESIGN

To stabilize the Uranus UAV, a controller based on state space representation of the quadrotor was designed.

As shown in Sect. II, the model of the quadrotor comprised 2 parts. The first part handles the non-linear relation between the rotation speed of the propellers and the introduced variables $u_1 - u_4$ denoting the forces and torques affecting the rigid body. Because this part of the model does not contain any dynamics, this relation can be expressed by purely static equations (3). To handle this, a mathematical block with inverse function (9) is implemented in the controller.

$$\begin{aligned}
n_1 &= \sqrt{\frac{2k_M u_2 - k_T l u_3 + k_M l u_4}{4k_M k_T l}} \\
n_2 &= \sqrt{\frac{-2k_M u_1 + k_T l u_3 + k_M l u_4}{4k_M k_T l}} \\
n_3 &= \sqrt{\frac{-2k_M u_2 - k_T l u_3 + k_M l u_4}{4k_M k_T l}} \\
n_4 &= \sqrt{\frac{2k_M u_1 + k_T l u_3 + k_M l u_4}{4k_M k_T l}} \quad (9)
\end{aligned}$$

This block, together with the first part of the model, will act as a linear section with unitary transfer function (while

operating in the working range of the rotor drivers). This allows direct use of the variables $u_1 - u_4$ as inputs to the system and enables us to treat these parts of the system as linear.

The schema of the system with the controller is shown in Fig. 2; here, a classical state space controller is extended by few enhancements. The first one is a bias (u_{4-0}) added to the input u_4 , whose function is to ensure the thrust needed to keep the quadrotor in hover flight. The next one consists in adding a new state I_Z , which is an integral of the altitude error, z . The purpose of this is to achieve zero steady-state error while hovering, because it is almost impossible to set up a thrust that exactly compensates the gravity force.

The design of the state space controller is based on a linearized model of the rigid part of the quadrotor. Equations (6) and (7) are linearized around the working point, which is hovering. For the equilibrium point, we have

$$\dot{\phi} = \theta = \omega_x = \omega_y = 0. \quad (10)$$

To reach this state, the thrust of the rotors must compensate the gravity force acting in z axis at horizontal flight. Thus

$$u_{4-0} = mg. \quad (11)$$

The system matrices **A** and **B** required for the controller design are derived from linearized Equations (4), (5), (6) and (7). The matrix **C** is an identity matrix, and the matrix **D** is a zero matrix. The matrix **K**, which determines the control law, was obtained experimentally by the pole placement method.

Matrix **K**:

	ω_x	ω_y	ω_z	v_x	v_y	v_z	ϕ	θ	ψ	x	y	z	I_z
u_1	1.05	0	0	0	0.29	0	2.66	0	0	0	0.12	0	0
u_2	0	1.05	0	-0.29	0	0	2.66	0	-0.12	0	0	0	0
u_3	0	0	0.66	0	0	0	0	0	0.44	0	0	0	0
u_4	0	0	0	0	0	-4.08	0	0	0	0	0	-6.32	-3.06

It can be observed from this matrix that the linearized state space description decoupled the system into independent subsystems for each axis.; the controller therefore handles it in this manner. In fact, from the perspective of attitude control, the state space controller can be understood as a set of separated PI or PD controllers. The results then prove that the approach involving a space controller and the approach employing PID controllers provide comparable results in this mode of flight.

First, the real behavior of the designed controller was tested in a laboratory. Figs. 3 - 5 show the step responses for the angular rotations roll, pitch, and yaw, which were all measured independently on a special tool. For the altitude stability and response, data from an ultrasound sensor were used (Fig. 6).

IV. IMPLEMENTATION

A. Construction

The Uranus UAV has a well-known, four-rotor structure often referred to as quadrotor or quadcopter. The construction consists of two orthogonal aluminum beams, which form a symmetric cross. All the electronic devices and printed circuit boards (PCB) are situated in the middle of the cross

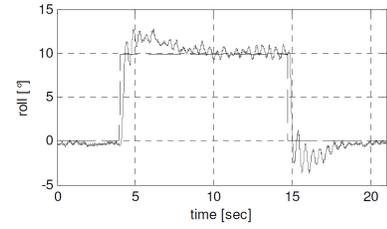


Fig. 3. Roll angle step response

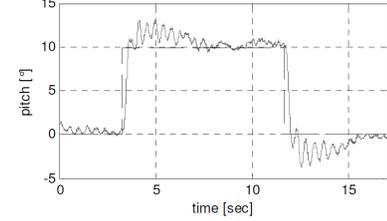


Fig. 4. Pitch angle step response

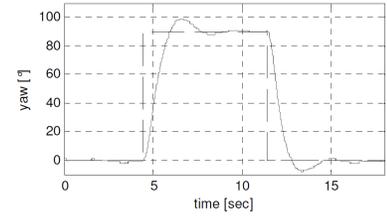


Fig. 5. Yaw angle step response

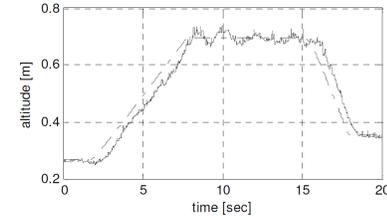


Fig. 6. Altitude response

to ensure that the center of gravity is also in the middle. At the end of each beam there is one brushless DC (BLDC) AXI 2212/34 electromotor equipped with a Graupner 11"x 5" carbon propeller. To suppress the mechanical vibrations, the motors are mounted on the rigid quadrotor frame through rubber pads.

The UAV overall dimensions are 701x701x188 mm, and the weight is 1.05 kg. The maximal combined motor thrust is about 24 N, so it is possible to carry a 150 g payload without reducing the maneuverability. The construction is mechanically simple and also allows additional modifications.

B. Hardware

The UAV is equipped with several PCBs. The central one is the control board, which contains a control microcontroller (MCU), a power supply, and various sensors. The other boards are the communication module, BLDC controllers, and ultrasound sensor. The sensor unit and GPS receiver are encased

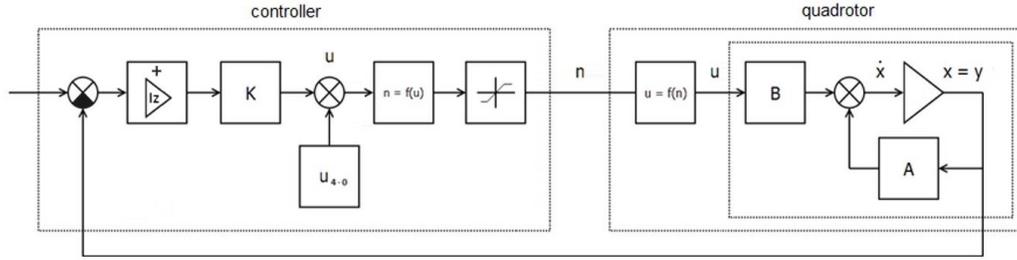


Fig. 2. Designed controller

in their own packages, and they are connected to the control board. The block scheme of the Uranus UAV system is shown in Fig. 7.

All the electronic devices on the UAV are powered by a Li-Pol accumulator. In this construction, a 3-cell accumulator with the nominal voltage of 11.1 V, and the capacity of 2200 mAh is used. This voltage is fed to all the engines, while the remaining electronics are supplied with 3.3 V and 5 V. The voltage level reduction is performed by switching and LDO (Low-dropout Regulator) stabilizers.

The control board was designed specifically for this application. The main part of the board consists in an MCU LM3S8962, which is based on the 32-bit ARM Cortex M3 architecture and is able to operate at frequencies up to 50 MHz. It provides sufficient computing power [4], all the required communication buses (SPI, UART, I²C), and internal A/D converters.

An Xsens MTi-G is applied as the main sensor unit; this component is an inertial measurement unit (IMU) and an attitude and heading reference system (AHRS). Firstly, instead of this IMU, a VectorNav VN-100 was used. With this IMU, however, we obtained worse results in stabilization, since the linear movement also cause a change in the measured angles. Thus, it was replaced with the Xsens MTi-G, a component widely used in other UAV projects, e.g. [5], [6]. The unit contains triaxial digital devices, namely an accelerometer, a gyroscope, and a magnetometer; further, the unit also comprises a GPS receiver in one package with an external GPS antenna. An internal Kalman filter provides precise orientation estimation with 1 deg RMS roll/pitch and 2 deg RMS heading dynamic accuracy and the accuracy of 2.5 meters CEP in position.

The control board is also equipped with a BMP085 barometric sensor for the purpose of altitude measurement. An ultrasound sensor, which is used at low altitudes (during take-off and landing), is situated on the special PCB. The battery level is determined on the basis of the voltage and current measurement. Regularly available BLDC controllers BL-Ctrl V2.0 provide the continuous current of 35 A for each motor and a very compact design. The controllers communicate with the control board via an I²C bus. It is also possible to mount an IP wireless camera on the construction and broadcast the image to the base station.

C. Microcontroller software

The software for the MCU was written in the embedded C language due to high requirements for the real-time behavior



Fig. 8. UAV control board and the other electronics

and reliability. The software does not use a commercial RTOS but a simple and non-preemptive OS written specifically for this application. To achieve the real-time behavior, a system timer with the period of 100 ns was used; this is the shortest possible period to handle any application task. The application also uses MCU hardware interrupts to handle various asynchronous events.

The interval for the computing of the main quadrotor stabilization (angular rotations stabilization) is 10 ms. At the same period, the data from the IMU/AHRS are read and the actuator speeds updated. The sample period in altitude measurement depends on the used sensor; for example, we thus have the period of 25 ms for the ultrasonic sensor. The implemented control structures for the stabilization are constant, but their parameters can be tuned according to the current UAV configuration.

The application operates in several modes: the *configuration/stop mode*, applied when the UAV is switched on but the actuators are disabled; the *flight remote mode*, used for standard remotely controlled operation; the *signal lost mode*, activated in the flight mode when the signal is lost and the UAV is landing automatically; and the *flight auto mode*, which is ready for future autonomous flying using GPS.

D. Communication

The communication between the robot and the base station is wireless. It is provided by two paired X-Bee PRO S2B RF modules transmitting within the band of 2.4 GHz; the modules use the ZigBee protocol and operate in a transparent mode. In this mode, the 2 devices work as a virtual UART wire. Thus, every UART frame send to the input of one module appears

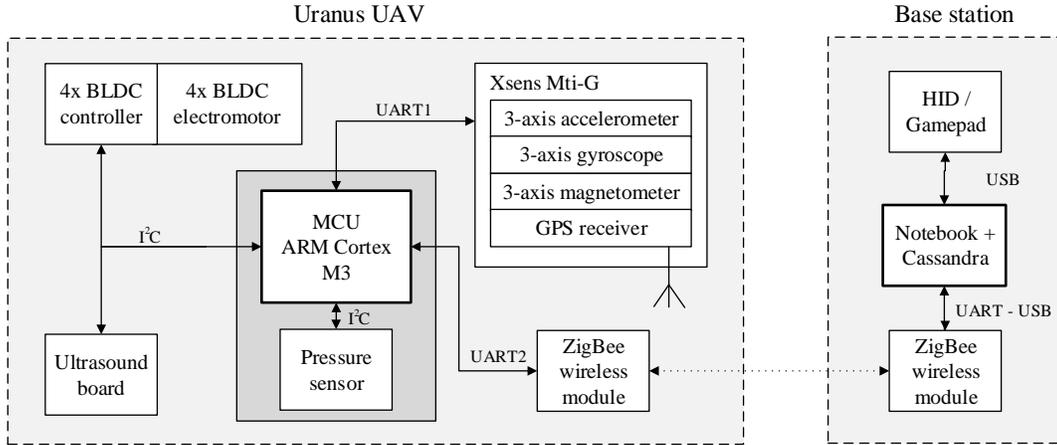


Fig. 7. The block scheme of Uranus UAV system

on the output of the other (if the RF channel is established). The Baud rate of this line is 57.6 kbps.

Above this layer of communication, the UranusLink protocol was created. It was designed as a packet-oriented, non-reliable protocol. This protocol defines the packet structure as well as the transmitted data representation. As shown in Table I, every packet in UranusLink consists of 6 fields: the preamble (PRE); the sequence number (SQN); the message identification (MID); the data length (LEN); the data as such; and the checksum (CS). All the data are in the big endian format.

The preamble has the constant value of 0xFD. It is used to determine the beginning of the packet. The SQN is intended for the numbering of the packets; while every sent packet increases this number by 2, the SQN can only take even values. This property is utilized in one of the packet validation mechanisms. Another important application of the SQN is in packet loss detecting. The MID determines the interpretation of bytes in the DATA part of the message, while the LEN specifies the length of this part. There are 24 different packet types distinguished by the MID. The most important ones are: *Compounded joystick command*, *Robot mode*, and *Acknowledge* in the direction towards the robot; and *All data stream*, *Battery voltage*, and *Acknowledge* in the direction towards the base station. The *Computed joystick command* holds the required roll, pitch, yaw, and thrust. The *Robot mode* is used to switch between the robot modes, as described in Sect. IV-C. This packet is specific, because it must be confirmed by the *Acknowledge* packet from the robot. The *Acknowledge* packet carries, in its data section, the SQN number of the packet which is confirmed by it. Towards the base station, the *All data stream* carries important telemetric data such as the robot attitude and altitude, GPS position, and status (e.g., if the GPS data are valid or the inertial measurement unit was initialized successfully). While these data are sent every 10 ms, the *Battery voltage*, for example, is sent once per second. In the direction towards the robot, the *Compounded joystick command* is sent approximately every 100 ms and the *Robot mode* only when it is required to switch the mode. As stated above, this packet must be confirmed by the *Acknowledge* packet. If it is not, the station begins retransmitting it and continues until confirmation is obtained. This is carried out

TABLE I. URANUSLINK PACKET FORMAT

PRE	SQN	MID	LEN	DATA ...	CS
1B	2B	1B	1B	1-252 B	1B

to ensure that the robot mode will be switched even when the packet is lost. The other packets remain unconfirmed, because the system overheads will be more significant than the advantage of ensuring that all flight data will be transmitted OK.

If there is a packet outage lasting longer than 0.5 s, the robot will handle this as signal loss and take an action. It will set the required roll, pitch and yaw to zero; also, the system performs slow descent and can proceed to eventual landing.

E. Base station

One of the requirements for the Uranus was its integration in the Cassandra system. This system has been developed at CEITEC and the BUT for several years. This system comprises a heterogeneous group of robots and a base station. The core of the base station consists of software written in C#, communication HW, and human-device interface. This interface includes a joystick and a virtual reality helmet, or, in the simpler version, a monitor screen and an optional keyboard. The base station software can be run on a classical PC, a laptop, or an embedded PC; thus, the base station can be a mobile or even wearable system. A new communication HW was built to operate the Uranus UAV. This portable device consists of X-Bee modules (as described in section IV-D), a high-gain antenna, and an UART-over-USB converter, all placed in box attached to a foldable tripod stand to ensure a higher position of the communication system above the ground (and, thus, better coverage).

From the base station, the operator can control all the powered robots by switching among them. The display provides the operator with a first person view from the robot. The robot movements are controlled via the joystick. If the camera on a given robot is capable of rotating, its motion can be controlled via movement of the operator's head (in case that the virtual reality helmet is used) or by means of the joystick.



Fig. 9. Base station and Uranus UAV

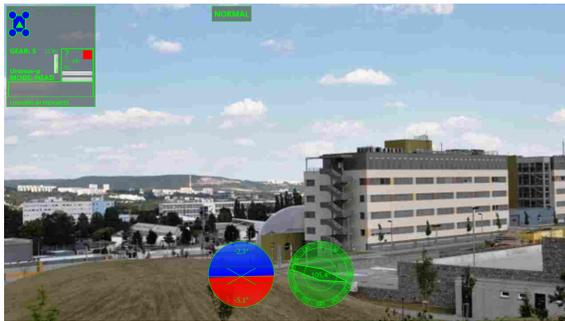


Fig. 10. The user interface of the Cassandra system

The Cassandra base station display provides, besides a view from the onboard camera, important telemetric and other mission-related data, as shown in Fig. 10. The bottom part of the display presents attitude information, namely the actual roll and pitch angle in the artificial horizon in the left circular indicator, and the yaw angle in the compass in the right one. The upper left corner includes the robot status indicator. This contains the name of the active robot, its battery voltage, signal outage indicator, actuator state, robot mode, and other information if required. The lower part of this indicator can show important text information and warnings. All the indicators can be hidden, plus there are couple of indicators not opened by default. Such indicators display, for example, the status of the other robots, the actual position of the robot on the map, or the actual camera orientation.

V. OPERATION

For flying with the Uranus UAV, we use the following configuration: a laptop with the Cassandra system installed; a joypad; and a designed communication device. The Uranus is controlled by a joypad with 2 joysticks, of which the left one regulates the pitch and roll and the right one modifies the thrust and yaw. Before the flight, the following steps need to be carried out: after switching the power on, the Uranus UAV is in the *configuration/stop mode*. This mode has to be switched to the *run mode* from the joypad. Subsequently, all the rotors start and operate at a very low speed. After that, the right joystick, which controls the thrust, needs to be pushed all the way down. Then the rotors will gradually achieve the speed corresponding to the actual joystick position. This solution was implemented to prevent the rotors from starting at very high RPMs if the

mode is switched with the joystick in default position. From this moment on, the Uranus UAV is ready to flight.

If the thrust joystick is released, the robot will take off and hover at about 1 meter above ground. To precisely suppress the gravity force by the thrust, the output levels sent to the motors from the controller are modified according to the battery voltage. This occurs because the thrust of the motors varies in accordance with the battery voltage. The battery voltage level is also reported to the base station to warn the pilot if the voltage drops too low. A warning is also produced by signal loss. If this happens, the Uranus UAV will automatically land.

During the test flights, it was proven that Uranus is stable, and while the control joysticks are released, the quadrotor hovers in position. But because no position stabilization has been implemented to date, the Uranus could slowly drift from the given position. This feedback will be integrated only in the next versions. Controlling the Uranus with the joypad was easy, and the operator was able to operate the UAV effectively after a short while.

VI. CONCLUSION

Regarding the successful test flights, it can be concluded that the assembled mathematical model clearly expresses the quadrotor behavior; thus, the designed control system was able to stabilize the robot suitably. Although the controller was designed as a state-space device, it was proven that the resulting controller can be interpreted as a set of separate PI or PD controllers in each axis. This is due to the model linearization around the working point, namely hovering. For intended use, the described approach provides good results: the quadrotor was able to hover and to maneuver. To facilitate integration in the Cassandra system, a new communication protocol was designed, and the Cassandra system was modified to be capable of handling the flying robots. As the purpose of the developed Uranus UAV was to enable the Cassandra robotic system to perform air missions, it can be stated that the given goal was achieved. For the future, we plan to navigate the Uranus UAV by the GPS, to design our own IMU, and to create a bigger airframe with a higher payload capacity.

ACKNOWLEDGMENT

This work was supported by the project CEITEC - Central European Institute of Technology (CZ.1.05/1.1.00/02.0068) from the European Regional Development Fund.

REFERENCES

- [1] Zalud L, Florian T, Kopečný L, Burian F (2011) Cassandra - Heterogeneous Reconnaissance Robotic System for Dangerous Environments. Proceedings of 2011 IEEE/ SICE International Symposium on System Integration:1-6
- [2] Bouabdallah S (2007) Design and Control of Quadrotors with Application to Autonomous Flying Lausanne EPFL
- [3] Solc F (2010) Modelling and Control of a Quadcopter. Advances in Military Technology 5 (2):29-38
- [4] Lim H, Park J, Lee D, Kim H.J. (2012) Build Your Own Quadrotor IEEE Robotics & Automation Magazine
- [5] Grzonka S, Grisetti G, Burgard W (2012) A Fully Autonomous Indoor Quadrotor IEEE Transactions on Robotics, vol. 28, no. 1
- [6] Alexis K, Papachristos C, Nikolakopoulos G, Tzes A (2011) Model Predictive Quadrotor Indoor Position Control 19th Mediterranean Conference on Control Automation (MED)