2013 IFAC Intelligent Autonomous Vehicles Symposium
The International Federation of Automatic Control
June 26-28, 2013. Gold Coast, Australia

An Autonomous Indoor UAV with a Real-Time On-Board Orthogonal SLAM

Mirco Alpen, Klaus Frick, Joachim Horn [*]

[*] *Helmut-Schmidt-University / University of the Federal Armed Forces Hamburg, Department of Electrical Engineering, Institute of Control Engineering, P.O. Box 700822, D-22008 Hamburg, Germany*
(mirco.alpen@hsu-hh.de, klaus.frick@hsu-hh.de, joachim.horn@hsu-hh.de)

Abstract: Here we present the development and implementation of an innovative real-time on-board orthogonal SLAM (**s**imultaneous **l**ocalization **a**nd **m**apping) algorithm for an industrial quadrotor. The algorithm is discussed particularly with regard to the decision variables and the influences of these variables to the resulting map. The capability of the whole algorithm is demonstrated by experiments. The algorithm delivers a 2D floor plan of the investigated area and the results will show that essential SLAM features like loop closing are covered.
All needed algorithms, microcontrollers and sensors are part of self-constructed **i**ntelligent **c**ontrol **a**nd **m**easurement unit named ICAM. Additionally, all required controllers for an autonomous flight are realized on this unit which is installed below the flight platform. Due to the fact that all needed algorithms for exploration and mapping of an unknown indoor environment run on-board the robot is able to act with full autonomy.

Keywords: Orthogonal SLAM, autonomous flight, indoor UAV

1. INTRODUCTION

Mobile robots in general are used wherever hazardous environments are too dangerous or unacceptable for humans or missions are too time-consuming. In the recent years, exploring and mapping of unknown environments by mobile robots is of increasing interest. To act with full autonomy in an unknown environment a mobile robot must be able to build a map and localize itself in it. The solution of this problem is known as **s**imiltaneous **l**ocalization **a**nd **m**apping (SLAM) and various methods have been proposed and can be divided into two main groups. The first group includes algorithms based on landmarks like Fast-SLAM (Montemerlo et al. (2002)) for example. The other group contains algorithms needing no information about the environment. Therefore, algorithms of this group, like DP-SLAM (Eliazar and Parr (2003)), are suitable to get full autonomy of mobile robots. A general survey of the SLAM history is given in (Durrant-Whyte and Bailey (2006)) and (Bailey and Durrant-Whyte (2006)).

The focus of our work is to enable a quadrotor to explore and map unknown indoor environments with full autonomy. Thus the navigation strategy of the robot is a key feature. Based on this strategy the reference values for the required position and motion controllers are computed. To ensure that all actual values for the control loops are available is particularly for the robot's orientation not that easy in our case. Due to the ferromagnetic materials, working with a magnetometer causes lots of problems. Furthermore the reception of a GPS (**g**lobal **p**ositioning **s**ystem) signal cannot be ensured in an indoor environment. In general an IMU (**i**nertial **m**easurement **u**nit) is used to compute the orientation of a flight robot (Castillo et al. (2005)). To ensure appropriate accuracy of this method, a sample rate around 100 Hz is needed. As said before, we use an industrial quadrotor for our experiments and that is why we are not able to get the needed information with the required sample rate. However, long time stability cannot be ensured with this method. In our application we estimate the robot's orientation based on the data of the optical sensor by computing and evaluating angular histograms. Our method is based on the cross correlation approach presented in (Weiss et al. (1994)) and is discussed in section 3.

An indoor environment often has a predominant rectangular structure. Therefore, the orthogonal SLAM (Nguyen et al. (2006)) is a useful way to build a floor plan with a low memory requirement. We assume that the environment can be represented by lines that are parallel or orthogonal. In our version of the orthogonal SLAM algorithm presented in (Alpen et al. (2010b)) only one step of iteration is needed to integrate a new scan into the global map. This allows an evaluation of the computing time without worst case assumptions and a clear statement for the real-time behavior of the complete system. The used SLAM algorithm and the influences of the decision variables to the cycle time and the resulting map are discussed in section 4.

In general, the mapping of an indoor environment by a flying object is a 3D problem. With the assumption that the flying robot only acts with a constant altitude in a structured indoor environment like corridors above of furniture like chairs or tables, it can be reduced to a 2D problem. The fundamental structure of the explored area will be clear with this restriction. At the end of this paper in section 5 we present some experiments. They show the capability of the combination of a simple motion strategy, the motion controllers and the orthogonal SLAM.

The industrial quadrotor used for our work is produced by a German company named AirRobot®. The small UAV (**u**nmanned **a**erial **v**ehicle) itself is named AR100B®. We equipped this robot with our self-constructed functional group named ICAM (**i**ntelligent **c**ontrol **a**nd **m**easurement unit). All used hardware components are presented in the following section 2.

2. INTRODUCTIVE SYSTEM DESCRIPTION

In this section we present the used hardware. In the first part we introduce the industrial quadrotor. The focus of the second part is on the functional group ICAM.

The industrial quadrotor AR100B® is shown in figure 1. It has a diameter of 1 m and its weight is around 1.2 kg. The robot's working load is denoted with 0.4 kg (Airrobot GmbH&Co.KG (2009)). This quadrotor is by default equipped with an IMU, GPS, magnetometer, barometer and a camera. We replaced the camera by our ICAM presented in the second part of this section.

In general the robot acts in a manual operated mode. The user can control the robot by a commercial hand held transmitter. To ensure safety in our application this hand held transmitter is embedded in the transmission line of the actuating variables. Due to this we can switch to a manual operation mode at any time.

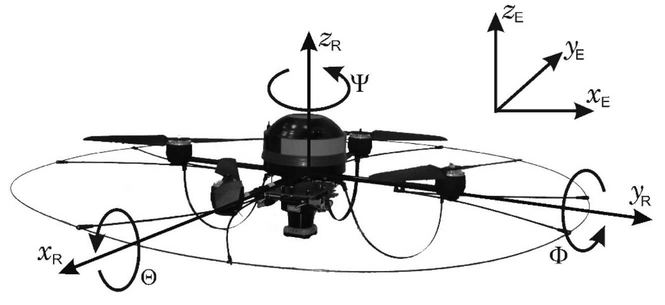

Fig. 1. AR100B® and the two reference frames

As can be seen from figure 1 the robot's orientation is represented by the state Ψ. Because the orientation is not constant during the robot's motion we need to establish two coordinate systems. The coordinates of the robot's body fixed grid are denoted as x_R, y_R and z_R, the world coordinates as x_E, y_E and z_E.

To enable this industrial quadrotor to act autonomously one has to design and implement several controllers. Regarding the unknown on-board attitude control realized by the AirRobot® company, the six degrees of freedom $(x_R, y_R, z_R, \Theta_R, \Phi_R, \Psi_R)$ of an uncontrolled quadrotor (Bouabdallah and Siegwart (2007)) can be reduced to four $({}^{R}x_E, {}^{R}y_E, {}^{R}z_E, \Psi)$. The two indices 'R' and 'E' are used because the robot's position in world coordinates is of intrest for our application. In one of our former papers (Alpen et al. (2009)) we presented a method to model a quadrotor with an unknown on-board attitude control. Regarding this, we give only some basic information about the modeling process in the next section.

The self-constructed functional group ICAM is installed below the robot's body. The ICAM which is shown in figure 2 is composed of a Hokuyo® laser range finder with an effective reach of 30m and two 16bit Infineon® microcontrollers. All time-critical functions are implemented on these microcontrollers. One is used for the autonomous flight with a sampling rate of 5Hz and the other is used for the SLAM algorithm. This algorithm works with a sampling rate of 1Hz. For indoor environments this frequency turns out as a reasonable compromise between accuracy and the amount of data, which is to be kept as low as possible.

The 270 degree laser range finder will be used with an angle resolution of 1 degree. Because it is located at the robot's center regarding the x_R and y_R axes, the origin of the laser beams is located on the z_R axis. This is important for the estimation of the robot's orientation specified in section 3.

Furthermore, the ICAM comprises a WLAN (**w**ireless **l**ocal **a**rea **n**etwork) module to enable monitoring on a ground station. The weight of this functional group shown in figure 2 is 0.45kg and the power consumption is approximately 10W.

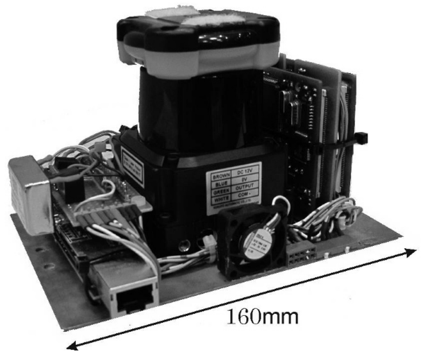

Fig. 2. The functional group ICAM

One of the microcontrollers is connected with the quadrotor via RS232 to receive the current states of the robot (altitude, orientation) and to transmit the corresponding actuating variables. This interface is destined by the producing company of the robot for several applications. As mentioned before a hand held transmitter is embedded in the transmission line of the actuating variables to ensure safety and to enable a manual operated mode at any time.

3. MODELING AND CONTROL OF THE QUADROTOR

In this section some basic information of modeling and control of the industrial quadrotor are given. In the first part of this section the nonlinear dynamical model of the quadrotor with its on-board attitude control is introduced. In the second part the focus is on the needed controllers and the determination of the actual values for the control loops.

Due to the on-board attitude control the nonlinear dynamical model has four inputs $(u_x, u_y, u_z, u_\Psi))$ and four outputs $({}^{R}x_E, {}^{R}y_E, {}^{R}z_E, \Psi)$. Figure 3 shows the structure of the developed dynamical model. The reason for the two indices of the outputs was given in section 2. To compute the actuating variables for the four motors of the quadrotor, one has to switch to the robot's body fixed grid. This transformation depends on the robot's orientation represented by the state Ψ. If the robot acts with a constant orientation such that both reference frames are congruent, this transformation can be neglected.

The modelling of the position in x_{E} and y_{E} plane is summarized in the middle part of figure 3. Among others, because of the air friction F_{air} ${}^{\mathrm{R}}x_{\mathrm{E}}$ and ${}^{\mathrm{R}}y_{\mathrm{E}}$ have to be represented in one dynamical model. The nonlinear function $F(\boldsymbol{u})$ represents the connection between the actuating values u_{x} and u_{y} and the forces F_{x} and F_{y}. This function is based on the connection between the input values (u_{x}, u_{y}) and the corresponding angles (Φ_{R}, Θ_{R}). In comparison to the important dynamics for the control loops the quadrotor adopts theses angles very fast. Thus, this connection can be modeled as static function. Additionally, the gravity constant $g = 9.81\,\mathsf{m/s}^2$ and the mass of the whole system represented by the parameter m have to be taken into account. As mentioned before, the robot's orientation has to be considered as well.

The forces F_{x} and F_{y} leads to the accelerations ${}^{\mathrm{R}}\ddot{x}_{\mathrm{E}}$ and ${}^{\mathrm{R}}\ddot{y}_{\mathrm{E}}$. With regard to the air friction F_{air} one can come to the robot's velocity. The corresponding block is marked as nonlinear because it depends on the squared of the velocity.

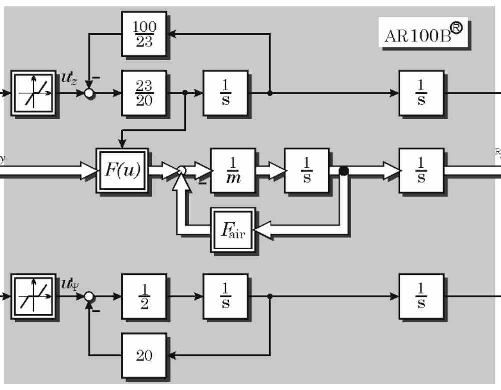

Fig. 3. Structure of the nonlinear dynamical model

As can be seen, there is a connection between the function $F(\boldsymbol{u})$ and the state ${}^{\mathrm{R}}\ddot{z}_{\mathrm{E}}$. As mentioned before, the function $F(\boldsymbol{u})$ depends on the balance of forces and therefore the derivation of the climbing rate has to be considered. If the robot only acts in a constant altitude, this connection can be neglected and the forces F_{x} and F_{y} can be computed by the following equations.

$$
\begin{aligned}
F_x &= \tan(\Phi_{\mathrm{R}}) \cdot m \cdot g \\
F_y &= \tan(\Theta_{\mathrm{R}}) \cdot m \cdot g
\end{aligned} \tag{1}
$$

The part of the model representing the robot's altitude or the motion along the z_{E} - axis is given in the upper part of figure 3. Due to the weight of the ICAM which is slightly higher than the nominal capacity of the robot, this is a nonlinear model, too. The zero position of the actuating values regarding to a positive climbing rate moves from the middle of the operating range towards the upper border. This circumstance is represented by the asymmetric dead-zone. In the lower part of 3 one can find a symmetric dead zone in the rotation around the z_{E} - axis. It is caused by quadrotor setup for the hand operated mode.

To identify and validate the different parts of the model, several flight experiments were done. Within these experiments we ensured, that the robot acts in a way that the couplings between the different parts of the model can be neglected. Measurement data and results of the validation can be seen in (Alpen et al. (2009)). The model presented in this paper is slightly different due to the higher weight of the ICAM. However the modeling was done equivalently here.

Regarding the presented nonlinear model with its interrelationships between the different parts, one could think of the design of a nonlinear multivariable control. As mentioned before, the main focus of this project is on exploring and mapping of the unknown indoor environment. Furthermore, all required algorithms for this application have to run in real-time on a 16bit microcontroller. Thus, one has to find a compromise between a reasonable accuracy of the controllers and the resulting cycle time for the computation of the actuating variables. Because of this, we assumed a decoupled model structure for the controller design. Moreover, the three parts of the model have been linearized around a reasonable operating point for the given application. Based on these simplifications the controllers were synthesized based on classical design techniques.

The actual values for the control of the robot's altitude are given by a barometric sensor. The position data on the x_{E} - axis and the y_{E} - axis are given by the laser range finder. The robot's heading is estimated based on this data, too. In the first step an angular histogram of the current set of measurement values is computed. Regarding a predominant rectangular structure of the environment, such a histogram has typically four maxima. With information about the robot's initial orientation one can estimate the robot's orientation at each time by comparing the current angular histogram with an ideal angular histogram which consists of four Dirac functions for the whole 360 degree cycle. Therefore, the computation of a cross correlation is not necessary because the Dirac function $\delta(t)$ is the neutral element of the correlation. More information about this method is given in (Alpen et al. (2012)). The accuracy of the angle estimation is within ± 1.5 degree. In relation to the full circle with 360 degree, this leads to an error below 1%. Thus, this method to estimate the robot's orientation is quite successful and it is used for control and the SLAM algorithm in the following experiments. A simple linear controller is used and the resulting closed loop system has an adequate accuracy for our application.

4. INNOVATIVE ORTHOGONAL SLAM

The basic idea of our orthogonal SLAM algorithm was presented in (Alpen et al. (2010b)) and is shown in figure 4. Thus, some parts are only discussed briefly in this section. The focus will be on the important decision variables and their influences to the cycle time and the resulting map.

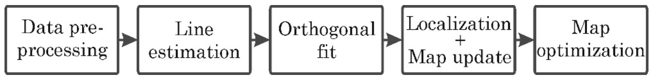

Fig. 4. Basic flowchart of SLAM algorithm

Figure 4 shows that our algorithm comprises five steps. Within the first step named 'Data preprocessing' erroneous measurements of the laser range finder are excluded. The measurement values are given in polar coordinates in the robot's body fixed grid and transformed into the world coordinates regarding the robot's orientation Ψ. The way how we estimate the orientation is described at the end of the previous section 3. During the second step the set of measurement values

$$P_\mathrm{v} = \{d_1,\ d_2\ \dots\ d_n\}, \tag{2}$$

where n is number of valid scan values, is divided into several clusters

$$P_\mathrm{v,S} = \{S_1,\ S_2\ \dots\ S_m\}. \tag{3}$$

For this separation, the decision variable q is needed. If the Euclidean distance between two consecutive scan values d_i and d_{i+1} is larger the parameter q a new cluster S_i is separated from the set of measurement values P_v. The choice of this constant q is quite important regarding the cycle time of the SLAM algorithm. The larger a cluster is, the longer it will take to estimate the corresponding lines afterwards. But if the clusters get to small, there will be a lot of very short lines and the map update, which is explained later on in this section, might be time consuming. Thus one has to find a compromise between those two cases, to ensure a small cycle time. In our application it is set to $q = 0.3\,\mathrm{m}$. However, each segment represents a coherent structure of the environment. Figure 5 shows a small example with four segments.

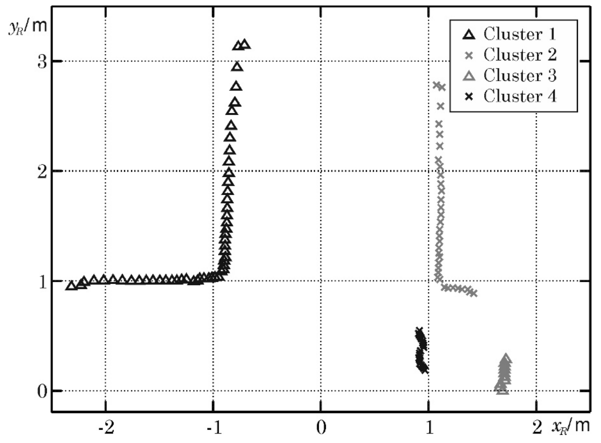

Fig. 5. Set of scan values P_v divided in clusters S_1 to S_4

In the given example of figure 5, the laser range finder is located at the origin. After the separation lines are estimated within each cluster based on the split and merge algorithm presented in (Choi et al. (2008)). For this procedure the choice of two threshold values is important. The first one belongs to the splitting. If the Euclidean distance between two consecutive scan values within a segment is lager than $0.05\,\mathrm{m}$, the current data set is splitted. The second one regards the merging process. Two lines are merged, if the distance of their touch point to the direct line from the beginning of the one line two the end of the second line is below $0.05\,\mathrm{m}$. These two values have a lasting effect of the map accuracy and complexity. The smaller theses values are, the more complex and accurate the map will be. As mentioned before, a larger complexity leads to a higher cycle time of the algorithm due to the map update. Thus, one has to find a tradeoff between a reasonable accuracy and cycle time at this point , too. The resulting lines for this example are the grey ones in figure 6.

These lines are checked for orthogonality within the next step. Extracted lines that are slightly non-orthogonal are changed to orthogonal ones. All other lines are neglected and will not be part of the global map. For this orthogonal fit the parameter $\Delta\phi$ is defined. This parameter gives the maximum angle tolerance to an orthogonal line. It is not a constant value, because it is useful to take the length of the certain line into account. The longer a line is, the smaller the angle tolerance $\Delta\phi$ should be. It is not sufficient to have a constant angle tolerance $\Delta\phi$. If it is too small, too many lines are neglected and if it is too large, the map gets inaccurate. Thus, we choose the function given in figure 7. The longer a line is, the more scan values d_i are represented by the line. Thus, this information is used for the computation of the angle tolerance $\Delta\phi$.

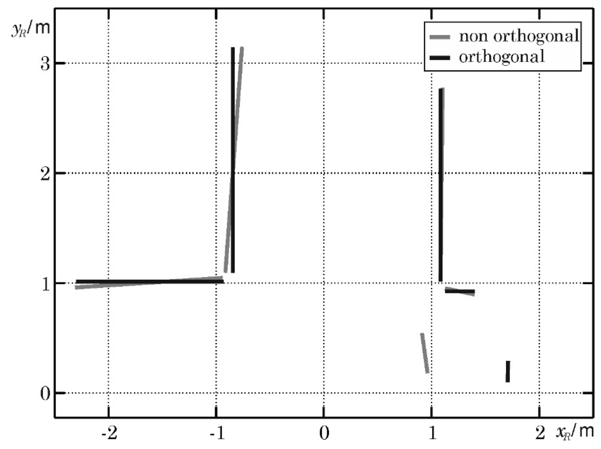

Fig. 6. Extracted lines from the current dada set P_v

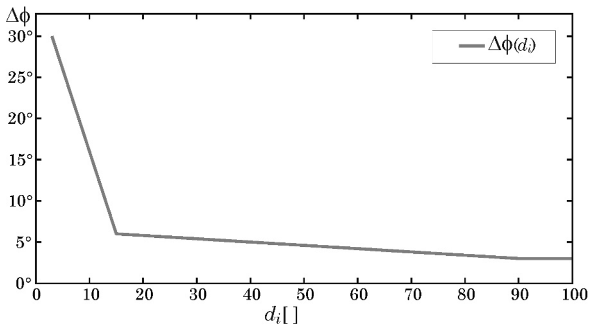

Fig. 7. Function for the choice of the angle tolerance $\Delta\phi$

For the given example there is one grey line in the lower right corner of figure 6 that failed the orthogonal fit. Thus, this set of measurement values P leads to five orthogonal lines which have to be integrated in the global map. This happens within the fourth part of the SLAM algorithm. To integrate the new lines successfully one has to estimate the robot's motion accurately. In several applications this is done by the ICP (**i**terative **c**losest **p**oint) algorithm or the PSM (**p**olar **s**can **m**atching) (Diosi and Kleeman (2005)). In our case this is done by just one step of iteration. As mentioned before the data processing is done considering the robot's orientation Ψ. Therefore, the orthogonal lines are all parallel to one of the axes of the world coordinates x_E or y_E fixed based on the first scan and the calculation of the robot's motion can be divided in these two directions. For each direction all displacements within the bounds of possibility are computed. The choice of the corresponding threshold values depends on the robot's velocity. Afterwards an accumulation point is extracted from the set of displacements. Based on this accumulation point the robot's motion is estimated. In (Alpen et al. (2010b)) we showed that this procedure is quite accurate.

In the last part of the algorithm named 'Map optimization' double lines are merged to reduce the total number of lines in the global map. For this procedure the choice of the demand threshold values is important for the same reason given in connection with the split and merge topic. If two lines are parallel and the lack of overlapping and their displacement is below 0.1 m they are merged in our case. With this step the computing time of the following map update and the memory requirements are reduced.

5. EXPERIMENTAL RESULTS

In this section we present some experimental results of the whole system consisting of the industrial quadrotor AR100B® and the self-constructed functional group ICAM. To show the performance of the system we used the sports hall of our university. Within this hall we are able to build up different scenarios with movable walls. Thus we can ensure a test environment free of dynamic obstacles. Figure 8 shows the quadrotor in this test environment. In the background of this picture one can see the movable walls. During the experiments, the robot was connected to a ground station via WIFI for data monitoring. All essential algorithms for the SLAM and the robot's movement were running on-board.

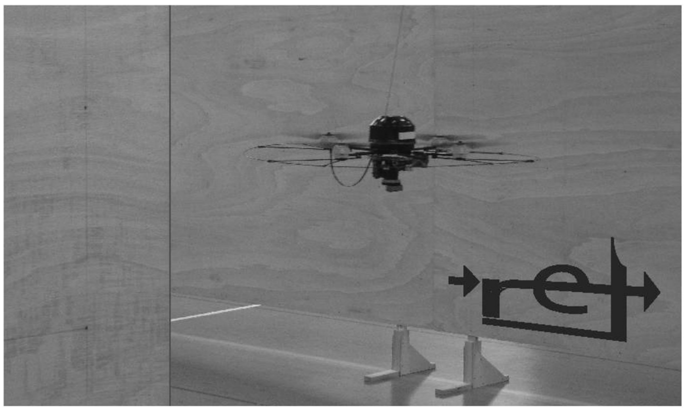

Fig. 8. Flight robot in the test environment

To explore an unknown environment, the robot needs a certain motion strategy. In these experiments we choose a very simple strategy. The robot has to follow the wall on the left hand side. If there is an opening in the leading wall caused by another corridor or an open door, the robot should turn if the opening is wide enough. Due to the mentioned drawbacks of the industrial quadrotor related to the performance of the implemented controllers, a corridor should have at least a width of 4 m. The used strategy of motion is very simple and might be quite time consuming, if the environment has a complex structure. For ground robots with a lot more of computing power we have presented a strategy which is much more time efficient (Alpen et al. (2010a)). However, within these experiments the focus is on the real-time on-board SLAM algorithm. Thus, we decided to use this simple strategy.

In the first test scenario the robot has to move through an approximately 4 m wide corridor with an angled characteristic. The grey lines in figure 9 represent the corresponding floor plan. The whole scenario has a surface of approximately 150 m^2. The robot starts at the origin of figure 9 and should move with constant velocity and altitude through the environment.

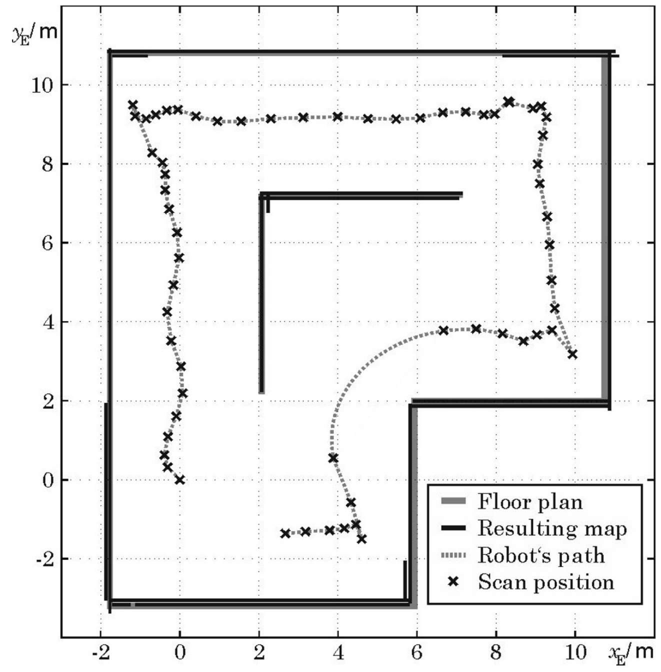

Fig. 9. Result of the on-board orthogonal SLAM

The black lines in figure 9 represent the resulting map. Due to the reasonable choice of the described decision variables the robot produced a very well arranged map. Otherwise the algorithm might give much more lines or less accuracy. The worst case would be the termination of the algorithm during the mission. One can see that the loop-closing problem which is an important feature of SLAM-algorithms is solved with this algorithm. Furthermore, this result in figure 9 demonstrates that the way we estimate the robot's orientation works quite well, too. The grey line with the black marks represents the robot's way through this environment. Each scan point which is considered for the map is marked with a black cross. Based on the distribution of these marks one can see that the robot moved with a more or less constant velocity except at the corners. If the robot turns on the spot or around a corner, the SLAM algorithm is paused. Thus there are no marks in the lower right part of figure 9.

In the second experiment the setup of the environment is slightly more complex. Here we have a corridor with another diverging corridor to the left ending up in a room. The grey lines in figure 10 represent the corresponding floor plan. In this scenario the robot should move into the room on the left hand side and afterwards it should continue its way along the corridor.
Regarding the dashed grey line representing the robot's way through the environment, one can see that the goal of this experiment was reached. The distance to the left hand wall is not as constant as in the previous experiment. The robot has to move closer to the wall and this causes disturbances regarding the rotating blades and the resulting air flow. The black crosses on this line of figure 10 give the scan positions of the measurements considered

in the map represented by the black lines. The scaling of the on-board map is quite accurate but there are several double lines. Nevertheless, the structure of the investigated environment is visible without any doubt. Thus the current setup for the decision variables is sufficient for an structured indoor environment.

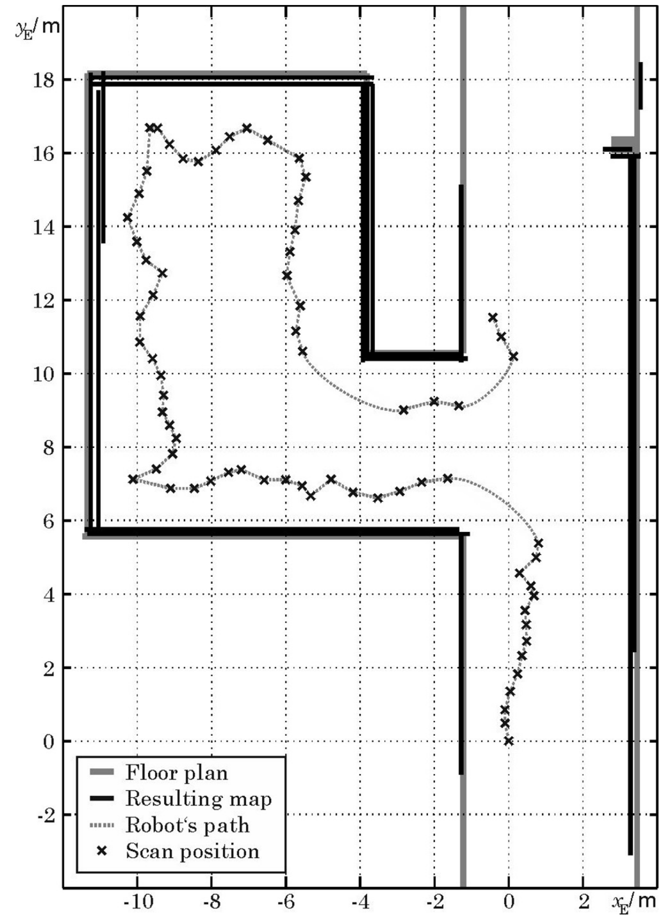

Fig. 10. Result of the on-board orthogonal SLAM

6. CONCLUSION

In this paper we presented the implementation of a real-time on-board orthogonal SLAM algorithm for an industrial quadrotor with regard to the important decision variables. All needed algorithms, microcontrollers and sensors are part of our self-constructed ICAM. Therefore, the robot was able to act with full autonomy in an unknown indoor environment. The capability of the whole system was demonstrated by two experiments. In each experiment the algorithm delivers a 2D floor plan of the investigated area with a quite accurate scaling. In both experiments the loop closing as an essential SLAM feature was covered as well. Furthermore we showed that the used setup for the decision variables worked sufficiently in indoor scenarios.

In the near future we will use a self-constructed quadrotor. The knowledge of the attitude controllers of this vehicle will lead to a huge improvement of the controllers needed for the autonomous flight. Thus, the exploring of areas with more narrow corridors will be possible. Furthermore, we want to implement a navigation strategy which leads to a more intuitive way of exploring unknown indoor environments.

REFERENCES

Airrobot GmbH&Co.KG (2009). *AR 100B - Operation Instructions.* Airrobot GmbH&Co.KG, Arnsberg, Germany.

Alpen, M., Frick, K., and Horn, J. (2009). Nonlinear modeling and position control of an industrial quadrotor with on-board attitude control. *IEEE International Conference on Control and Automation (ICCA), Christchurch, New Zealand*, 2329–2334.

Alpen, M., Frick, K., and Horn, J. (2012). A real-time on-board orthogonal slam for an indoor uav. *5th International Conference in Intelligent Robotics and Applications (ICIRA), Montreal, Canada.*

Alpen, M., Stuedemann, S., and Horn, J. (2010a). Time efficient strategy to explore unknown indoor environments by mobile ground robots. *3th International Conference in Intelligent Robotics and Applications (ICIRA), Shanghai, China.*

Alpen, M., Willrodt, C., Frick, K., and Horn, J. (2010b). On-board slam for indoor uav using a laser range finder. *SPIE Defence, Security and Sensing, Orlando, USA,.*

Bailey, T. and Durrant-Whyte, H. (2006). Simultaneous localization and mapping (slam): part ii state of the art. *IEEE Robotics & Automation Magazine*, 13(3), 108–117.

Bouabdallah, S. and Siegwart, R. (2007). Full control of a quadrotor. *International Conference on Intelligent Robots and Systems, IEEE/RSJ, San Diego, USA*, 153–158.

Castillo, P., Lozano, R., and Dzul, A.E. (2005). *Modelling and control of mini-flying machines.* Advances in industrial control. Springer, London.

Choi, Y.H., Lee, T.K., and Oh, S.Y. (2008). A line feature based slam with low grade range sensors using geometric constraints and active exploration for mobile robot. *Autonomous Robots*, 24(1), 13–27.

Diosi, A. and Kleeman, L. (2005). Laser scan matching in polar coordinates with application to slam. *International Conference on Intelligent Robots and Systems, IEEE/RSJ, Edmonton, Canada*, 3317–3322.

Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and mapping: part i the essential algorithms. *IEEE Robotics & Automation Magazine*, 13(2), 99–110.

Eliazar, A. and Parr, R. (2003). Dp-slam: Fast, robust simultaneous localization and mapping without predetermined landmarks. *18th Int. Joint Conf. on Artificial Intelligence (IJCAI), Acapulco, Mexico*, 1135–1142.

Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. (2002). Fastslam: A factored solution to the simultaneous localization and mapping problem. *Proceedings of the AAAI National Conference on Artificial Intelligence, Edmonton, Canada.*

Nguyen, V., Harati, A., Martinelli, A., Siegwart, R., and Tomatis, N. (2006). Orthogonal slam: a step toward lightweight indoor autonomous navigation. *International Conference on Intelligent Robots and Systems, IEEE/RSJ, Beijing, China.*

Weiss, G., Wetzler, C., and von Puttkamer, E. (1994). *Keeping Track of Position and Orientation of Moving Indoor Systems by Correlation of Range-Finder Scans.* International Conference on Intelligant Robots and Systems (ICRA).