Validation of a Limit Ellipsis Controller for Rescue Drones

Balint Varga[1], Christopher Doer[1], Gert F. Trommer[1] and Sören Hohmann[1]

***Abstract*— In recent years, more and more robotic systems have been supporting rescue forces in their missions. This paper presents the control algorithm and its application of an unmanned areal vehicle (UAV), which can support emergency personnel in their work. An adaption of the so-called Limit Ellipsis Controller (LEC) for indoor UAV is proposed. This adaptation enables the LEC to be used in semi-structured environments with static and dynamic obstacles. The main benefit of the LEC is that it can prevent deadlocks caused by other methods in complex environments. Furthermore, the LEC function is implemented on an experimental UAV system and tested in various environments. With our technical system, the conventional search of a building by a human can be replaced by the semi-autonomous UAV, saving valuable rescue time. The UAV can fly into the building and explore the interior without collision. The results show that the proposed controller can adequately avoid local minima, guide the UAV to the desired target and provide essential information for the rescue team in real demonstration scenarios.**

I. Introduction

In recent years, the use of unmanned areal vehicles (UAV) in various applications has become popular, see e.g. [1] or [2] for a general overview. One of the most promising applications is the use in life-critical situations, e.g. fire service, flooding area, etc. For example, a fire brigade usually has to enter burning buildings filled with smoke or with toxic gases (see [3]). In order to prevent danger, a UAV is able to fly into the building to explore the scene and determine if lives are in danger. Such an UAV can quicken the rescue operation, saving time and lives. An exploration requires an autonomous UAV, as not every firefighter can manage to fly a drone, and first-person view control has many limitations that can easily damage the drone.

Therefore, such drones need to navigate fully or semi-autonomously in the unstructured environment. Camera-based systems are not suitable for use in darkness or smoke. For this reason, the use of laser scanners is proposed in our work, see [4].

Due to the unstructured environment, a robust controller for the rescue UAV is necessary [5, Ch. 47]. One possible approach is the Limit Ellipsis Controller (LEC), which is originally presented for ground robots in [6]. Another benefit of the LEC is that local minima are handled and deadlocks can be avoided even in unstructured environments. An adaptation to UAVs is presented in [7], [8] and [9], in which only verifications in simulations are addressed, but no real flights.

[1]Balint Varga, Christopher Doer, Gert F. Trommer and Sören Hohmann are with Institute of Control Systems, Karlsruhe Institute for Technology, 76131 Karlsruhe, Germany `balint.varga2@kit.edu`

The contributions of this paper are the following:

1 The presentation of the overall UAV system suitable for rescue missions,
2 a novel adaptation of the LEC for indoor drones and
3 the validation of the control concept with rescue professionals in an experimental setup with different realistic scenarios operating in a semi-structured environment.

One of the main focuses of the scenarios is the collision-free navigation in a dark warehouse. In this scenario, the uses of vision-based navigation and corresponding controllers is not possible.

This paper has a structure as follows: Section II presents the state-of-the-art and the literature related to indoor UAVs. In Section III, the adaptation of the LEC method and the challenges of the parameter design are presented, which is crucial reaching for a smooth motion of the UAV. This is followed by the presentation of the UAV in Section IV. The verification of the control algorithm in simulation is discussed in Section V. The validation on the real UAV in different test environments and the results are given in Section VI. Finally, Section VII summarizes this paper.

II. Related Works

A. Obstacle Avoidance Algorithm for UAVs

Obstacle avoidance algorithms are used to navigate autonomously robots between obstacles without collision and to reach the desired goal of the robot [5, Ch. 47]. In [10], different categorizes are distinguished: geometry-based methods, optimization-based approaches, artificial potential field (APF) and visual avoidance methods. Vision-based approaches uses optical flow to determine the motion and the position of the UAV. For their application good light

Fig. 1. A picture of the UAV system used in this work

conditions are inevitable. The optimization-based approaches set up a cost function, which can minimize e.g. energy consumption, the maneuver time e.g. [11], distance to obstacle or the path length e.g. [12]. Optimization-based approaches require a detailed map of the environment to be able to construct the cost function. A geometric approach usually takes more than one UAV into account to compute and evaluate their paths and avoid collisions, see e.g. [13]. APF methods use an artificial force field, which attracts (e.g. the goal) and repels (e.g. obstacles) the UAV. The resulting force governs the UAV between the obstacles towards its goal.

B. Obstacle Avoidance with Artificial Potential Field Method for Robotic System

In this subsection, general APF methods are discussed more in detail as APF methods suits the best for our rescue scenario. An obstacle avoidance method with APF is presented in [14] for the first time. The benefit of this concept is its fast computability and general use for obstacles with different form. In [15], APF concept is applied to ground robots. The main problem of the APFs is the possibility of local minima through the form and configuration of the obstacles (e.g. L- or U-form). To this end, in [6], the basic idea of the limit ellipsis controller (LEC) method is introduced. The core idea of the LECs is the introduction of circles or ellipses, which surrounds the obstacle and using these cycle to bypass the obstacle, see Fig. 2 with the blue attractors and the red ellipsis. In that way, locale minima and deadlocks can be omitted. The LEC was applied for UAV in [7], in which the LEC was extended for 3 dimensional flights. However, the concept is only verified in simulations, which does not necessarily prove the robustness of the controller.

In [16], a laser based path planning algorithm for UAVs is presented, in which the authors focused on the navigation and not on the controller problem of the system. Furthermore, the challenge of the local minima is not addressed. Consequently, in the literature, there is no work presenting LEC methods with real UAV flights.

III. Adaptation of Limit Elliptic Navigation Controller

This section first presents the general approach of the LEC. The LEC is solely suitable for obstacle avoidance, therefore it is combined with an attractive APF. Finally, our adaptation and extension of the LEC-APF is proposed.

A. Obstacle Detection - Elliptic Approach

Classical APF uses attractive and repulsive potentials to avoid obstacles. The use of LEC requires a combination with attractive potentials to reach the goal. LEC is applied instead of the repulsive potentials.

The obstacle detection happens with a 2-D laser scanner. The sensor scans the environment and provides a 2D point cloud. Instead of using the sum of the sample points of the obstacles for the computation of the UAV's trajectory, the detected points are clustered in groups. For each group an ellipsis is identified. This ellipsis is an attractor for the motion of the UAV enabling a collision-free path. Fig. 3 is a schematic illustration of one detected obstacle and the identified ellipsis (yellow ellipsis).

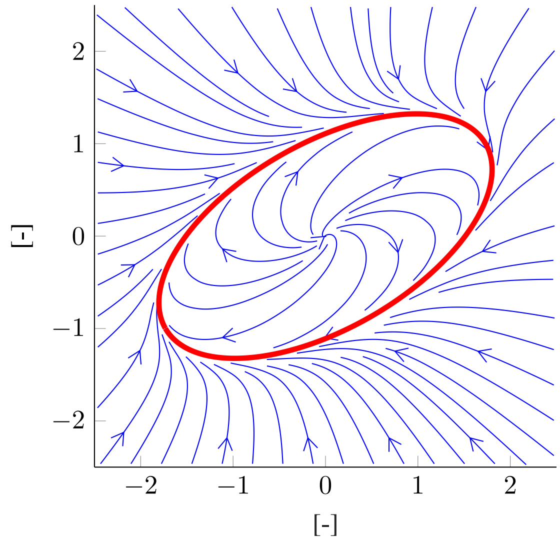

Fig. 2. Example of a limit cycle and the blue attractors. Crucial for a smooth motion is course of these blue attractors, which can be adjusted by a proper parametrization

To reach the desired goal, an attractive force pulls the UAV to its goal, cf. Subsection III-B. Fig. 3 also shows the shortest way between the start and goal point in blue, which would lead to a collision. At the beginning, the UAV moves along a straight line toward the goal due to the attractive force of the APF. Approaching the obstacle, the limit cycle is enabled, and the UAV flies toward the cycle (cf. Fig. 2).

In [9], the clustering algorithm is presented, which is applied in this work. Therefore, in the following, the key aspects of the ellipsis computation and the controller design are only presented.

An ellipsis has the mathematical equation

$$A\cdot(x-x_{0e})^2+B\cdot(y-y_{0e})^2+C\cdot(x-x_{0e})\cdot(y-y_{0e})=1, \quad (1)$$

where, x_{0e} and y_{0e} are the centre points of the cycle. The parameters A, B and C are computed as

$$A = \frac{\sin^2(\varphi)}{a^2} + \frac{\cos^2(\varphi)}{b^2} \quad (2a)$$

$$B = \frac{\cos^2(\varphi)}{a^2} + \frac{\sin^2(\varphi)}{b^2} \quad (2b)$$

$$C = \left(\frac{1}{b^2} - \frac{1}{a^2}\right) \cdot \sin(2\varphi), \quad (2c)$$

where a, b and φ are the major axis, the minor axis and the tilt angle of the ellipsis, respectively, cf. Fig. 3.

For the navigation, an additional safety range (Δ) ensures an adequate distance to the obstacle $a_{\text{nav}} = a + \Delta$ and $b_{\text{nav}} = b + \Delta$, which is set in (2).

This safety range is necessary due to the uncertainties of the navigation and due to the time delays of the low level controller and the navigation, see Section IV-C.

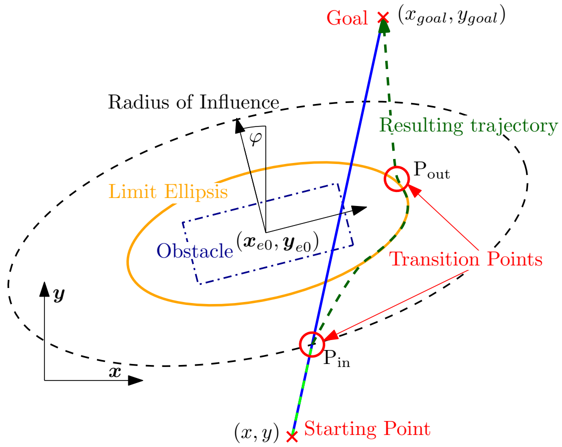

Fig. 3. Schematic Illustration of the Limit Ellipsis, its radius of influence and the transition points between the APF and the LEC controllers

This ellipsis provides the trajectory for the motion of the UAV, where the velocity components are

$$\begin{aligned}\dot{x}(t) =& p \cdot (By + 0.5 \cdot Cx) \\ &+ \gamma \left(1 - Ax^2 - By^2 - Cxy\right) \qquad (3a)\\ \dot{y}(t) =& - p \cdot (Ay + 0.5 \cdot Cx) \\ &+ \gamma \left(1 - Ax^2 - By^2 - Cxy\right), \qquad (3b)\end{aligned}$$

where $p = \pm 1$ depending on the flying direction of the UAV around the obstacle. The parameters are computed from the clustering algorithm. They are continuously updated during the flight. The design parameter γ is used for the shaping of the attractive trajectories toward the ellipsis. Smaller γ lead to a smoother heading while larger values generate more dynamic trajectories.

B. Attractive Artificial Potential Field

To reach the goal of the UAV, an attractive APF is specified and combined with the LEC. The potential of the APF is chosen to

$$U(t) = k_{\mathrm{lin}} \cdot \|\boldsymbol{d}(t)\|^{k_{\mathrm{exp}}}, \qquad (4)$$

where k_{lin} and k_{exp} are the linear and the exponential design parameters, respectively. The length of vector $\|\boldsymbol{d}(t)\|$ is the distance between the actual (x, y) and the goal (x_g, y_g) positions, cf. Fig. 3. The resulting acceleration profile is

$$\boldsymbol{a}_{\mathrm{APF}}(t) = k_{\mathrm{lin}} \cdot k_{\mathrm{exp}} \cdot \boldsymbol{d}(t)^{k_{\mathrm{exp}}-1}, \qquad (5)$$

which provides the guiding velocity of the UAV

$$\tilde{\boldsymbol{v}}_{\mathrm{APF}} = \int \boldsymbol{a}_{\mathrm{APF}}(t)\,\mathrm{d}t. \qquad (6)$$

The parameters k_{lin} and k_{exp} need to take into account the physical limits of the system. They have to ensure smooth starts and stops of the UAV. Furthermore, the overall aim of the mission is the broad exploration of the indoor environment. Therefore, the maximal velocity is limited by

$$\boldsymbol{v}_{\mathrm{APF}} = \max\left(\|\tilde{\boldsymbol{v}}_{\mathrm{APF}}\|;\ v_{\mathrm{max}}\right) \cdot \frac{\tilde{\boldsymbol{v}}_{\mathrm{APF}}}{\|\tilde{\boldsymbol{v}}_{\mathrm{APF}}\|}. \qquad (7)$$

The maximal velocity of the UAV, v_{max} has to be chosen in accordance with the LEC parameters, see Subsection C.

The desired orientation is computed from the direction of the guiding velocity vector $\boldsymbol{v}_{\mathrm{APF}}$:

$$\theta_{\mathrm{UAV,d}} = \arctan\left(x_{\mathrm{g}}, y_{\mathrm{g}}\right), \qquad (8)$$

where x_{g} and y_{g} are the x and y components of the guidance velocity, respectively. A P-controller

$$\dot{\theta}_{\mathrm{UAV,d}} = -k_{\mathrm{rot}}\left(\theta_{\mathrm{UAV}} - \theta_{\mathrm{UAV,d}}\right) \qquad (9)$$

sets the actual orientation of the UAV θ_{UAV}, which enables the exploration of the indoor environment.

C. Adaptation of the transition points

In contrast to [8] and [17], this paper does not use a hierarchical controller structure switching between the APF and LEC, which provides a more simple software architecture and an easier integration of inputs or commands of a user of the system. The LEC and attractive APF both provide the guidance velocity v_{guid} of the UAV. For a smooth transition between APF and LEC, their parametrizations are crucial, which is not handled in the literature. The low level control receives this guidance velocity v_{guid} to control the UAV (cf. Section IV-C).

In our work, a good transition between these two controllers is ensured by a systematic parameter design. This means the following:

- The radius of influence to the active obstacle (r_{RoI}) is chosen in accordance with the maximal velocity:

  $$r_{\mathrm{RoI}} \geq \frac{v_{\mathrm{max}}}{t_{\mathrm{break}}}, \qquad (10)$$

  where t_{break} is the time to stop the UAV from the maximal velocity.
- The LEC at the transition point $\mathrm{P_{in}}$ should have the same velocity as the APF does

  $$\boldsymbol{v}_{\mathrm{LEC}}^{\mathrm{P_{in}}} = \boldsymbol{v}_{\mathrm{APF}}. \qquad (11)$$

  This transition point velocity is the result of (3). Larger r_{RoI} leads to an earlier transition between APF and LEC. It is assumed that the UAV flies in most of the time with the maximal velocity v_{max}, which is used for the computation of the transition points.
- The exit transition point $\mathrm{P_{out}}$ is chosen according to the longitudinal distance to the active obstacle. If the UAV passed the obstacle, the new velocity is computed by

  $$\tilde{\boldsymbol{v}}_{\mathrm{APF}} = \boldsymbol{v}_{\mathrm{LEC}}^{\mathrm{P_{out}}} + \int \boldsymbol{a}_{\mathrm{APF}}(t)\,\mathrm{d}t, \qquad (12)$$

  to ensure a smooth transition between LEC and attractive APF.

IV. Technical System

This section gives a short overview of the technical system consisting of the mission ground station and the hardware and software setup on the UAV.

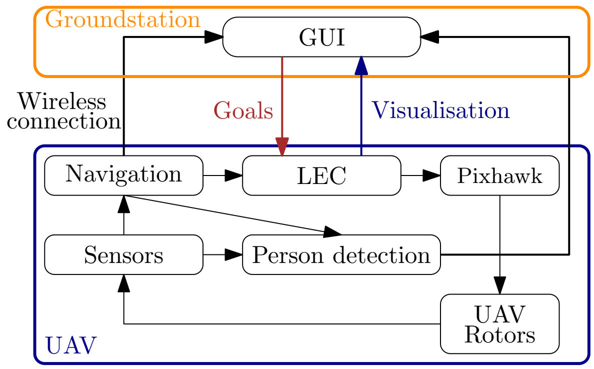

Fig. 4. The software structure of the system. All the navigation and control algorithms run on onboard computer of the UAV. The Groundstation is used for visualization alone.

A. Rescue Mission Ground Station

The technical system has two main components: The mission Groundstation (GS) and the UAV, see Fig. 4. The communication between the GS and the UAV happens via WLAN. Using the GS, the operator can interact with the UAV. The operator can choose goal points with the graphical user interface (GUI), see lower left in Fig. 5. These points are reached by rescue personnel by clicking on the map. Between the goal points, the UAV flies autonomously avoiding collision with static and dynamic obstacles as shown in the Section III. The operator sees the actual states of the sensor and controller subsystem on the left side of the GUI. The UAV is able to fly into an unknown area, explore it and provide map information about it. Furthermore, a person detection is implemented on the UAV, which supports the operator to find persons in the rescue area. The detected obstacles and the ellipses computed by the LEC-APF are shown in different colors on the GUI supporting the operator to choose the goals properly.

B. Hardware Components of the UAV

The UAV consists of different hardware elements:

- Sensors
  - Inertial Measurement Unit (IMU),
  - A 2-D laser scanner for the detection of the obstacles and for online mapping,
  - A barometer for the estimation of the absolute altitude,
  - A laser distance sensor for the relative altitude measurement,
  - Global Positioning System (GPS)
  - digital compass
  - A thermal camera
  - A conventional RGB camera
- A custom microcontroller for reading and synchronizing sensor data
- A Next Unit of Computing (NUC) bare bone mini PC from Intel for onboard processing of all algorithms and software modules
- The Pixhawk controller driving the rotors.

C. Software Structure of the UAV

The software components are depicted on Fig. 4. All the computations (sensor data fusion; detection and avoiding obstacles; creating online map; and neuronal network for the person detection) run on the NUC of the UAV. The GS is solely used for visualization of the generated map and for handling user interactions. The following software components run on the NUC:

1) Online Navigation: The navigation running on our UAV is presented earlier in [4] and in [18]. This navigation enables seamless indoor-outdoor and outdoor-indoor transitions. This means the estimated trajectories are smooth and

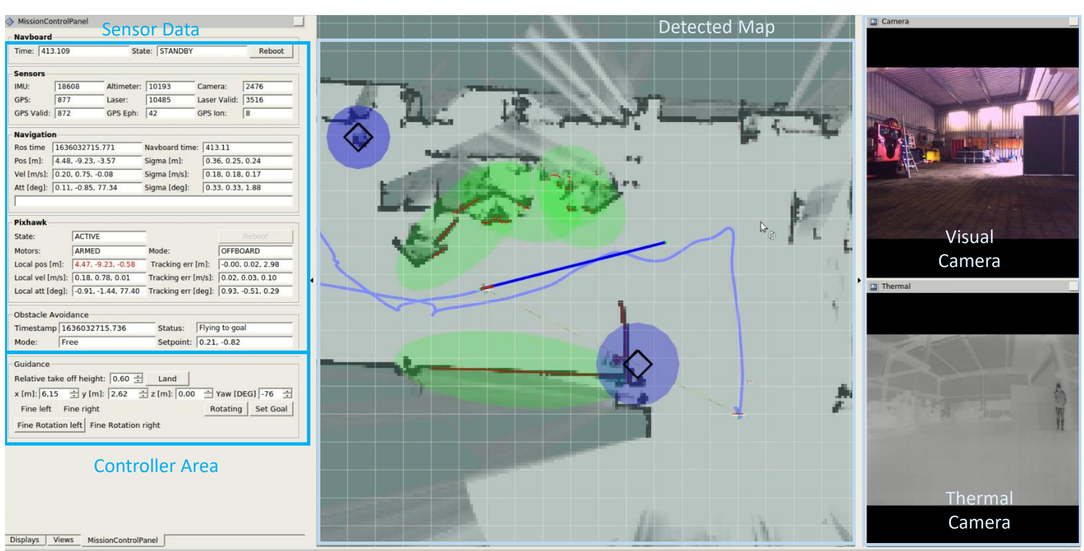

Fig. 5. Screenshot of the Groundstation GUI. On the left side, the sensor data (Navboard, Sensors, Navigation, Pixhawk), the states of the obstacle avoidance and the control user interface can be seen. In the middle, the online map of the indoor, the detected ellipses and the path flown by the UAV are shown. The pictures of thermal camera and the RGB-camera are located on the right side of the GUI

do not have sudden changes even in such transition phases. The navigation system fuses inertial data with laser scan data and, if available, GPS measurements using an Extended Kalman Filter. The navigation filter provides state estimation at IMU rate, which is 205 Hertz in our case.

2) Limit Ellipsis Controller: For a collision-free flight, the LEC proposed in Section III is implemented on the NUC.

3) Person Detection: To support the rescue team, a person detection using an artificial neuronal network which runs also on the NUC. To achieve real-time detection and to take computational load of the Intel NUC, inferencing of the neural net is done with dedicated hardware. Specifically, we used the Edge TPU USB Stick from Coral.

4) Low-Level Controller, Pixhawk: The Pixhawk PX4 Drone Autopilot, Professional Open Source Autopilot Stack [19] software is used to control the rotor speeds of the UAV. Pixhawk enables a stable low-level control of the system, which accelerates the development of autonomous functions.

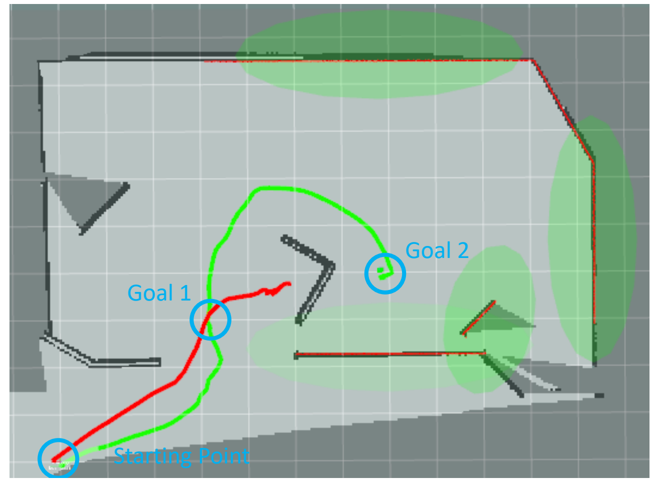

Fig. 6. Picture of the simulation results in Rviz showing the map of the room and the resulting motions with the classical APF (red line) and the proposed LEC-APF method (green line)

V. Simulation

In this section, the proposed LEC-APF concept is analyzed and compared with a classical APF concept in simulations first. The classical APF uses attractive and repulsive potentials.

A. Scenario and Controller Setup

The chosen simulation framework is Gazebo [20], [21], in which a realistic model of our UAV is implemented enabling extensive and comprehensive tests without risks. The structure of the room for the simulation is shown in Fig. 6. The scenario starts with the entering of the building. Next, the UAV heads to two goals (blue circles) between the obstacles. The configuration of the obstacles is disadvantageous and can lead to local minima, see V-formed obstacle in the middle of the room in Fig. 6. The first goal is easily reached by both algorithms. While, the second goal lies behind an obstacle with a V-form.

The two controller have the same data and navigation solution in the simulation. Both run at 20 Hz and the low level Pixhawk controller has the same setup. The benchmark APF controller is a path follower and obstacle avoidance controller provided by Mathworks ROS-Toolbox controller [22]. The generated real-time controller is parametrized to obtain a smooth flight and the controller structure with the APF is not modified.

B. Results

Fig. 6 shows the resulting trajectories with our LEC-APF controller (green line) and the results of the classical APF controller (red line). On the results, it can be seen that the proposed LEC-APF controller can reach both of the goals. In contrast, the classical APF controller can only handle the first goal and has difficulties with the V-formed obstacle. There is a local minimum of the APF, where the UAV reaches a deadlock.

Both controller work reactively, meaning they only use actual data of the laser scan and no prior knowledge about the map. The classical APF computes the actual velocity based on obstacles points directly, in contrast to the proposed LEC-APF method, which calculates ellipses for the obstacle avoidance. Due to the ellipsis calculation, the LEC-APF can out-perform the classical APF in such situations.

VI. Validation

After the simulative verification of the LEC-APF controller, this section discusses its operation on our real UAV. First, the necessity of the adaption of the transition points is addressed. Additionally, a demonstrative rescue operation is carried out in a dark warehouse looking for missing persons.

A. Adaptation of the Transition Point

As a next step, the LEC-APF concept is investigated in a simple setup with one or two obstacles in a gym at the Karlsruhe Institute of Technology. In this first investigation, the influence of the following parameters are examined: The ellipsis radius of influence r_{RoI}, safety range Δ and the maximal velocity $v_{\max}$.

The results show the following findings:

1. The parameters γ and r_{RoI} are strongly velocity dependent
2. The safety range Δ does not significantly influence the performance of the LEC
3. The maximal velocity $v_{\max}$ should not be chosen to large, because the extrapolation of the room requires time

These finding have not been addressed in the literature yet, however they are essential for real flights implementation due to the reactive character of the LEC. With these results of the practical examination, the overall rescue scenario is carried out.

B. Rescue Scenario

1) Scenario setup: The realistic rescue scenario indicates the general applicability of the LEC-APF controller: An

TABLE I

Times in minutes for the person rescuing based on empirical estimation of fire brigade Dortmund and the measurements on the demonstration

	Getting dressed System start	Searching	Saving persons
Human exp.	4 min	11 min	5 min
Drone exp.	2 min	5 min	5 min

operator uses the system to find missing persons in a dark warehouse. The demonstration is carried out in Dortmund in a warehouse at the IKEA distribution center.

The system is shortly introduced to the rescue team. The control of the system is intuitive, and no long training is necessary. It uses standard fire brigade symbols for the missing persons, which makes its use easy for the head of operations. The operator's tasks are: choosing the goals with the high-level GUI, motoring the UAV and checking the results of the person detection algorithm.

In the scenario, three hidden persons in the warehouse have to be found by the rescue team. The head of operations has the possibility to use the indoor drone system instead of letting the rescue personal enter the building.

2) Results and Subjective Feedbacks: The main result of the demonstration is the collision-free flight of the UAV. Fig. 5 shows the resulting trajectory (light blue lines) and the detected persons (blue circles with a square). The UAV is able of crossing the narrow passages, after the entrance. The system is able to handle invalid goals, which are chosen by the operator during the operations. The map generated in the scenario helps the rescue team to find the missing persons. The benefit of the system is measured with the time being saved. The Table I provides an example of the time saving. Exploring the room by the rescue personal takes ca. 20 minutes based on the estimation of the fire brigade. In contrast, using the UAV system reduces the necessary time to 10 minutes. The subjective feedback pointed out that the system is intuitive to use and the resulting map with the missing persons is self-evident even for novel operators.

VII. Conclusion and Outlook

This paper presents the development and the practical usage of obstacle avoidance method for indoor UAVs. The system designed for emergency personnel, can navigate safely in unknown indoor rooms with high level commands from an operator. The obstacle avoidance uses an improvement of the method of Limit Ellipses. The proposed Limit Ellipses Controller is able to navigate safely in an unknown room based on a single 2D laser scanner. The system helps the rescue team to find the missing persons faster. The practical usability of the system is demonstrated in a realistic scenario with the fire department Dortmund.

Acknowledgement

Here, we would like to thank the fire brigade Dortmund for the valuable feedbacks during the development of the system. This work is supported by the Federal Ministry of Transport and Digital Infrastructure, in the mFUND research initiative with the Project number 19F2074C.

References

[1] K. P. Valavanis and G. J. Vachtsevanos, Eds., *Handbook of Unmanned Aerial Vehicles*. Springer Netherlands.

[2] A. Završnik, Ed., *Drones and Unmanned Aerial Systems*. Springer International Publishing.

[3] R. Murphy, "Activities of the rescue robots at the World Trade Center from 11-21 september 2001," vol. 11, no. 3, pp. 50–61.

[4] K. Mueller, J. Atman, N. Kronenwett, and G. F. Trommer, "A Multi-Sensor Navigation System for Outdoor and Indoor Environments," pp. 612–625.

[5] Bruno Siciliano and O. Khatib, *Springer Handbook of Robotics*, 2nd ed. Springer Berlin Heidelberg.

[6] D.-H. Kim and J.-H. Kim, "A real-time limit-cycle navigation method for fast mobile robots and its application to robot soccer," vol. 42, no. 1, pp. 17–30.

[7] B. C. Min, E. J. Lee, S. H. Kang, and D. H. Kim, "Limit-cycle navigation method for a quad-rotor type UAV," in *2009 IEEE International Symposium on Industrial Electronics*. IEEE, pp. 1352–1357.

[8] L. Adouane, "Orbital Obstacle Avoidance Algorithm for Reliable and On-Line Mobile Robot Navigation," p. 8.

[9] S. Prophet and G. F. Trommer, "Reactive Navigation in Cluttered Indoor Environment for Autonomous MAVs," in *2020 27th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS)*. IEEE, pp. 1–8.

[10] Payal, Akashdeep, and C. Raman Singh, "A Summarization of Collision Avoidance Techniques for Autonomous Navigation of UAV," in *Proceedings of UASG 2019*, ser. Lecture Notes in Civil Engineering, K. Jain, K. Khoshelham, X. Zhu, and A. Tiwari, Eds. Springer International Publishing, vol. 51, pp. 393–401.

[11] M. Lao and J. Tang, "Cooperative Multi-UAV Collision Avoidance Based on Distributed Dynamic Optimization and Causal Analysis," vol. 7, no. 1, p. 83.

[12] S. Bortoff, "Path planning for UAVs," in *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No.00CH36334)*. IEEE, pp. 364–368 vol.1.

[13] J.-W. Park, H.-D. Oh, and M.-J. Tahk, "UAV collision avoidance based on geometric approach," in *2008 SICE Annual Conference*. IEEE, pp. 2122–2126.

[14] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2. Institute of Electrical and Electronics Engineers, pp. 500–505.

[15] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," vol. 19, no. 5, pp. 1179–1187.

[16] J. Q. Cui, S. Lai, X. Dong, and B. M. Chen, "Autonomous Navigation of UAV in Foliage Environment," vol. 84, no. 1-4, pp. 259–276.

[17] L. Adouane, A. Benzerrouk, and P. Martinet, "Mobile Robot Navigation in Cluttered Environment using Reactive Elliptic Trajectories," vol. 44, no. 1, pp. 13 801–13 806.

[18] J. Atman and G. F. Trommer, "Object Classification and Semantic Mapping," in *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. IEEE, pp. 137–143.

[19] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, "PIXHAWK: A system for autonomous flight using onboard computer vision," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, pp. 2992–2997.

[20] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3. IEEE, pp. 2149–2154.

[21] C. E. Aguero, N. Koenig, I. Chen, H. Boyer, S. Peters, J. Hsu, B. Gerkey, S. Paepcke, J. L. Rivero, J. Manzo, E. Krotkov, and G. Pratt, "Inside the Virtual Robotics Challenge: Simulating Real-Time Robotic Disaster Response," vol. 12, no. 2, pp. 494–506.

[22] "MATLAB," The MathWorks Inc. [Online]. Available: mathworks.com