International Journal of Aeronautics and Astronautics

https://dergipark.org.tr/en/pub/ijaa

2022, VOL:3, ISS:1, 7-19 e-ISSN: 2757-6574

Research Article

Safe land system architecture design of multi-rotors considering engine failure

Seyed-Yaser Nabavi-Chashmi*, Davood Asadi*, Karim Ahmadi

Department of Aerospace Engineering, Adana Alparslan Turkes Science and Technology University, Adana, Türkiye

Received	Revised	Accepted
December 5, 2021	February 5, 2022	**February 18, 2022**

Keywords

Emergency Scenarios
Functional Architecture
Multi-Copters
Physical Architecture

ABSTRACT

There is growing interest to use drones for application like delivery services, air taxi, surveillance, and inspection to reduce operational time and cost and increase the performance and functionalities. However, there are some risks related to using this technology one of them is the hazard to the humans and assets in the case of an emergency scenario like motor failure. The current technologies suggest to land as soon as possible in such these scenarios, however finding and selecting a suitable landing area considering the capabilities of the faulty drone and controlling the drone towards the selected point is a problem which requires provisions during the design process of the drone. The situations get more complex considering that the vehicle should be able to adopt itself by its time varying environment. In this paper the system approach is used to break the safety requirements to functional and physical requirements and based on these requirement analyses, the functional and physical architectures of the drone are designed. The proposed design suggests that the drones aggregate their perception about the environment to maximize the safety of people and assets through a special databank called potential landmark databank.

* Corresponding author, e-mail: dasadihendoustani@atu.edu.tr

Authorship contribution statement for Contributor Roles Taxonomy

Seyed-Yaser Nabavi-Chashmi: *Writing, Conceptualization, Methodology, Development, and Formal analysis specifically on using System Concepts, Requirement management and Functional and Physical architecture development.* **Davood Asadi:** *Investigation, Methodology, Supervision, Writing – review & editing.* **Karim Ahmadi:** *Formal analysis on Functional and Physical architecture development*

Conflicts of Interest: The authors declare that they have no conflict of interest.

Citation: *Asadi, D., Ahmadi, K., Nabavi Chashmi, S. 2022. Safe land system architecture design of multi rotors considering engine failure. International Journal of Aeronautics and Astronautics, 3(1), 7-19.*

1. Introduction

Using autonomous drones for operations like good delivery, transportation and remote sensing proposes very operational capabilities to improve the efficiency and reduce cost and time. However, they can lead to new challenges in subjects like safety, security and reliability. So, the development of drone system is not limited just to technical problems and many different perspectives should be included to develop a reliable and secure system. This offers to employ system perspective during the design process to integrate different disciplines to achieve the best operational solution. In spite of plenty of works which has been done on detailed problems like control system design [1-7], path planning, obstacle avoidance, and navigation [8-10], there are few works which look the problem in a system perspective. Forced landing of UAV was evaluated by [11] to propose an architecture to select the best landing site between some pre-known candidate landing sites. However, this architecture cannot handle the challenges raised by the changes in the environment. On the other hand, Ref. [12, 13] focuses on guidance of a remotely piloted vehicle using natural landmarks and its landing on a selected area. The proposed architecture relies on a remote operator which reduces its operability in some applications. Another problem which was not considered in the previous studies was about the presence of humans in the landing area. Ref [14] uses data sources like to incorporate the risks posed to property and people on the ground during the trajectory planning. The main focus of this reference is risk while other factors like UAV capabilities, trajectory planning should be also included it the functional and physical structure of the system. In this paper the problem of emergency landing is studied as a system problem in which the safety requirement is considered as the top-level requirements and is mapped to a concept of operations (CONOPS) and using this CONOPS the functional and physical architecture of the system is developed. In spite of other references which employ a Down-Up strategy to propose solutions for some detailed problems, in this paper the main problem is to convert the general problem of safe landing to some the detailed problems using an Up-Down breakdown strategy and find the inter-relationships between these variety of problems. Clearly presenting final solutions for all of these detailed problems are out of the scope of this paper. However, using other researches presented in the literature and some initial evaluations possible solution approaches will be also discussed in this paper.

2. Operational and Functional Architecture

To design an appropriate architecture for the multi-copter a precise definition of system of interest (SOI) and its environment is required first. As depicted in Figure 1, the safe landing system (SLS), as the SOI, is a system which should operate in an environment including the UAV, ground, flying space, peoples and assets and regulations. The SLS is a system which can be used by a multi-rotor to provide maximum possible safety for humans, natural environment and assets while following the mission of UAV. The top-level requirements, guaranteeing the safety of the system, as defined for this system as:

Req. 1 - The SLS must be completely autonomous

Req. 2 - The SLS should minimize the hazards and risks for the people and assets in UAV engine-failure scenarios

Req. 3 - The SLS should try to keep the UAV safe for future uses in engine-failure scenarios

Req. 4 - The SLS should try to enable the UAV to accomplish its operations in fault scenarios

Req. 5 - The SLS must be able to adopt itself with the changes in the environment (e.g., wind, night and day, new constructions, disasters)

Req. 6 - It is recommended that the SLS has minimum supporting systems dependencies and human workload

Req. 7 – The SLS should be able to keep and share flight recordings for future flights

https://doi.org/10.55212/ijaa.1032693

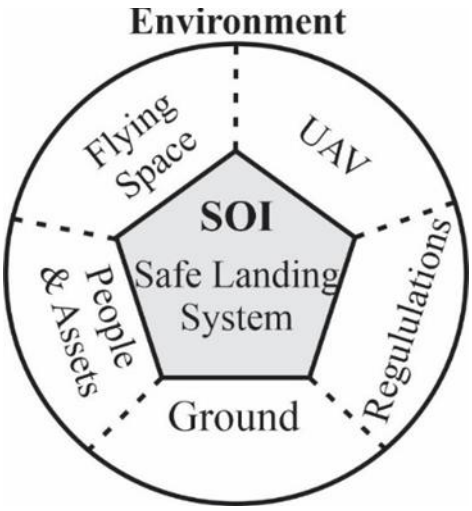

Fig. 1. System of interest and its environment

It is assumed that the UAV has a pre-determined trajectory to accomplish its mission in normal scenario. Here the problem is to how the UAV should flight when at least one of its engines fails. Clearly the trajectory should be re-planned considering the mission, safety issues and UAV capabilities. To solve this problem and based on the definition of SOI, the operational architecture of the system is proposed according to the Figure 2. In this operational architecture of the system, two definitions are used (Figure 3):

- Landing site: This is an area which the UAV can potentially land
- Landing region: Any desired area around the UAV which can fly to find the Landing Site

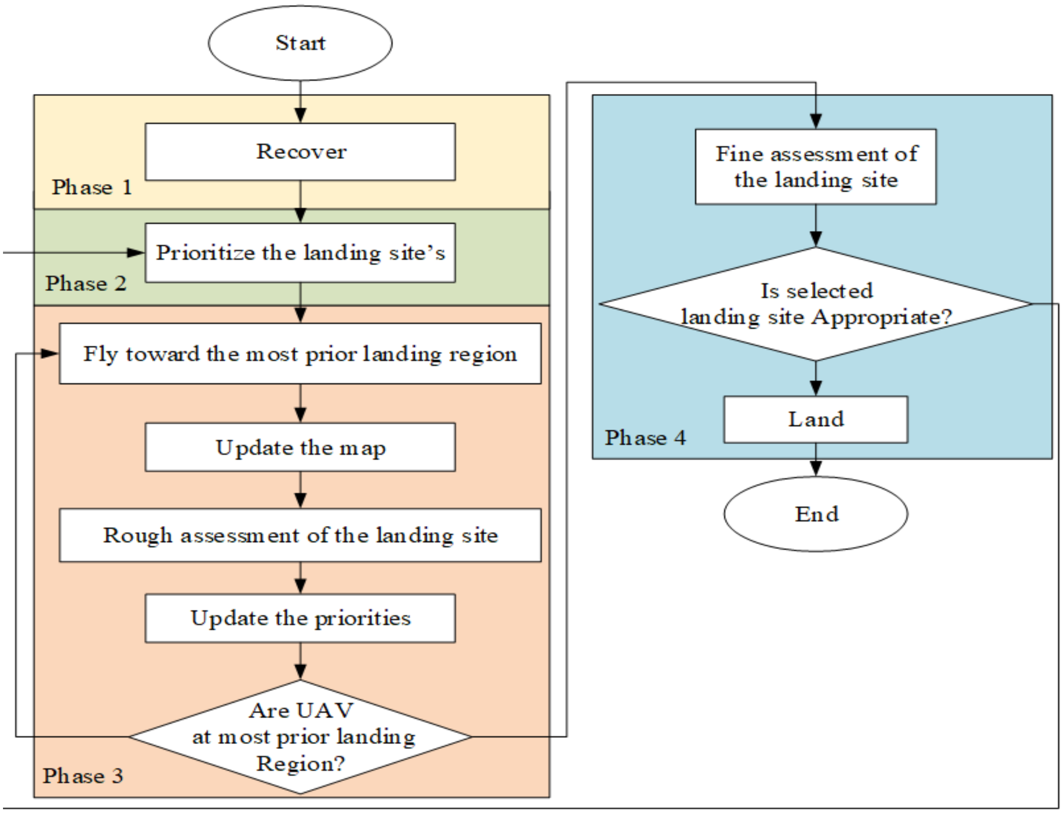
Start
Recover
Phase 1
Prioritize the landing site's
Phase 2
Fly toward the most prior landing region
Update the map
Rough assessment of the landing site
Update the priorities
Are UAV at most prior landing Region?
Phase 3
Fine assessment of the landing site
Is selected landing site Appropriate?
Land
Phase 4
End

Fig. 2. Operational architecture of SLS

https://doi.org/10.55212/ijaa.1032693

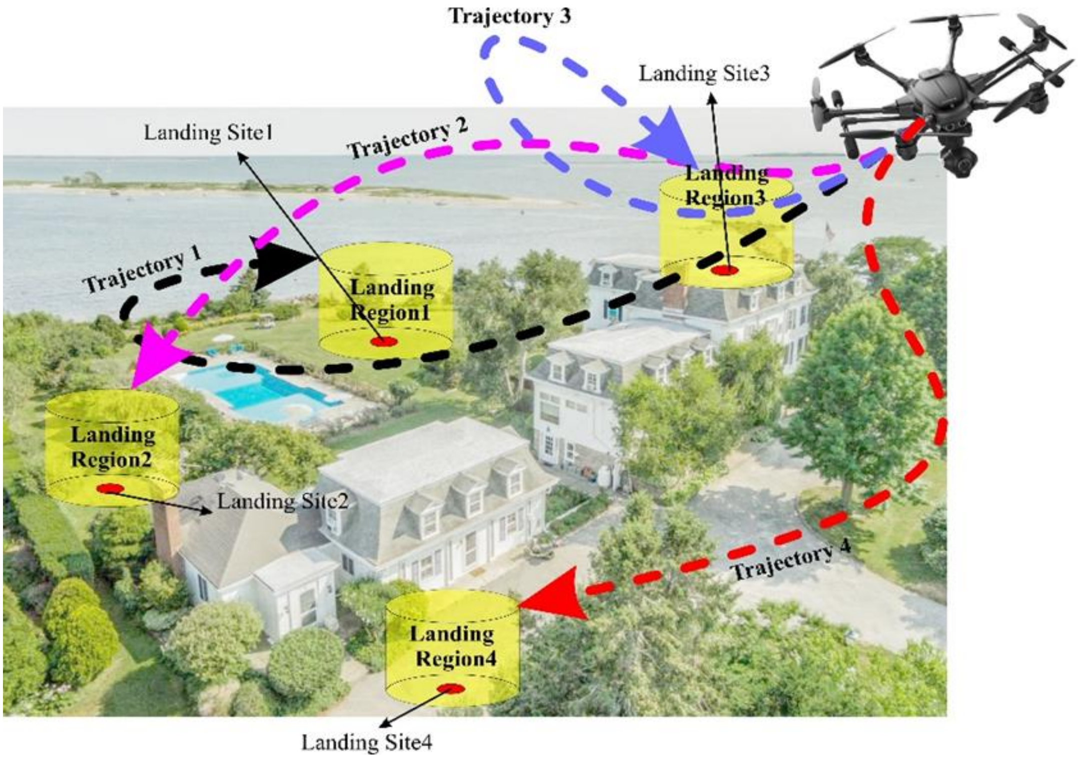

Fig. 3. Operational architecture of SLS

According to this algorithm the UAV priories and monitors the landing site, updates its perception about the environment and plans the trajectory to the selected site repeatedly during its flight. In this architecture the drone first makes a coarse analysis using its available map to identify the possible landing sites and then prioritize them based on criteria's like required trajectory, emergency considerations and capabilities of the UAV to reach the site. During the flight towards the selected site, the UAV uses its vision-based capabilities to update the map. Then the updated map will be used to re-assess the selected site and update it if necessary. This assessment is rough to increase the operational capabilities while using limited computational and sensing resources. When the drone reached to a specific region around the landing site, called here landing region, it should make a fine assessment of landing site considering criteria like suitability for land, collision risk in the case of presence of people or animals. The computational and measurement capabilities are key factors on determining the landing region dimension and the flight characteristics. Based on the fine assessment the UAV may continue to land or may abort and switch to the next saved priority. Based on this architecture the system level functions of the are:

FCN 1 - SLS should be able to detect new Landing sites

FCN 2- SLS should initialize the potential landing sites list

FCN 3- SLS should make rough assessment of potential landing sites

FCN 4- SLS should make fine assessment of selected landing site

FCN 5- SLS should be able to update potential landing sites list

FCN 6- SLS should be able to flight towards the selected landing site

FCN 7- SLS should be able to land on the selected landing site

FCN 8- SLS should be able to propose a suitable trajectory for the flight

FCN 9- SLS should be able to analyze the position of the UAV

https://doi.org/10.55212/ijaa.1032693

The system requires suitable physical architecture to support these functions. The proposed architecture and its relationship to the functions are presented in Table 1. According to this table the SLS system is decomposed into four physical subsystems. The subsystems which are presented here, are themselves systems which was broken to lower levels in the same manner. In the following section Lower-level requirements and operational and functional architecture are presented. The final Product Breakdown Structure extracted from these architectural analyses are presented in Fig. 4. The datils about the relationships between the PBS of subsystems and their functional and physical architectures are also presented in the following section.

Table 1. Functional and Physical architecture relationship for SLS

		Physical Element									
		LSD	CCU	RDSS	**Algorithms**						
					LA	RAA	FAA	SSA	TPA	FCA	LCA
Functions	1	√		√	√						
	2			√							
	3					√					
	4						√				
	5			√	√						
	6			√	√			√		√	
	7				√			√			√
	8								√		
	9	√			√						

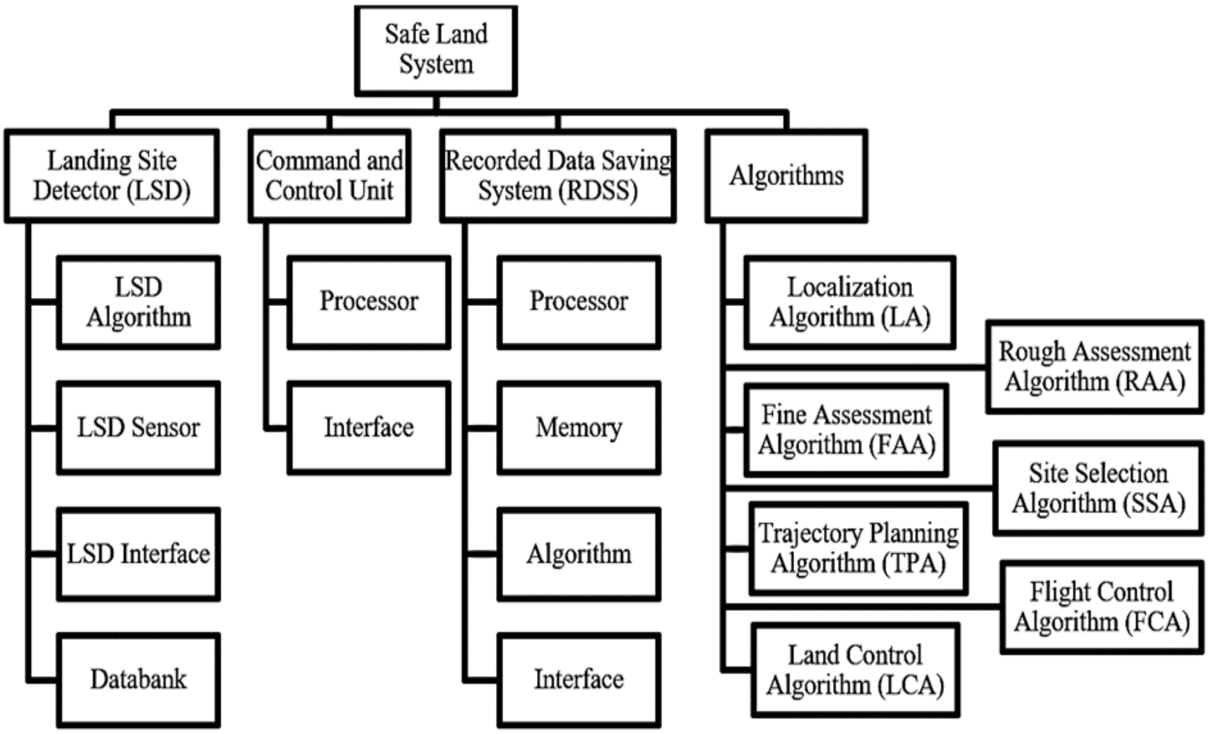
Safe Land System
Landing Site Detector (LSD)
Command and Control Unit
Recorded Data Saving System (RDSS)
Algorithms
LSD Algorithm
LSD Sensor
LSD Interface
Databank
Processor
Interface
Processor
Memory
Algorithm
Interface
Localization Algorithm (LA)
Rough Assessment Algorithm (RAA)
Fine Assessment Algorithm (FAA)
Site Selection Algorithm (SSA)
Trajectory Planning Algorithm (TPA)
Flight Control Algorithm (FCA)
Land Control Algorithm (LCA)

Fig. 4. Proposed Product Breakdown Structure (PBS) for SLS

3. Subsystem Level Operational Architecture

3.1. Land site detector

In this section the system approach is applied to the LSD subsystem to determine the functional and physical architecture of LSD. Considering the system level analysis, the LSD refers to

- all elements required to provide necessary information for detecting the landing sites
- databank for storing the landing sites during flight
- algorithms for detection the previously stored landing sites
- algorithms for finding the new landing sites

As the first step the requirements of LSD is extracted using the breakdown of the requirements at the system level. These requirements and their relationships with the system-level requirements are presented in Table 2. Based on these requirements the functional architecture of the land site detector is presented in Figure 5. According to this architecture the landing sites are initially set by the RDSS based on the planned trajectory. The SLS functional architectures suggests that during the flight it should be updated based on the new detected landing sites, as well as the changes of the loaded landing sites and also newly detected landing sites. So, it is required to measure the environment during the flight. Analysing the measured paramours and comparing it to the available data (considering the current position of UAV) two main questions can be answered: Is a new site detected? And is an available landing site conditions changed? Answering these questions helps to update the UAV databank of potential landing sites.

Table 2. Requirements of LSD and their relationships with system level requirements

		System Level Requirements						
No.	**Description**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1-1	The LSD must detect mapped land sites (in LSD) during flight		√	√				
1-2	The LSD must detect new land sites during flight autonomously	√				√		
1-3	It is recommended that the LSD should use natural landmarks for detection						√	
1-4	The LSD calculations should be done CCU					√		
1-5	The LSD should be able to detect the land sites during the flight	√						
1-6	The LSD should contain information about the pre-identified landing sites		√					
1-7	The LSD should be updatable automatically using every new flight	√				√	√	
1-8	The LSD should contain the risk of presence of people/assets in each landing site		√					
1-9	The LSD should contain some measures about the suitability of each landing site			√				

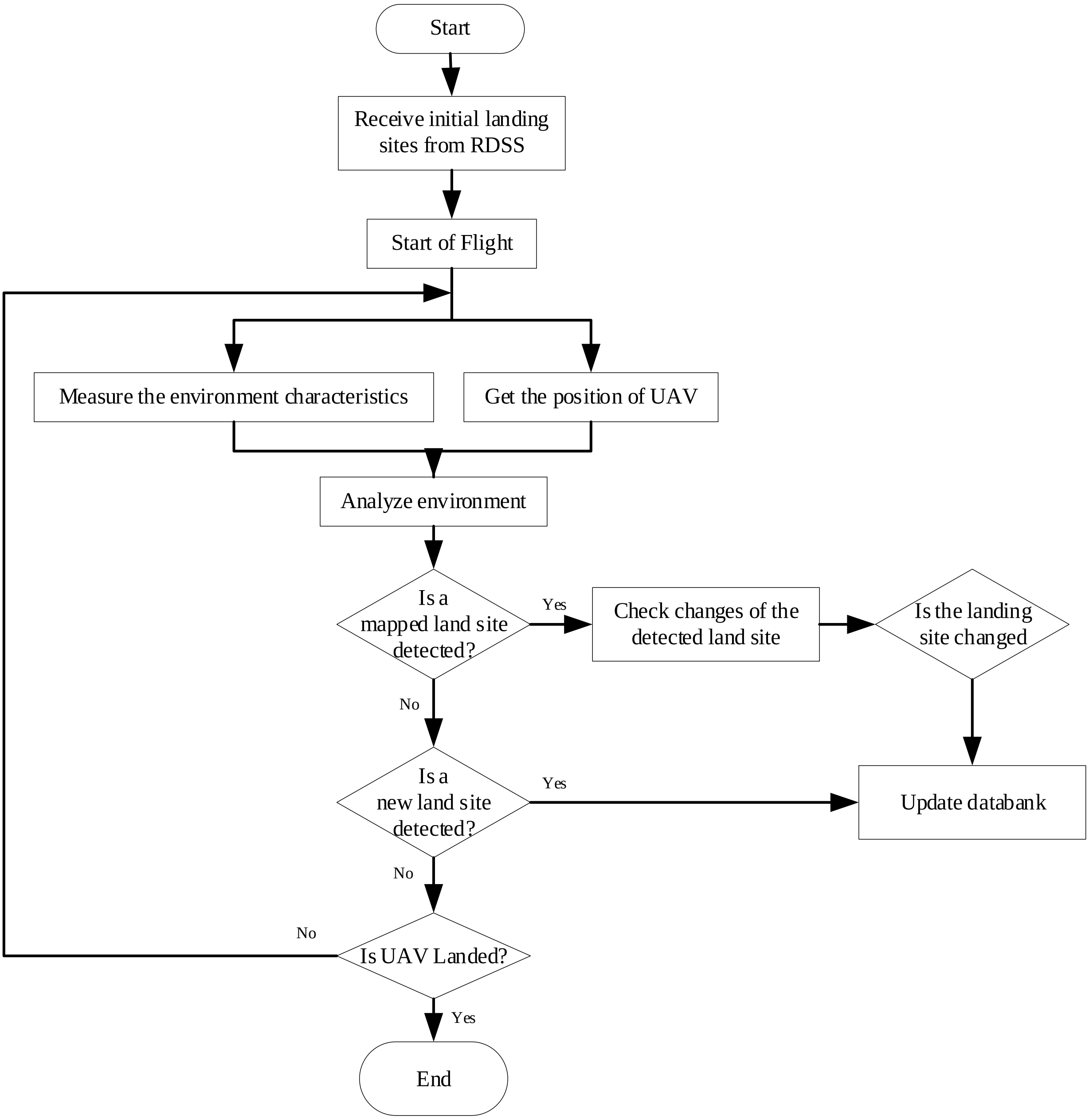
Start
Receive initial landing sites from RDSS
Start of Flight
Measure the environment characteristics
Get the position of UAV
Analyze environment
Is a mapped land site detected?
Yes
Check changes of the detected land site
Is the landing site changed
No
Is a new land site detected?
Yes
Update databank
No
No
Is UAV Landed?
Yes
End

Fig.5. Operational architecture of LSD

This architecture offers the LSD functions as presented in Table 3. The physical elements required to apply this functional architecture is also presented in this table. These elements should be analyzed similarly to determine their requirements and if necessary, they can be considered again as a system to be designed or as a part to be selected, ordered or implemented.

https://doi.org/10.55212/ijaa.1032693

Table 3. Functional and physical architecture relationship for LSD

			Element			
			LSD Algorithm	LSD Sensor	LSD Interface	Databank
Functions	1-1	LSD receives position data			√	
	1-2	LSD measures the environment characteristics		√		
	1-3	LSD analyzes the measured characteristics	√			
	1-4	LSD detects the mapped landing sites	√			
	1-5	LSD detects new landing sites	√			
	1-6	The LSD receives initial landing site from RDSS			√	
	1-7	The LSD receives the start signal from the CCU			√	
	1-8	The LSD checks the changes of current landing sites	√			
	1-9	The LSD update its databank				√
	1-10	The LSD receives the landing signal from the CCU			√	

3.2. Computation and communication unit

Similar to what which done for LSD the functional and physical architecture of CCU are determined in this section. The CCU is the system which is responsible of conducting the calculation required by all subsystems of the SLS. To design the CCU, the system level requirements have been broken down to the requirements related to this this subsystem as presented in Table 4. The functional architecture of CCU is designed based on the derived requirements as presented in Figure 6. According to this functional architecture three algorithms are running in parallel during the flight: Flight Control Algorithm (FCA), Land Site Detector (LSD) and Localization Algorithm (LA). Every time that LSD detects a new landing site or updates an available landing site, Site Selection Algorithm (SAA) and Trajectory Planning Algorithm (TPA) will be run to determine the new trajectory to the best landing site in failure scenarios. On the other hand, if LA shows that the UAV is in the selected landing site, the Fine Assessment Algorithm (FAA) will be ran to accurately analyse the landing site in terms of the changes, presence of the people and height map. If the selected landing site is suitable for land then Landing Control Algorithm (LCA) should be ran else the selected landing site and the trajectory should be updated. In Table 5 the LSD functions designed based on this architecture is presented and according to the analysis on requirements and functions two subsystems are proposed for CCU: Processor and Interface.

Table 4. Requirements of CCU and their relationships with system level requirements

		System Level Requirements						
No.	**Description**	1	2	3	4	5	6	7
2-1	The CCU should be installed on the UAV						√	
2-2	The CCU should have connection on-line or off-line with PLSD						√	
2-3	The CCU should be able to run all algorithms	√						
2-4	The CCU should be able to transfer new found data to the RDSS							√

Table 5. Functional and physical architecture relationship for CCU

			Element	
			Processor	Interface
Functions	2-1	CCU Run LA, FCA, LSD, FAA, TPA and LCA algorithms	√	
	2-2	CCU update RDSS		√
	2-3	CCU should receive data from LSD		√
	2-4	CCU should start its computational algorithm after the start of flight		√

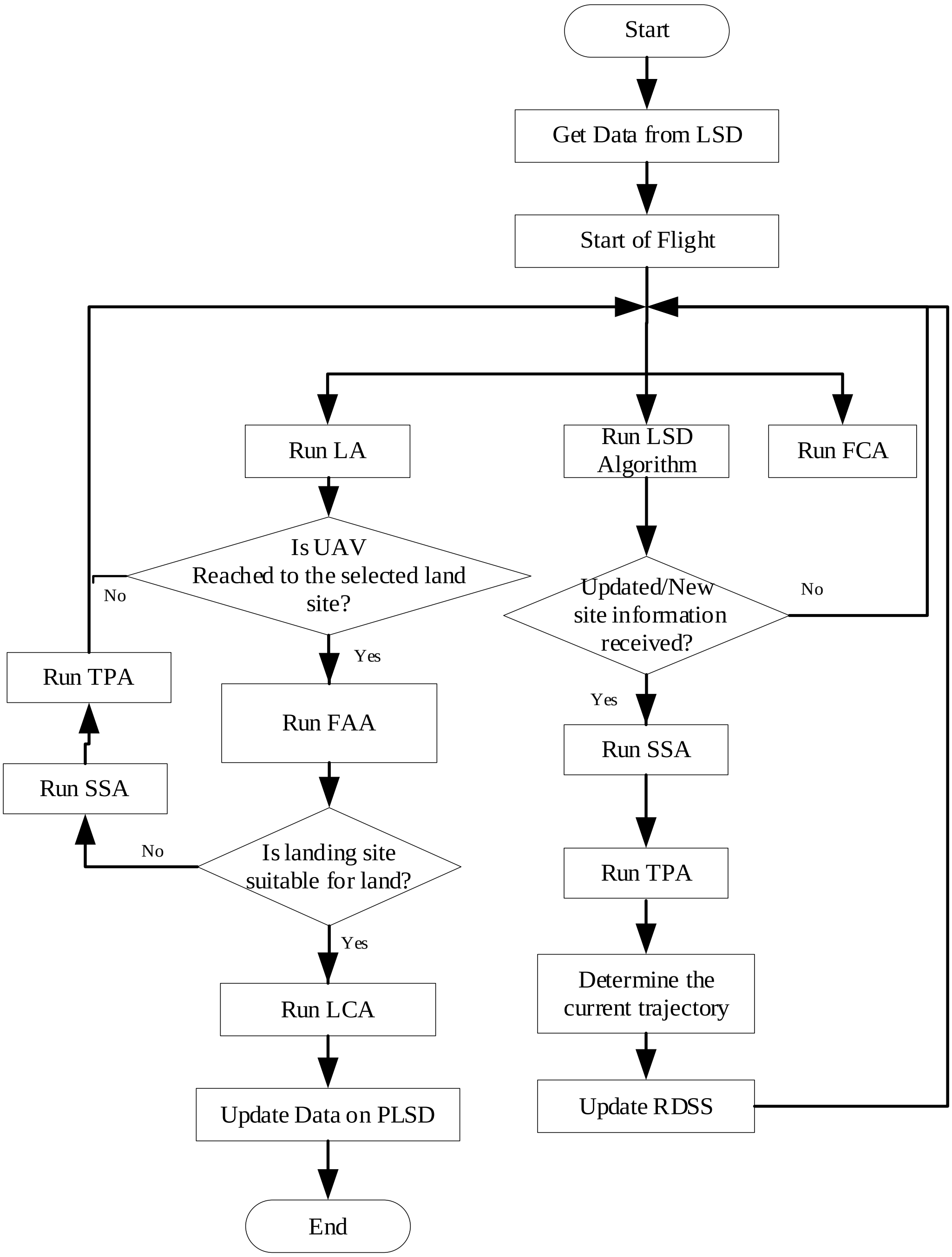

Fig. 6. Operational architecture of CCU

3.3. Algorithms

Algorithms are software elements which runs on CCU to provide necessary commands for the UAV to follow a suitable trajectory. These algorithms are: Localization Algorithm (LA), Rough Assessment Algorithm (RAA), Fine Assessment Algorithm (FAA), Site Selection Algorithm (Site Selection Algorithm (SSA), Trajectory Planning Algorithm (TPA), Flight Control Algorithm (FCA) and Land Control Algorithm (LCA). It should be noted that LSD and RDSS have their own algorithms which are considered directly as a part of these subsystems and are not presented here. At the system level it is sufficient to determine the requirements of algorithms and look them as parts which is not necessary to break more. The software developer uses the requirements to develop them. These algorithms should be delivered based on specific delivery criteria which should be determined during the design process. In Table 6 the requirements of different algorithms presented in this table are summarized.

Table 6. Requirements of algorithms and their relationships with system level requirements

			System Level Requirements						
	No.	**Description**	1	2	3	4	5	6	7
LA	4-1	The LA should be run on CCU to localize the UAV during the flight	√						
	4-2	The LA should be able work without the need of interference of human (tuning, prioritizing, …)						√	
	4-3	The LA should be able to work when the UAV is rotating due to some engine’s failures				√			
	4-4	The LA should be able to be adopted to localize the UAV during various environmental conditions					√		
RAA	5-1	The RAA should be run on CCU to assess and score the sites which are already available in the map	√					√	
	5-2	The RAA should consider the hazard probability to people and assets in scoring the sites (using LSD)		√					
	5-3	The RAA should consider the landing suitability in scoring the sites (using LSD)			√				
FAA	6-1	The FAA should be run on CCU and use SLS sensors to find and score new landing sites	√						
	6-2	The FAA should consider the hazard probability to people and assets in its scoring mechanism		√					
	6-3	The FAA should consider the suitability for landing in its scoring mechanism			√	√			
	6-4	The FAA should be able to update LSD							√
SSA	7-1	The SSA should be able to analyze the available sites (in LSD) to propose the best selected site	√						
	7-2	The SSA should minimize the hazards and risks for the people and assets in engine-failure scenarios		√					
	7-3	The SSA should try to keep the UAV safe for future uses in engine-failure scenarios			√				
	7-4	The SSA should try to enable the UAV to accomplish its operations in engine-failure scenarios				√			
	7-5	The SSA should decide the self-destruction mode if necessary		√					
TPA	8-1	The TPA must be run on CCU	√						
	8-2	The TPA must use data of PLSD to plan the trajectory	√						
	8-3	The TPA must propose trajectories which makes minimum risk to people and assets		√					
	8-4	The TPA must propose trajectories considering the failure/abort scenarios				√	√		
FCA	9-1	The FCA must be run on CCU	√						
	9-2	The FCA should be able to detect the failure of the UAV				√			
	9-3	The FCA should be able to control the UAV when some engines are failed		√	√		√		
	9-4	The FCA should be compensate the effects of environmental changes like wind		√	√	√			
	9-5	The FCA should use as minimum energy as possible							
LCA	10-1	The LCA must be run on CCU	√						
	10-2	The LCA should be able to land the UAV in normal scenario and when some engines are failed			√	√			
	10-3	The LCA should be compensate the effects of environmental changes like wind					√		
	10-4	The LCA should be able to land the UAV on unprepared landing sites				√			

https://doi.org/10.55212/ijaa.1032693

3.4. Recorded data saving system

The last subsystem is Recorded Data Saving System which is a ground system and is responsible to aggregate the data collected during different flights to propose initial landing sites to any new flight. These subsystem requirements are all related to the Req. 7 in system level requirements as presented in the Table 7. According to the operational architecture of RDSS (Fig. 7), this system operates in two modes loading mod (Transferring data to the UAV) and update mod (Receiving data from the UAV). In loading mod, the RDSS receives the planned trajectory of UAV and based on this trajectory, determines the part of the map which is necessary to be send to the UAV and then sends these data to the UAV. In update mod, the RDSS connected to a UAV after its operation to receive its updated landing sites. After that RDSS analyses the changes in the maps and decides that how its databank should be updated. This decision-making process is required to avoid updating databank based on any unreliable recordings by the UAVs. According to the decision-making process the map will be updated at the end. The functional architecture of the RDSS and its relationship with proposed physical architecture is presented in Table 8.

Table 7. Requirements of RDSS and their relationships with system level requirements

		System Level Requirements						
No.	**Description**	1	2	3	4	5	6	7
11-1	The RDSS should be able to get and store all flights recording from the UAVs							√
11-2	The RDSS should be able to create and load updated maps into the UAVs							√
11-3	The RDSS should be able to analyze and score the maps according to the requirements of LSD							√

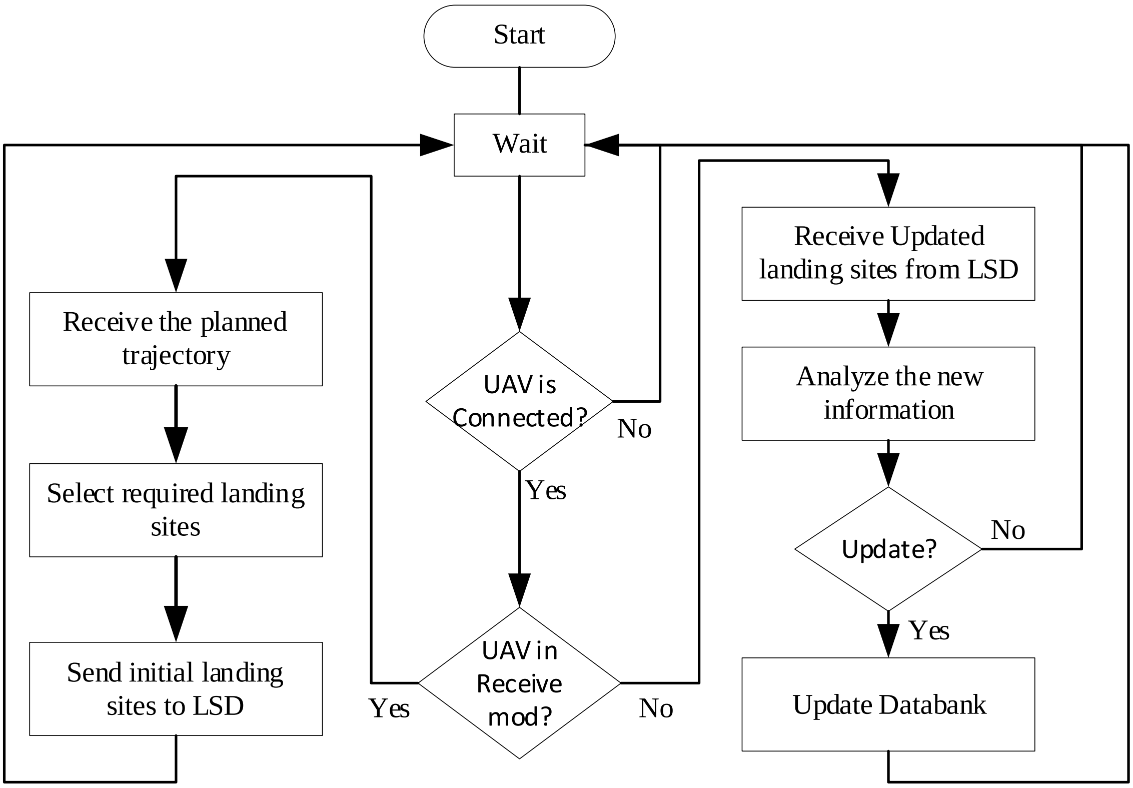
Start
Wait
Receive the planned trajectory
Select required landing sites
Send initial landing sites to LSD
UAV is Connected?
No
Yes
UAV in Receive mod?
Yes
No
Receive Updated landing sites from LSD
Analyze the new information
Update?
No
Yes
Update Databank

Figure 7. Operational architecture of RDSS

https://doi.org/10.55212/ijaa.1032693

Table 8. Functional and Physical architecture relationship for RDSS

			Element			
			Processor	Memory	Algorithm	Interface
Functions	1-1	Check that UAV is connection status and mod of connection				√
	1-2	Receive planned trajectory from the UAV		√		
	1-3	Select required landing site for the UAV	√	√	√	
	1-4	Send selected landing sites to the UAV				√
	1-5	Receive Updated landing sites from UAV				√
	1-6	Analyze new information received from the UAV	√	√	√	
	1-7	Update Databank		√	√	

5. Conclusion

In this paper a system approach has been adopted to design the operational, functional and physical architecture of the SLS. To breakdown the architecture between different levels of the system, requirements are used to guarantee that the system level requirements are satisfied. In each level of the system, these requirements are employed to propose an operational architecture for the subsystems. Based on this operational architecture the functional and physical architecture of system/subsystem is extracted. Then the requirements of the system/subsystem are breakdown as the requirements of the its subsystems. The process should be continued until a part is achieved which can be selected or ordered according to a specific set of requirements and delivery criteria. The proposed system, does not limit the SLS only to the UAV itself but it proposes RDSS as a ground system to aggregate the data gathered by different flights to ensure that always an updated perception about the map of area is available. This system also reduces the operational cost of the system and increases the safety of the system.

In this process the derived requirements are mostly used in the design process. To improve the process, other types of the requirements like design requirements can also be considered during the design process. More studies are also required about the relationships between the elements of the system using tools like Design Structure Matrix (DSM) or N*N Matrices.

Acknowledgment

This research is supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) under 3501 programs, with project number [120M793].

References

[1] Asadi, D., Ahmadi, K., and Nabavi, S. Y., “Fault-tolerant Trajectory Tracking Control of a Quadcopter in Presence of a Motor Fault,” *Int. J. Aeronaut. Sp. Sci.*, 2021, doi: 10.1007/s42405-021-00412-9.

[2] Ahmadi, K., Asadi, D., and Pazooki, F., “Nonlinear L1 adaptive control of an airplane with structural damage,” *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.*, 233 (1), 2019, doi: 10.1177/0954410017730088.

[3] Asadi, D., Ahmadi, K., Nonlinear robust adaptive control of an airplane with structural damage, Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng. 234 (2020) 2076–2088. https://doi.org/10.1177/0954410020926618.

[4] Asadi, D. and Bagherzadeh, S. A., “Nonlinear adaptive sliding mode tracking control of an airplane with wing damage,” *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.*, 232 (8), pp. 1405–1420, Feb. 2017, doi: 10.1177/0954410017690546.

[5] Bagherzadeh, S. A., Asadi, D., Detection of the ice assertion on aircraft using empirical mode decomposition enhanced by multi-objective optimization, Mechanical Systems and Signal Processing, Vol. 88, 2017, p. 9-24.

[6] Asadi, D., “Partial Engine Fault Detection and Control of a Quadrotor Considering Model Uncertainty,” *Turkish J. Eng.*, 6 (2), pp. 106–117, 2021, doi: 10.31127/tuje.843607.

[7] Asadi, D., Sabzehparvar, M., Atkins, E. M., and H. A. Talebi, “Damaged airplane trajectory planning based on flight envelope and motion primitives,” *J. Aircr.*, 51 (6), 2014, doi: 10.2514/1.C032422.

[8] Asadi, D., Sabzehparvar, M., Atkins, E. M., and Talebi, H. A., “Damaged Airplane Trajectory Planning Based on Flight Envelope and Motion Primitives,” *J. Aircr.*, 51 (6), pp. 1740–1757, Oct. 2014, doi: 10.2514/1.C032422.

[9] Asadi, D., Sabzehparvar, M., and Talebi, H. A., “Damaged airplane flight envelope and stability evaluation,” *Aircr. Eng. Aerosp. Technol.*, 85 (3), 2013, doi: 10.1108/00022661311313623.

[10] Asadi, D. and Atkins, E. M., “Multi-Objective Weight Optimization for Trajectory Planning of an Airplane with Structural Damage,” *J. Intell. Robot. Syst. Theory Appl.*, 91 (3), 2018, doi: 10.1007/s10846-017-0753-9.

[11] Mejias, L., Fitzgerald, D., Eng, P., and Liu, X., “Forced Landing Technologies for Unmanned Aerial Vehicles: Towards Safer Operations,” *Aer. Veh.*, 2009, doi: 10.5772/6481.

[12] Cesetti, A., Fronton, E. i, Mancini, A., Zingaretti, P., and Longhi, S., “A Vision-based guidance system for UAV navigation and safe landing using natural landmarks,” *J. Intell. Robot. Syst. Theory Appl.*, 57 (1), pp. 233–257, 2010, doi: 10.1007/s10846-009-9373-3.

[13] TURAN, V., AVŞAR, E., ASADİ, D., AYDIN, E.A., Image Processing Based Autonomous Landing Zone Detection for a Multi-Rotor Drone in Emergency Situations, Turkish J. Eng. 5 (2021). https://doi.org/10.31127/tuje.744954.

[14] Di Donato, P. F. A. and Atkins, E. M., “Evaluating risk to people and property for aircraft emergency landing planning,” *J. Aerosp. Inf. Syst.*, 14 (5), pp. 259–278, 2017, doi: 10.2514/1.I010513.