A Parallel Hierarchical Finite State Machine Approach to UAV Control for Search and Rescue Tasks

Vitor de Araujo, Ana Paula G. S. Almeida, Ciro T. Miranda and Flavio de Barros Vidal
Department of Computer Science, University of Brasilia, Brasilia-DF, Brazil

Keywords: UAV Control, Paralell Finite State Machine, Hierarchical.

Abstract: The process of developing a system for an Unnamed Aerial Vehicle (UAV) control is a complex task that we should be able to implement a set of elaborated algorithms with specific features as fast response for non-linear inputs, fault tolerance and easy maneuvering procedures. These features define the reachability and quality of the UAV overall control system. In this position paper, we propose a solution for the UAV control process using a Parallel Hierarchical Finite State Machine (PHFSM) that results in a high level system including all features required by an complex UAV control to search and rescue tasks. Following in this approach we can achieve many improvements than other usual implementations. Initials simulation experiments, using data from a simple UAV model, indicate in direction that many issues can be solved by this approach.

1 INTRODUCTION

Unnamed Aerial Vehicle (UAV) control is a complex challenge to artificial intelligence and robotic research areas, due to the singular involved features (Cavett et al., 2007)(Jeon et al., 2013)(Dayuan et al., 2013). Nowadays, there are many problems that can be solved using these vehicles in military and civilian fields(Ping et al., 2012), consequently the institutions and the researchers have made great efforts to offer better solutions to fulfill this need. The UAV control design is divided in two main groups: *High-Level* and *Low-Level*.

The High-Level approach encloses path planning (Changqing and Zhurong, 2013), mission aims (He et al., 2010)(Wang et al., 2009), swarm control (McCune et al., 2013)(Rabbath, 2013a) and orientation strategies. These approaches usually need more processing time and they have a slow response time to environment changes and complex design. Being directly the main influence to the efficiency and performance of the control process. In order to evaluate the requirements of each group, the High-level approach is mission dependent, because it needs to be coded to specific properties. For example, the concepts of indoor and outdoor environments have completely different control strategies.

The Low-Level approach is regulated by the hardware. Its main responsibility is to offer an Application Programming Interface (API) to high-level control layer. They are composed of small blocks of code and they need to have a fast response time to the environment changes. The relationship between these approaches and the control is described in Figure 1. The Low-level control must be more robust, compact and involving the vehicle model. These developments are complex, considering the constraints of a fast response time to compute uncountable asynchronous and non-linear inputs of the system. This response can be a simple arithmetic calculation or it can demand complicated calculations as evaluation of a stochastic filtering process.

Approaches based on Finite State Machine (FSM) are commonly applied to digital systems designs and

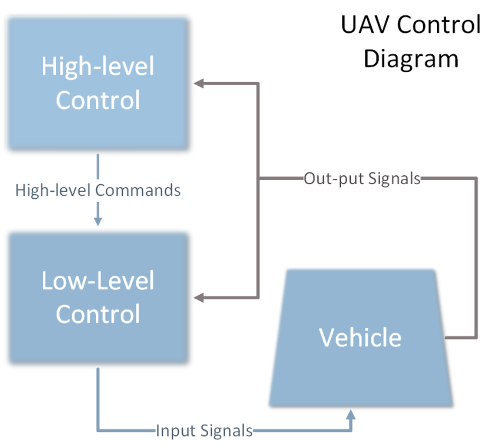

Figure 1: UAV control diagram.

de Araújo V., Paula G. S. Almeida A., T. Miranda C. and de Barros Vidal F..
A Parallel Hierarchical Finite State Machine Approach to UAV Control for Search and Rescue Tasks.
DOI: 10.5220/0005121104100415
In *Proceedings of the 11th International Conference on Informatics in Control, Automation and Robotics* (ICINCO-2014), pages 410-415
ISBN: 978-989-758-039-0

in many situations(Ercegovac, 2000). The main features are: forced modularization defined by the states, and easy response to the environment changes.

In according to (Sklyarov and Skliarova, 2008), (FSM, a) and (FSM, b) the FSM paradigm is divided into three basic models: sequential, hierarchical and parallel. Although, these are not the only way to solve UAV control issues. For example, a sequential FSM is not enough to a hybrid implementations using high-level and low-level approaches. An interesting development is by mixture of parallel and hierarchical models, as described in (Girault et al., 1999).

This propose is well adequate to solve all UAV control problems and even has the capability to include all advantages of the parallel paradigm. Other features are: the fast time response to asynchronous inputs, a system layer division and increase system failure robustness.

Supported by previous arguments, this paper presents a proposal for UAV control based on a parallel hierarchical finite state machine. The Section 2 describes the main related works about UAV control strategies. In Section 3 and 4 the proposed methodology and initial results are presented, respectively. Conclusions and further work are discussed in Section 5.

2 RELATED WORKS

Nowadays, many paradigms are discussed in order to design an UAV system(Devaud et al., 2012). Each of these have their advantages and disadvantages, earned from specific features that perform better results than others, affected by many conditions, for example the amount of available budget.

However, these development sometimes depends of a specific application and environment features that involves an UAV (Branco et al., 2011). In this case, it can result in expensive and slow development process, causing unfeasibility of some applications, due to economic restrictions.

Another issue comes from the necessity of using an expensive hardware with fault tolerance capability, as multiprocessors, to design the UAV control model ((Trimble, 1987), (Branco et al., 2011)).

In case of UAVs used to swarm control applications, there is an exponential growth of these issues due to the dependency of the software low-level layer. One solution proposed by (Rabbath, 2013b) uses the finite state machine (FSM) paradigms to improve the swarm control. If it is possible to determinate that the paradigm is implemented, then the sequence of the states has only impact on the mission task. In short terms, the implementation does not affect the application reliability. Although, (Rabbath, 2013b) proposed the use of FSM only for mission control, because it cannot individually solve these issues under implementation layer. This approach combines all involved algorithms using a FSM, avoiding the dependency between application and implementation layers. This strategy results in others advantages to the system, as volatility, robustness and a natural modularization (Sklyarov and Skliarova, 2008).

A pure sequential FSM approach in all application levels carries out control problem attributes, turning the implementation more complex, and occasionally, impossible.

Many of these problems need a fast time response to the input in low-level control and high-level processing. When these two features are combined, there is an implication of using too many state transitions, resulting in an increase of the cpu-time. One way to make this strategy still in according to these requirements is choosing a parallel hierarchical finite state machine (PHFSM) (Sklyarov and Skliarova, 2008). This paradigm has many advantages of a simple sequential FSM, considering that the main idea is classify the states and synthesize different FSMs to run in parallel. This feature results in many advantages to the overall system, as parallelism, hierarchy and the capacity of using FSM as a procedure. Summarizing, it is feasible to apply the hierarchical and parallel paradigm to solve the UAV control problem.

3 PROPOSED METHODOLOGY

This position paper proposes an abstract UAV control strategy based on a parallel hierarquical finite state machine. In this case, we develop all control inspired on use an UAV for *search and rescue* mission, that consists in to finding a specific target location and returning to base. For initial approach we have started building a simple FSM with reduced set of steps as described in Figure 2.

The high-level diagram of Figure 2 has the first state (*Start*), that is defined to takeoff procedures. In this approach, it is important to denote that all states have many sublayers to allow right task execution. After all takeoff procedures are ended, the next state transitions are related to search and rescue the target. In this case the control module should be able to locate the target and control all low-level processes, as well. These features are needed to keep the UAV in a safe flight and still search for the target. These states are described in Figure 2 by *Move to Search* and *Look for the target* respectively.

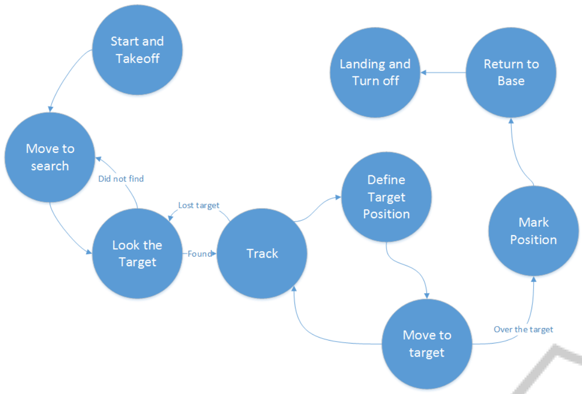

Figure 2: A simple FSM for Search and Rescue mission.

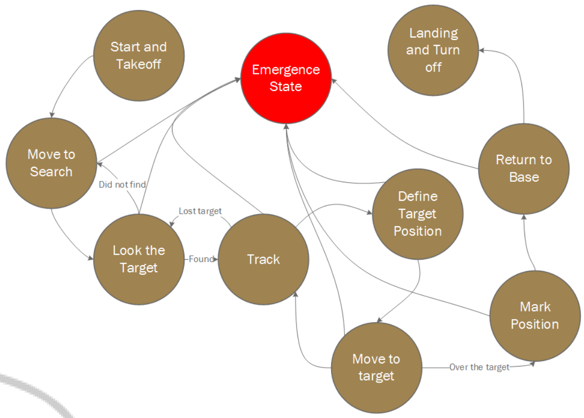

Figure 3: FSM to Search and Rescue task with Hierarquical Stage (HFSM).

After target detection, the next transition is to *Track* state that starts the target tracking procedures. In this stage, the UAV control should be able to run the tracking algorithm, follow the target and still in a safe flight. In the end, after reaching all task requirements, return to base and landing states respectively.

To reach all these features, we need to change a simple FSM, described in Figure 2, to a new FSM based on a parallel and hierarquical approach.

The parallel approach is necessary to allow all processes to run correctly in the same time, avoiding process deadlocks. The hierarchical approach has the function to handle all asynchronous inputs, that can affect the whole system control performance and allows a safe flight.

3.1 FSM with Hierarquical Stage

To include the Hierarchical Stage (HFSM), all states were divided in the high-level layer as groups, and setup a priority level in according to UAV control execution, restricted by the task to be performed. In this implementation, each state has a priority level, reflected to all low-level sublayers that belong to this respective state. Using this approach, it becomes able to handle all asynchronous inputs, since these inputs have a defined priority level.

In the Figure 3, we modified all FSM design, where all states are assigned with the same priority level, because each state is defined in a specific time, i.e., in this design the UAV control is not allowed to have two defined states, on the same level. We include an *Emergency State*, defined as an asynchronous input and assigned higher level priority for emergency situations.

3.2 HFSM with Parallel Stage

The Parallel Stage works directly with all sublayers interface, using and respecting commands from the high-level layer, even information from all highest priorities asynchronous inputs. Each sublayer has a HFSM controlled by a procedure from the high-level main UAV control and synchronized by the specific layer clock time. This clock time can be the same of the highest level or defined by the sublayer task. When using this approach, we can find some control hazards problems: For example, if an asynchronous input from Emergency State turns on. In this case, after hazard handling process the system should be able to return for the last state in all sublayers.

Based on modifications from HFSM (Figure 3), we include the parallel stage as described in Figure 4 to achieve the PHFSM approach. In order to improve the proposed control design, we include the vehicle model information to aid in the navigation process and all states are grouped in according to specific task. Each group is described with a specific color. Under each state has a sublayer with a FSM that is capable to aid the UAV control auxiliary tasks.

In the end, the proposed UAV control design has been developed to allow using an UAV for several tasks, not only to search and rescue tasks.

4 RESULTS

In this section, we will show some partial results achieved from the proposed methodology described in Section 3. These partial results are based on a search and rescue task in which the UAV control steps are: UAV's take-off; find and track a person(target); return to home; and; landing.

The Figure 5 shows the simulation structure used to evaluate the achieved results so far. The computer simulation framework is build using resources from a small UAV system (Parrot AR.Drone) available in

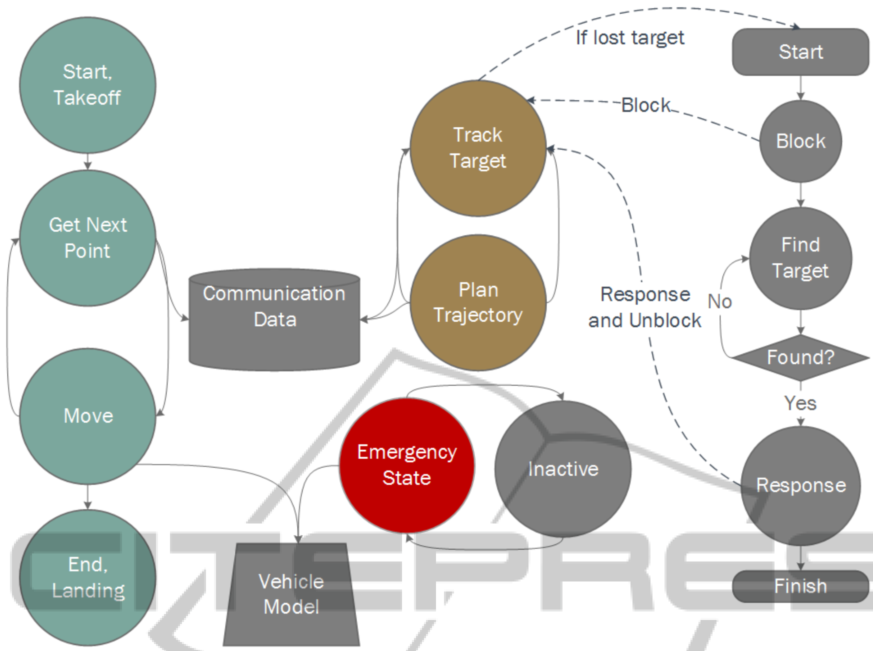

Figure 4: HFSM of Search and Rescue task with Paralell Stage.

ROS library (Quigley et al., 2009) with the TUM package (Huang and Sturm, 2014).

In this experiment, we define a set of sequential and parallel processes working together. For example, a sublayer is responsible to keep the UAV flying on safe mode, that runs in parallel with the sublayer tracking target. A sublayer tracking the target is hierarchically higher than the sublayer of human face detection.

As a way to evaluate the control system reliability, the preliminary results will be assigned assessment strategies: ability to perform the search and rescue task. Based on this approach we can clearly indicate, but is not enough, that the proposed methodology is able to achieve its main objectives.

To assess the ability of the proposed control methodology in accomplishing the task of search and rescue, we decided to use the Fault Tree Analysis (FTA) techinique (Ericson, 1999). In according to (Ericson, 1999) a Fault tree analysis is defined as a top down deductive failure analysis in which an undesired state of a system is analyzed using Boolean logic and lower-level events.

This analysis method is used in safety and reliability engineering to define how and when systems can fail. Therefore, this technique is very important in the parallel and sequential stages analysis, because it is able to measure influences of parallel paradigm when it works together with a sequential FSM. For example, the Figure 6 describes how the FTA analysis works with a sequential stage in the human face detection.

For the sequential approach (Figure 6) when the current state fails, the next state is not reached, causing the UAV control to return to the previous state. However, for the parallel approach when the current state fails, the system follows the execution task flow, before reaching the overall system failure.

Besides, when using the FTA it is possible to measure the number of failures on parallel and sequential stages. In Table 1 are described the number of state transitions measured in track and human face detection sublayers respectively. It is clearly observed that the fault number in parallel approach is reduced, when compared to the sequential approach.

Table 1: Example of system faults evaluation with FTA.

Sublayer	Faults
Sequential Human Face Detection	145
Sequential Tracking	243
Both algorithms in parallel	42

These preliminaries values reinforce that the proposed methodology provides a significant reduction of the number of state transitions or indirectly a reduction of computational cust of the UAV control.

5 CONCLUSIONS

In this position paper we presented new strategies for UAV control, that uses a parallel hierarquical finite

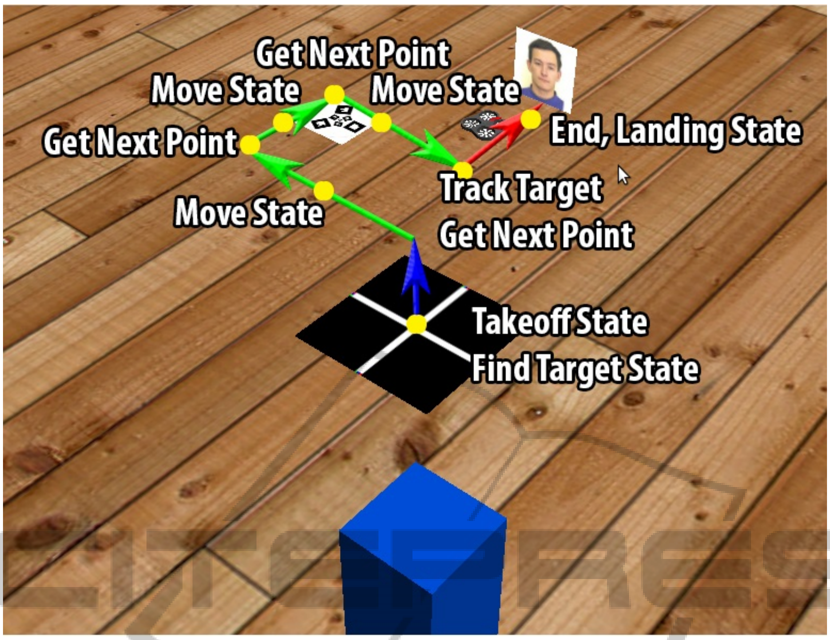

Figure 5: Example of a complete task execution.

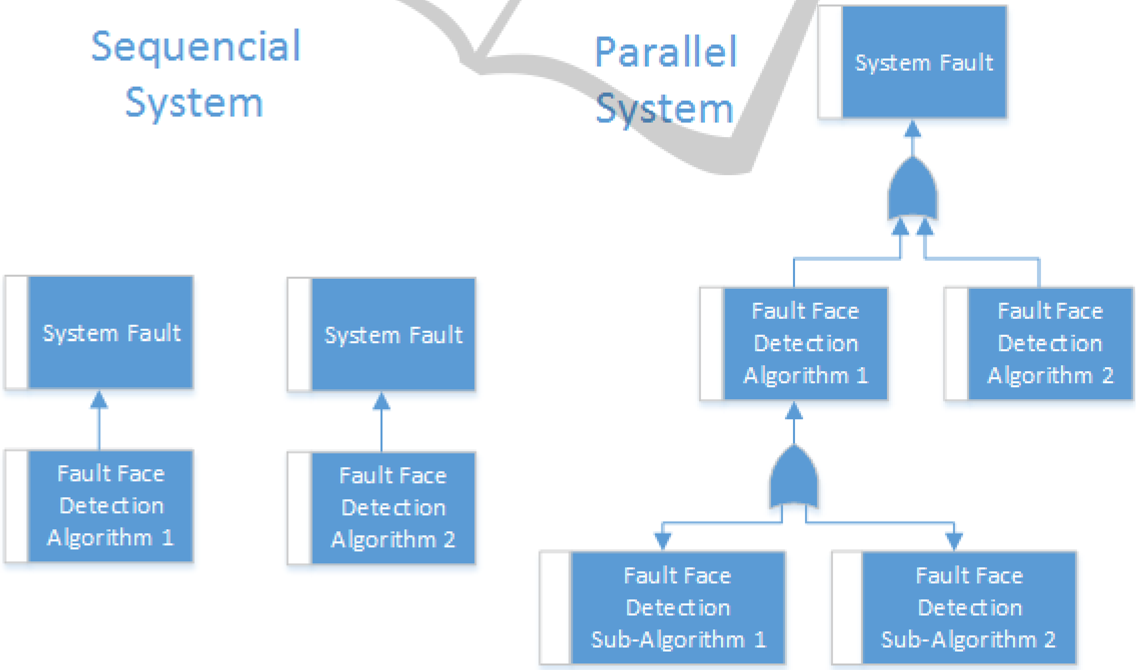

Figure 6: FTA for sequential and paralell implementation.

state machine in order to improve search and rescue tasks. The partial tests show that the proposed methodology (PHFSM) achieved better results when compared to the classical techniques of FSM UAV control, especially in improve the overall system failure (by FTA) and evidence of a significant reduction in the computational cost of the control process.

The proposed approach could provide improvements for UAV control due to the fact that the algorithm also offers flexibility in situations where there is no previous informations about the task to be tracked, increasing the system robustness to task failures.

Further works may include the implementation of the proposed strategies on a high level programming language in order to enable its operation in real time scenarios (including timing analysis) using a real UAV and also perform more comparisons with the latest techniques available in UAV control literature.

REFERENCES

G. De Micheli, Synthesis and Optimization of Digital Circuits, McGraw- Hill, Inc., 1994.

T.Villa, T.Kam, R.K.Brayton, A.Sangiovanni-Vincentelli, Synthesis of Finite State Machines: Logic Optimization, Kluwer Academic Publishers, 1997.

Branco, K., Pelizzoni, J., Oliveira Neris, L., Trindade, O., Osorio, F., and Wolf, D. (2011). Tiriba - a new approach of uav based on model driven development and multiprocessors. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1–4.

Cavett, D., Coker, M., Jimenez, R., and Yaacoubi, B. (2007). Human-computer interface for control of unmanned aerial vehicles. In *Systems and Information Engineering Design Symposium, 2007. SIEDS 2007. IEEE*, pages 1–6.

Changqing, Y. and Zhurong, W. (2013). Uav path planning using gso-de algorithm. In *TENCON 2013 - 2013 IEEE Region 10 Conference (31194)*, pages 1–4.

Dayuan, G., Zheng, M., and Hai, Z. (2013). Modeling and control of unmanned aerial vehicle using self-organizing map multiple models. In *Control and Decision Conference (CCDC), 2013 25th Chinese*, pages 4177–4182.

Devaud, J., Najko, S., Le Nahedic, P., Maussire, C., Zante, E., and Marzat, J. (2012). Full design of a low-cost quadrotor uav by student team. In *System Engineering and Technology (ICSET), 2012 International Conference on*, pages 1–6.

Ercegovac, M. (2000). *Introducao aos Sistemas Digitais*. Bookman.

Ericson, C. A. (1999). Fault Tree Analysis A History. In *Proceedings of The 17th International System Safety Conference*.

Girault, A., Lee, B., and Lee, E. A. (1999). Hierarchical finite state machines with multiple concurrency models. *IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems*, 18:742–760.

He, R., Bachrach, A., Achtelik, M., Geramifard, A., Gurdan, D., Prentice, S., Stumpf, J., and Roy, N. (2010). On the design and use of a micro air vehicle to track and avoid adversaries. *Int. J. Rob. Res.*, 29(5):529–546.

Huang, H. and Sturm, J. (2014). Tum package.

Jeon, J., Lee, H., Han, S., Lee, H., Lee, C., Kim, Y. B., and Choi, H. R. (2013). Design and control of an unmanned aerial vehicle (uav) based on the coanda effect. In *Ubiquitous Robots and Ambient Intelligence (URAI), 2013 10th International Conference on*, pages 129–129.

McCune, R., Purta, R., Dobski, M., Jaworski, A., Madey, G., Madey, A., Wei, Y., and Blake, M. (2013). Investigations of dddas for command and control of uav swarms with agent-based modeling. In *Simulation Conference (WSC), 2013 Winter*, pages 1467–1478.

Ping, J., Ling, A. E., Quan, T. J., and Dat, C. Y. (2012). Generic unmanned aerial vehicle (uav) for civilian application-a feasibility assessment and market survey on civilian application for aerial imaging. In *Sustainable Utilization and Development in Engineering and Technology (STUDENT), 2012 IEEE Conference on*, pages 289–294.

Quigley, M., Conley, K., Gerkey, B. P., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Y. (2009). Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*.

Rabbath, C. (2013a). A finite-state machine for collaborative airlift with a formation of unmanned air vehicles. *Journal of Intelligent & Robotic Systems*, 70(1-4):233–253.

Rabbath, C. (2013b). A finite-state machine for collaborative airlift with a formation of unmanned air vehicles. *Journal of Intelligent & Robotic Systems*, 70(1-4):233–253.

Sklyarov, V. and Skliarova, I. (2008). Design and implementation of parallel hierarchical finite state machines. In *Communications and Electronics, 2008. ICCE 2008. Second International Conference on*, pages 33–38.

Trimble, G. (1987). A multiprocessor system for auv applications. In *Unmanned Untethered Submersible Technology, Proceedings of the 1987 5th International Symposium on*, volume 5, pages 208–219.

Wang, Y., Yan, W., and Yan, W. (2009). A leader-follower formation control strategy for auvs based on line-of-sight guidance. In *Mechatronics and Automation, 2009. ICMA 2009. International Conference on*, pages 4863–4867.