Journal Homepage: -www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

ISSN NO. 2320-5407

Article DOI:10.21474/IJAR01/3196
DOI URL: http://dx.doi.org/10.21474/IJAR01/3196

RESEARCH ARTICLE

DEVELOPMENT OF INTELLIGENT HYBRID ARCHITECTURE FOR AUTONOMOUS UAV.

Mostafa Moussid, Adil Sayouti and Hicham Medromi.

……………………………………………………………………………………………….....

Manuscript Info

……………………….

Manuscript History

Received: 03 December 2016
Final Accepted: 12 January 2017
Published: February 2017

Key words:-
Intelligent UAV, Control architecture; Autonomous navigation; Hierarchical control architecture; Reactive control; Deliberative control; Intelligent control systems, Multi-agents systems.

Abstract

………………………………………………………………

The emerging area of intelligent unmanned aerial vehicle research has shown rapid development in recent years and offers a great number of research challenges for distributed autonomous robotics systems. However, with the development of more complex robots that must operate in uncontrolled and dynamic environments, an autonomous UAV is understood to be intelligent robot capable of performing complex operations in dynamic, real-world, uncertain, sometimes hostile environments without any explicit human control over its movements and must constantly reconfigure itself to adapt to the external conditions and its own goals. To provide the aerial vehicles with these capabilities, robot control architecture is necessary. The challenge is to develop a UAV control system capable of obtaining intelligent, suitable responses to changing environments and adapt the software to the current situation. The definition of control architecture to manage these reconfigurations becomes of paramount importance for increasing the level of autonomy and successful navigation of such robots. The Control architectures define how these abilities should be integrated to construct and develop an autonomous navigation with little or no human intervention. Numerous intelligent control architectures do exist in the literature for mobile robots. However, none of these are specifically targeted at providing the required support for a wide range of UAV missions. Operations of UAV require robust methods for dealing with emergency scenarios such as performing forced landings and collision.

In this work, we firstly study and analyze the different architectures adopted in the literature, on the basis of their flexibility, ease of implementation, reactivity, robustness, efficiency and other architecture specifications. This analysis led us to propose intelligent and hierarchical control architecture, decentralized, generic and reusable, applicable to autonomous aerial vehicles flying in an unknown, dynamic and potentially hostile environment; our control is a hybrid architecture based on multi-agent technology (MAS), which can handle unpredictable events in an unstructured world, composed of distributed, independent and asynchronous behaviors. In addition, it integrates multiple knowledge representation approaches to build cognitive models and intelligent systems that significantly advance the level of intelligence we can achieve. Our architecture is a family of intelligent control systems, hybrid and decomposed into flexible autonomous subsystems, its containing elements of sensory processing, world

Corresponding Author:-Mostafa Moussid.

modeling, localization, Mission planning & high level Expert system, and action processes to achieve or maintain its goals. The reactive part will guarantee that simple tasks are achieved under time constraints (real-time) while deliberative part will grant planning and reasoning. The whole architecture assures the safety of the UAV and the environment, so it provides the mechanisms to deal and reduces the impact with hardware and software failures of onboard.

……………………………………………………………………………………………………....

Introduction:-

In the past several years, the rapid development of unmanned aerial vehicles equipped with autonomous control devices have become a real center of interest, and different kinds of autonomous vehicles have been studied and developed all over the world. UAV are mostly being used for military applications, but with the evolution of avionics technology (The newest sensors, microprocessors, lighter and propulsion systems are smaller and....), a huge market in civil applications is now emerging; major advantages are offered when used in emergency situations in disaster monitoring and contaminated environments (After a natural or industrial disaster such as wildfires, an active volcano, earth-quakes, a flood, or a nuclear disaster). Indeed, UAVs are better suited for dull, dirty, or dangerous missions than manned aircraft. The low down side risk and higher confidence in mission success are two strong motivators for the continued expansion of the use of unmanned aircraft systems. However, to realize these applications, seamless operation of UAV will be required. Increasing the levels of onboard autonomy will help to address this requirement. Additionally, increased autonomy also reduces the impact of onboard failures, potentially lower operational costs, and decrease operator workload. Autonomous intelligent control is execution of a given control strategy without human intervention and in an optimal manner, and capability to adapt autonomously and in a fast and efficient manner to a new set of circumstances on-line sensing, information processing and control reconfiguration[1]. It is essential for an autonomous intelligent unmanned aerial vehicle system to sense the surroundings, perceive the working environment, make decisions, plan a path and execute appropriate reaction using the sensors information. The design of perception, navigation, planning and control systems is a crucial step in the development of such autonomous flying machines. It must be able to fly within a partially structured environment, to react and adapt to changing environmental conditions. To provide the UAV with these capabilities, control architecture is necessary; it's one of essential part of robotics system development. Architecture is the structure that identifies, defines, and organizes components, their relationships, and principles of design; the assignment of functions to subsystems and the specification of the interfaces between subsystems.

Indeed, the main components that any autonomous robot has to include are the following: a deliberative component enabling it to reason about how to achieve high level goals, a reactive component to deal with the environment; and perception and actuators to deal with the physical world.

This work deals with architecture for controlling a mobile robot in accordance with the main objective in robotics, which is to develop a UAV control system capable of intelligent and suitable responses to changing environment. Our control architecture has to possess a number of desirable features: flexibility, real-time response, coherent behavior, adaptability, fault tolerance, easy design and granularity. There are good reasons for organizing the control of large systems in a hybrid distributed hierarchy. Among these are: deeper understanding facilitated by the hierarchical structure, reduction in complexity of communication and computation, modularity and adaptability to change, robustness, scalability and autonomy. UAV's actions are both the result of intelligent reasoning from superior and executive decision-making layers knowing the situation and actions that respond directly to environmental stimuli [2].

The key contributions of this work are tow fold: Firstly comparing each of these control architectures on the basis of their flexibility, ease of implementation, reactivity, robustness, efficiency and other architecture specifications. Secondly we propose a distributed architecture for autonomous unmanned aerial vehicle, in order to provide a system with several types of intelligence (Reactive intelligence, Deliberative intelligence and creative intelligence). An illustration of our study will be given in an application of control of an autonomous hexarotor developed by the team architecture of systems, to the ENSEM of Casablanca.

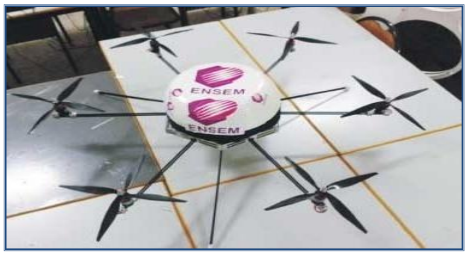

Figure 1:- A picture of the developed hexarotor (EAAS\ENSEM)

The paper remainder is organized as follows: In the next Section, an overview of architecture for UAV control is presented while section III gives and describes the proposed UAV control architecture based on a multi-agent system, for the autonomous navigation, allowing a UAV to navigate in an unknown and hostile environment. This architecture is being developed in a modular and incremental way allowing the incorporation of several techniques of mapping, localization and path planning independently of the reactive strategy. Finally, some conclusions and future works are presented in Section IV.

Overview of Architectures Control:-

Autonomous systems are typically quite complex, it is expected that the robot will be able to achieve high level goals while interacting with complex and dynamic environments. The robot must deal with its own dynamics, noise and uncertainty and has to be reactive to unexpected changes.

Well-designed software architectures can provide concepts, constraints and tools that make it easier to design, implement, and debug such systems. There are many ways to structure a robot, yet everyone will fall into one of the basic architecture control systems that can be found in technical literature: hierarchical/deliberative, reactive/behavior-based and hybrid. The categories differ largely in how they handle task achievement and in their reactivity. The aim of this section is to describe the three major paradigms of control strategies for a completely autonomous navigation [3]:

•Deliberative strategy: look ahead, think and plan, then act.
•Reactive strategy: no look ahead, react (time-scale)
•Hybrid strategy: think slowly, react quickly.

Brief descriptions of the above mentioned control strategies, the significance, advantages and drawbacks of the architectures are presented, discussed and compared with each other in the following paragraph.

Historical control architecture:

An overview of existing architectures in robotics is presented in this paragraph to identify relevant architecture design methodologies and give the benefits and shortcomings of different architectures. Robot architectures and programming began in the late 1960s with the Shakey robot at Stanford University. Generally, the literature on control architecture proposes three types of control architecture [4]. Their conceptual representation is presented in Figures 2/3.

Reactive *VS* Deliberative Control Architectures:

Reactive architecture is a fast response sense-act type of organization, which is particularly suitable for a dynamic environment. Since a planning stage is absent, it has low computational burden, but has relatively low level intelligence. Many modules connects several inputs sensors/actuators, each module implements a behavior. These behaviors are called \reactive" because they provide an immediate output of an input value, and cognitive otherwise [5]. Reactive architecture serves best when the real world cannot be accurately characterized or modeled. Very often, uncertainty, unpredictability and noise from the world cannot be removed. Reactive architectures were developed in response to this difficulty (figure 2-a).

An alternative to the reactive paradigm, for increasing decision capabilities a second type of architecture, called deliberative, is proposed (figure 2-b). This architecture presents generally three independent levels [6]: sense, plan and act. This decomposition facilitates considerably the development of each part of the architecture. In this kind of architecture, each layer provides sub-goals to the layer below. They include a global world model which is modified and updated through perception. Based on this world model, planning and reasoning for making decision are carried out that result in actions to be performed by the UAV. As reasoning takes a significant amount of time, it becomes a bottleneck in the architecture.

Deliberative architecture is a high level, intelligent, top-down, suitable for a static environment. All the sensing data is utilized to set up a world model. Based on the world model, the planner then generates an action sequence.

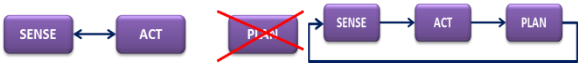

Figure 2: (a) Reactive architecture (b): Cognitive architecture

As shown in table 1, the deliberative layer uses a purely symbolic representation and the reactive layer is free to choose its representation model. The reactive layer is usually represented in a way that facilitates the translation into actuators commands. Thus, there is a need for a common world model or knowledge system which shares information between these layers.

table 1:comparison with the deliberative and reactive architectures.

Deliberative	Reactive
⟵	⟶
(Purely symbolic)	(reflexive)
Speed of response ⟶	
Predictive capabilities ⟵	
Dependence on accurate, complete world models ⟵	
- Needs internal representation - Slower response - High-level intelligence -Capable of learning / prediction - Finds strategic solution	- No internal representation - Real-time response - Low-level intelligence -very fast in terms of motions and computation
Limitations	
– Planning requires search through potentially all possible plans ⇒ these take a long time, – Requires a world model, which may become outdated, – Too slow for real-time response	- No/minimal state - No memory of the world - Unable to plan ahead - Unable to learn - Difficulty in coordination among the behaviors.

Hybrid Control Architectures:

Neither the purely reactive scheme nor the purely deliberative architectures perform well when performing complex tasks, because of difficulties in modeling the world and relying too much on inadequate sensors. Hybrid Architectures aims to combine the best of both Reactive and Deliberative approaches, trying to reduce the restriction on the scope of each of these approaches: reactivity, so they can respond in real-time to changes in dynamic environments and deliberation, so they can plan and provide the adequate sequences of actions needed to achieve the goal using higher reasoning and an internal knowledge representation of the world, so the goals of the robot can be achieved efficiently. Thus, a hybrid paradigm connects deliberation and reaction reducing the response time of the robot to environmental changes and performing plans (figure 3). Control architectures for complex autonomous mobile robots have largely settled on hybrid architectures for their suitability at dealing with the opposing forces of planning and reactivity [6].

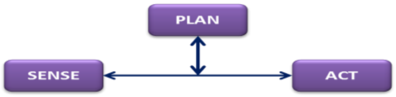

Figure 3: Hybrid architecture

The hybrid control architecture specifications, each specification is described in Table 2.

TABLE 2: ANALYSIS OF THE CONTROL SYSTEMS ARCHITECTURE.

Specifications	*Deliberative* *RAP*	*Reactive* *BERRA*	*Hybrid* *AURA / SSS*
Goal oriented	VG	NG	G
Flexibility	VB	VG	VG
Ease of application	VB	VG	G
Reactivity	VB	VG	G
Optimal operation	VG	VB	G
Task learning	VG	M	M
Robustness	NG	G	VG
Planning	VG	NG	G
Efficiency	NG	VG	VG

Discussion of control architectures:

The robotic researchers all agree that control architectures should be deliberative, reactive, robust, generic, modular, and intelligence.

An autonomous robot is understood to be an intelligent machine capable of performing tasks in the outside world by itself, without any explicit human control over its movements. Also, an intelligent machine is taken to be a machine able to extract information from its environment and use knowledge about its world to move safely in a meaningful and purposive manner.

To achieve a comprehensive control system, robot needs more abilities that exceeds deliberative and reactive paradigms such as perception and world representation ability to enable information gathering and processing, fast reacting for static or dynamic obstacle avoidance, world modeling ability to insure the robot to localize itself relative to the environment, inference and decision making ability to make reliable decisions based on that particular information.

Various control architectures for autonomous navigation of mobile robot have been described and developed for building intelligent systems. Some of these (such as SOAR [7] and Expert Systems architectures [8]) are designed to model high-level cognitive elements of human reasoning. However, they do not address the low-level details of perception and real-time behavior in uncontrolled and dynamic environments. Others (such as Subsumption [9] and its many derivatives) have been designed to model low-level reactive behaviors. However, these do not address the high-level elements of cognition, knowledge representation, reasoning, and planning. Still others (such as AuRA [10] and RCS [11]) are hybrid architectures designed to combine high-level planning with low-level behaviors. The review of this architectures showed that the hybrid scheme has the best performing supervisory control architecture and it is more prosperous and promising dealing with unknown, dynamic navigation problem.

The review of this architectures showed that the hybrid scheme has the best performing supervisory control architecture and it is more prosperous and promising dealing with unknown, dynamic navigation problem.

After analyze these architectures, a list of important features has been defined. They include the way the architecture is built, its capacity to deal with real-time, the manner in which coordination is performed as well as the method used to do so, communication requirements, adaptability to different conditions and environments, capability to detect and repair failures, scalability, granularity and the level of abstraction used to program the components of the architecture.

The first step in our study for the conception of our architecture is to identify and analyze the qualities we want the architecture to have. The main objective is to use and provide a system with several types of intelligence to evaluate the performance of various algorithms in operational conditions and to study their robustness. In order to fulfill this objective our architecture requires much more qualities than : robustness, security, modularity, to ease maintenance, adjustable autonomy, and self-reconfiguration, re-usability, communication, concurrency, interoperability, parallelism and data flow synchronization. Of course, it is obvious that in a first time, our architecture must have a maximum of functionality that can contribute to the global autonomy of the systems (figure 4).
Here are described the main qualities we want to provide to our architecture [12]:

To ensure intelligent behaviors: The intelligence results in perception, reasoning and action capacities. The perception translates acquired information into knowledge on the environment. The decisional system generates plans of operations that describe actions to undertake in order to reach objectives of a mission and to react in the face of asynchronous events. The amount of intelligence is closely linked to the different kind of environments in which the robot has to evolve, as well as to the complexity of tasks it has to fulfill [13]. The intelligence of the robot can be situated in several levels. The first one is associated to the local environment of the robot. Thus, in the case of an unknown environment, it is indispensable to endow the robot of an intelligent behavior allowing it to avoid obstacles met on a nominal path. This behavior relies on an on-line control of this path. The second level of intelligence is situated at the control level of the robot's behaviors. It is therefore necessary to have a mechanism that allows changes of strategy in order to adapt the robot's behavior to external events. In other words, this level of intelligence allows adopting adaptively an adequate behavior of the robot from evaluations of its internal state and those of its environment (We like that our architecture provide a system with several types of intelligence: reactive intelligence, deliberative intelligence and creative intelligence).

Cognition: From perception to action to learning: Cognition is the key to how robots will deal with unconstrained environments, learn from their encounters, and apply the new knowledge to similar situations in the future. Cognition is the process by which intelligent entities receive and handle information [14]. It is not one discrete thing, but a synergistic combination of multiple capabilities. For robotics, cognition is a combination of perception, understanding, motion planning, and automated learning. Improved cognitive ability means robot can work in diverse, dynamic, and complex environments autonomously and improve performance by learning from experience.

To ensure rapid sensing and reactivity to the environment: The mobile robot has to be able to manage external asynchronous events in real time so as to respect the dynamics of the environment (the capability to sense external events rapidly). An external event can have several origins: presence of an unforeseen obstacle, sudden breakdown, request from another robot, etc. The reactivity generally implies a real time processing of these events. The real time implies constraints on the reply delays and on some information flows (the ability to respond within a limited time period to external events occurring in its domain). These constraints depend on the equipment type and the way those events are managed. Thus, the command system has to include the notion of priority and urgency of event processing.

Self-reconfiguration: this ability is very important. First, in case of failure of one or more modules, or when the chosen modules are no more able to fulfill the designed task, the system must self-adapt and find a new module or series of modules to efficiently do the task. Second, the architecture must fit the needs of the users and adapt itself to his change (from a full remote control interaction to a supervised remote control one for instance). The architecture must also update and change the data exchange between the modules depending of the circumstance

To ensure modularity and composability: The modularity of the control architecture of a mobile robot is achieved by the decomposition in modules that can be developed, implemented, and realized separately. The ability to be reconfigured and to be extended are two characteristics that allow any command system to evolve by the addition of new functionalities and the endowing of a flexibility of adaptation. The main advantage of distributed controlled robots and subsystems is the decentralized task execution by the system components. This way, properties for the design of flexible control architectures like modularity, fault-tolerance, integrability and extendibility are easy to obtain, further it is possible to use the concepts of distributed knowledge and decentralized world representation.

Maintenance: the architecture must be designed to ease the maintenance. Especially the reconfiguration and the re-launch of a module must be possible while the system is running and without interrupting the experiments.

Moreover the module must be able to record and save online their internal data and their interfaces so that in case of failure, it is possible to identify the module responsible of this dysfunctional execution. The modules can also be tested alone, their input and their output perfectly controlled.

To manage interruptability: Higher priority environmental threats must be able to interrupt normal operations of the robot. The robot must also be able to resume its original task after responding to the threat. Therefore, the robot's control system must be able to halt an existing control process and later resume that process after completing the new control cycle initiated by the higher priority task.

To manage Fault-tolerance: one of the most primordial aspects in robotic control architecture is the robustness to the execution failure. All must be done in the architecture to avoid the system stop working in the correct way. Whatever the circumstances, the system must be as fault-tolerant as possible. In others words, the failure of a part of the system, e.g. of one or more modules, should not be synonymous with the failure of all the architecture, whatever the nature of this failure (lack of memory, data reading mistake, segmentation fault, etc.) [15]

To develop an architecture capable of integrating and validating new technologies, such as different kinds of actuators and sensors.

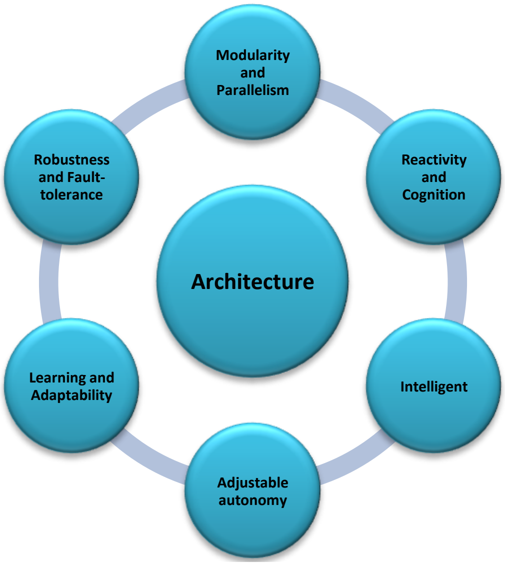
Modularity and Parallelism
Reactivity and Cognition
Intelligent
Adjustable autonomy
Learning and Adaptability
Robustness and Fault-tolerance
Architecture

Figure 4. Main features of our architecture.

The Proposed Control Architecture:-

In this section, we propose hierarchical/intelligent control architecture for an unmanned aerial vehicle (UAV), including a deliberative part and a reactive part. The proposed architecture aims to supply autonomous behavior in unknown environment considering the uncertainties of the UAV's sensors and mainly the possibility of existence of mobile or stationary obstacles which are not expected in the navigation plan. The specificity of the control architecture that we propose, is the organization between perception (sensors), making decision and action (actuators) around the loops executed at different time scales: real-time loop closely linking sensors and actuators, and another loop taking place on a slower time scale that manages one hand the representations of the environment that builds drone, and others from various events that can happen to unforeseen moments. Hybrid architecture is used to control our UAV because he has both desirable properties: reactivity, so they can respond in real-time to changes in dynamic environments and deliberation, so they can plan actions ahead in time.

Our architecture is a family of intelligent control systems, distributed and decomposed into flexible autonomous subsystems, its containing elements of sensory processing, world modeling, localization, makes decisions, creates plans, and controls actions to achieve or maintain its goals as shown in figure 5.

The flow of information between the World Model and Mission planner is bidirectional. While the World Model provides Mission planner with information regarding the state of the external world, Mission planner provides the

World Model with information about the state of the task. This enables the World Model to represent what task is in progress, and what commands are currently being generated at each echelon in the Mission planner hierarchy. Mission planner also informs the World Model about plans for possible future actions. The World Modeling processes can then simulate the probable results of these possible future actions, computes an estimate of cost, benefit, and risk. This enables Mission planner to choose among alternative future courses of action. The flow of information between the World Model and Sensory Processing is also bi-directional. While Sensory Processing keeps the World Model updated, the World Model provides context and predictions that assist Sensory Processing in the interpretation of sensory data [16],[17].

Our architecture consists in five blocks organized around a sixth: perception processes, representation and world modeling, localization, mission planning, action processes and expert system. The core of the architecture relies on Expert system.

Our architecture, nommed “EAAS\ENSEM”, is potentially applicable to diverse robotic systems that could include aircraft, spacecraft, ground vehicles, surface water vessels, and/or underwater vessels. EAAS\ENSEM (see figure3) includes an integral combination of five coupled agents: a dynamic planning engine, a behavior agent, world model of knowledge, expert system agent and a perception agent. The perception and dynamic planning engines are also coupled with a memory in the form of a world model. EAAS\ENSEM is intended to satisfy the need for two major capabilities essential for proper functioning of an autonomous robotic system: a capability for reaction to unanticipated occurrences and a capability for re-planning in the face of changing goals, conditions, or resources.

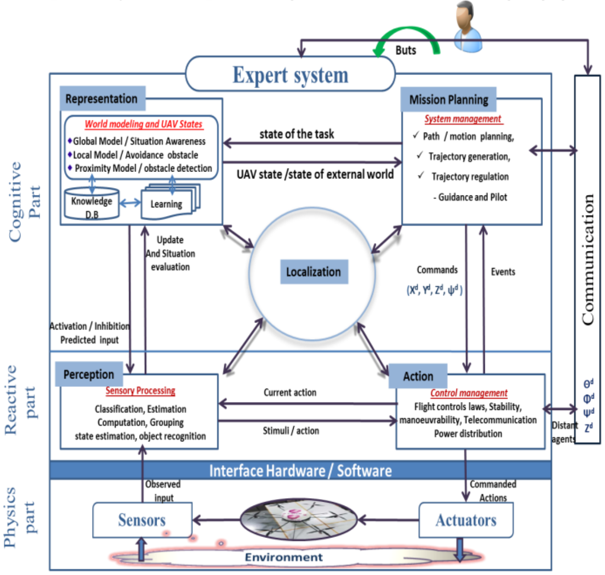

Figure 5:- Overall architecture of UAV (EAAS/ENSEM)

Fundamental capacities of our architecture encompass autonomy, Distribution of data and control, Robustness and reliability, Flexibility and Scalability, Real-time response, extensibility, coherent behavior, reliability and parallel execution. The architecture features a useful organization structure for high-level skills and offers flexible construction options for low-level behavior hierarchies.

Bottom-Up description of the Components:-

The five basic of processing modules from which our architecture is built, as can be seen in figure 5 are:

Expert System module: Our aim is to build a real-time expert system to make intelligent inferences from the environmental data. It must employ an efficient control strategy and must meet the specifications listed in the previous section. This module defines the meta-behavior of the UAV. It has the information about the overall mission objectives and constraints. This information, in conjunction with the sensory and situational awareness, is used to make appropriate decisions as trade-offs between the mission success and vehicle survivability. The decisions reached are relevant to achieving assigned missions efficiently and safely. it acts as an interface between the Mission Planner and the rest of the architecture. It ensures that changing the operating modes of the aerial robot is done in the correct step sequences. It also reports the execution status of the current action to the Mission Planner. It also monitors some safety measures regarding the rules of the competition and conflict resolution. Also, this layer is responsible for collision avoidance, mission retaking, data analysis, fault diagnostics, and goal reassessment. It manages the data flow and ability to carry out fault detection/diagnosis procedures and accommodate faults (in the actuators and sensors) so as to assure an acceptable performance level (fault tolerance ability). It also manages the asynchronous events coming from the environment. Moreover, it allows adapting an appropriate behavior by aggregating several behavior modules in front of special situations. In other words the expert system here should be part of a conventional feedback loop with a process, a controller, a parameter/state estimator, a fault detector/ isolator and a supervisor.

Mission planning (The highest level): This level can be defined as the “driver or cognitive” of a UAV that comprises various autonomy-enabling functions to achieve assigned goals. This is the hierarchical level of this architecture where the modules in Part deliberation which decompose the mission in executable tasks and decide what action to perform based on his knowledge of the environment and the internal state UAV. It takes inputs from the censoring system and uses targeting information (mission goals) to make appropriate decisions at its high level and to generate Autonomous, path planning, reference trajectories and commands for the Automatic Flight Control System at its low level. The UAV system must have the capability to plan and replan its own flight path. This results in the requirement for a high level computing environment where flight planning algorithms can be run. At this level, the important design challenge is to arrive at efficient algorithms (search optimization) for on-line generation and execution of a motion plan that enables the UAV to move to a desired location and perform a given task, even while avoiding obstacles. Given different way-points along a desired path, the objective of the autonomous trajectory generation system is to fit a feasible trajectory through the way-points, given the UAV and control input constraints. Many of the trajectories can be calculated off line and stored. However, in the presence of hazards and subsystem or component failures, the trajectory may need to be reconfigured on-line to reflect the new environment, or the new achievable dynamics, or both. Indeed, In the event of system faults, the UAV must have the capability to reconfigure itself and re-plan its flight path in a fail-safe manner. The control system will need to generate and execute the movement plan in near real-time and in an environment with a complex topology and with dynamically changing and uncertain components.

We have broken down in our architecture that level in tow modules with specific functions. These are prioritized and contribute to dissociate the different tasks in clearly identified functions: path planning and trajectory generation [18].

Path Planning: determining an optimal path for UAV to follow while meeting mission objectives and constraints, such as obstacles or energy requirements. The role of this layer is to generate the motion plan for the overall mission, and compute spatial and other constraints needed for the design of the desired trajectories. Many of the routes and constraints can be computed off-line to cover different situations. The constraints are computed in the form of safe set boundaries around the waypoints. In our architecture, we have included an automated planning systems onboard can potentially improve mission efficiency and reduce the need for laborious input from a ground based human operator. Dynamic path planning refers to onboard, real-time. He receive a description of the state of the world and a goal, and then in turn compounds produced plans of actions and implementing rules sequences corresponding to the realization of this objective. The supervisor associates sends him to realize the objective, and then monitors the execution of the plan in light of the events produced by the execution or by a changing environment. The adopted strategy is that instead of giving the mobile system a path to follow, it is more concerned to grant him a goal and let the control architecture independence in defining the optimal path to follow. This can be expressed in different forms: set of points in rallying, in a specific order, to reach position. This strategy needs an internal representation of the environment that is to define places of space in which perceptions are the same, and associate an action with each of them.

Trajectory Generation**:** The Trajectory Generation is determining control maneuvers to take in order to follow a given path or to go from one location to another. this level of planning receives as input, the paths to follow, and provides more accurate trajectories, taking into account local information from the environment. The aim of this layer is to fit a feasible trajectory through the way-points. A trajectory generator has the role of computing different motion functions (reference position, reference heading, etc.) that are physically possible, satisfy UAV dynamics and constraints, and can be directly used as reference trajectories for the flight controller. Reference trajectories can be preprogrammed and uploaded, or generated in real time onboard the UAV. Trajectory generation is commonly based on minimization of a given criterion (e.g. time between the way points, energy consumption, or low exposure to known stationary threats), and can be generated either on-line or off-line. In the case of failures, upsets, or other anticipated or unanticipated events, the path planning layer automatically reconfigures the desired path by modifying the waypoints. This level is considered the level of refinement, and its existence is essential. Indeed, the upper level, the representations of the environment and tasks are necessarily incomplete because they are too abstract, they can't express in particular all interactions with the environment of the drone, the intrinsic parameters of the UAV system are generally fixed, as against the constraints of the environment are often vague and scalable.

Environment modeling and Learning: The world model is the system's internal representation of the external world. It acts as a bridge between sensory processing and behavior by providing a central repository for storing sensory data in a unified representation (Knowledge database). It decouples the real-time sensory updates from the rest of the system. During the mission, the modeling functions will help incrementally build models of the environment, through aggregation (or rather merge) successive models developed from sensory data corresponding to the various acquisitions.

World modeling processes maintain a rich and dynamic database of information about the world in the form of images, maps, entities, events, and relationships at every level. Other World modeling processes use that information to generate estimates and predictions that support perception, reasoning, and planning at every level.
We distinguish following spatio-temporal three criteria:

The instantaneous patterns: are constructed from common sense data and values of observed, estimated, and predicted attributes and state variables (corresponding to a given sensory acquisition).

Local models: the result of the merger of several flash patterns acquired in the same topological location. It's a short term memory containing iconic and symbolic representations of geometric entities and events that are the subject of current attention.

Global models: The global models are maintained update by a local models modeling process, aggregations of all local models built during a given mission. The global models includes models of portions of the environment, images, maps, models of entities, events, rules, task knowledge, abstract data structures, and pointers that represent relationships, and a system model that includes the intelligent system itself. For our architecture, this part can be defined as the process of data acquisition, data analysis, and extraction and inference of information about the vehicle's states and its surrounding environment with the objective of accomplishing assigned missions successfully and safely (figure 6).

It creates and keeps the knowledge database current and consistent (of maps, situations, relationships, and knowledge of task skills and laws of nature and relationships among them). It gives a best estimate of the state of the world to be used as the basis for predicting sensory feedback and planning future actions (learning). It predicts sensory observations based on the estimated state of the world. It simulates results of possible future plans based on the estimated state of the world and planned actions [19].

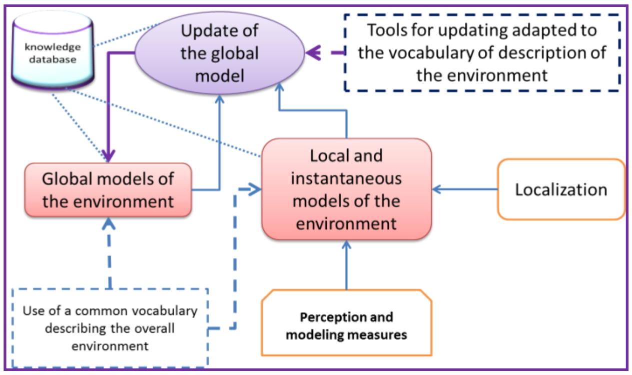
knowledge database
Update of the global model
Tools for updating adapted to the vocabulary of description of the environment
Global models of the environment
Local and instantaneous models of the environment
Localization
Use of a common vocabulary describing the overall environment
Perception and modeling measures

Figure 6: General schema of the modeling of the environment

Learning has the advantage that it allows the agents to initially operate in unknown environments and to become more competent than its initial knowledge alone might allow. Learning agents have the potential to learn about their environment and about other agents through adding to the knowledge held in their working memories.
For some problems, it is desirable to create agents that are capable of more fundamental learning: where the internal structure and processing of the agents adapt to changing circumstances. There are two techniques commonly used for this: neural networks and evolutionary algorithms such as the genetic algorithm (figure 7). It is possible for either an individual agent to be modeled using a neural network, or a whole society to be represented by a network, with each neuron given an interpretation as an agent (although in the latter case, it is hard to build in all the attributes of agents usually required for multi-agent modeling).[20]

Localization: Localization is a technique that permits the robot to give an answer to this question [21]: Where am I? It is the main point in any success physical interaction. For many applications an imperative need for UAV autonomy is the ability to self-localization in the environment, especially for extended periods of time, when estimator drift tends to destroy alignment to any global map. Indeed, precise localization is crucial in order to achieve high performance flight and to interact with the environment.

Increasing innovation in the field of electronic communications has led to a current trend of utilizing sensing system such as Global positioning system (GPS), radio technologies or vision-based solutions for localization of UAVs. Fusing data from different sensors helps to improve performance of the overall sensing system. For aerial navigation outdoors, fusion of GPS measurements with INS measurements by means of filtering techniques delivers the level of localization precision required by UAV missions. The proposed architecture provides routines for corrections in the positioning through the combination of information of the Mapping, Sensing and Location modules.

Perception processes (Filter, detect, recognize, and interpret): Perception in robotics means the ability to collect process and format useful information to the UAV to act and react to the world around. It covers the acquisition components, filtering, detection, segmentation, tracking, identification and interpretation. Strong perceptual abilities are a basic requirement for a robot working in an environment that was not specifically designed for the robot. Such a surrounding might be completely unknown or may change over time, so that a model cannot be provided to the robot a priori. The perception includes obtaining data about the vehicle and its environment and extracting useful information from the data. The Perception can be further divided into various functions on different levels such as mapping, obstacle and target detection, state estimation, object recognition and Situational Awareness (the perception of elements in the environment within a desirable volume of time and space, the comprehension of their meaning, and the projection of their status in the near future).

The sensory processing is a set of processes by which sensory data interacts with a priori knowledge to detect or recognize useful information about the world. Sensory processing accepts signals from sensors that measure properties of the external world or conditions internal to the system itself. Correlations between sensed observations and internally generated expectations are used to detect and classify entities, events, and situations. Differences between sensed observations and internally generated predictions are used to update the knowledge database.
Most people would only judge a robot to be truly intelligent if it perceives its environment, understands what is happening around it and acts accordingly (A robot that moves through an environment and interacts with it has to know what is going on around it, where it is, where it can go, and where objects necessary for its task are located). The correct interpretation of raw sensor data is often a crucial part when one aims at applications in the real world. A robot must be able to understand its surrounding, in order to work in it and interact with it. Without appropriate sensors a robot is very restricted in what it can achieve and is only able to work at very specific tasks. The topic of this module is therefore the interpretation of low-level sensor information and its application in high-level tasks.

Action (Flight Control low) : For UAV, the design of flight controllers low consists of synthesizing algorithms or control laws that compute inputs for vehicle actuators to produce torques and forces that act on the vehicle in controlling its motion (position, orientation, and their time derivatives). At this lowest level, we have the actual interaction with the physical plan: this is sometimes referred to as skill or reflexive level, and includes the traditional control functions (stabilization, regulation, commands tracking). The aim is to convert a trajectory into orders to be performed by the action. At this level, the desired role of the inner-loop controller is to assure rapid stabilization of the overall system in the presence of failures, control input and vehicle constraints, and improve accuracy of vehicle models through on-line learning. A hierarchical flight controller uses a system based on the nonlinear model of rotorcraft unmanned aerial vehicles (UAV) and considers a system's non linearity's as well as coupling between the rotational and translational dynamics. By exploiting its structural properties, the standard mathematical model of rotorcraft UAVs has been transformed into two cascaded linear subsystems that are coupled by a nonlinear interconnection term.

In this part, we present the main steps for designing a hierarchical flight controller using the inner and outer-loop control scheme: when the flight path is laid out, a flight control system is required so that the UAV can follow the planned flight path and execute the mission. Control inputs are generated based on the reference paths and the current states. The flight control loop generates actuator signals for the control surfaces and thrust vector. The set points for low-level stabilizing controllers whose function is to maintain the vehicle in a stable state and to follow accurately the commanded trajectory are provided. After synthesize control laws for each subsystem, there by resulting an outer loop with slow dynamics that controls the position and an inner loop with fast dynamics that controls the orientation. The asymptotic stability of the entire connected system is proven by exploiting the theories of systems in cascade. The resulting nonlinear controller is thus easy to implement and tune, and it guarantees the asymptotic stability of the closed-loop system.

Physical layer: Finally the physical layer represents the physical part of the robot, i.e. the articulated mechanical system and actuators to move the robot. This constitutes the basis on which the entire architecture is built.

The hardware link agent is an interface between the software architecture and real robot. Changing the real robot require the use of a specific agent but no change in the overall architecture.

Our architecture is a reference model architecture that provides a theoretical foundation for designing and integrating intelligent systems software for unmanned aerial vehicles (how their software components should be identified and organized). It prescribes a hierarchical control principle that decomposed high level commands into actions that employ physical actuators and sensors. Each module of our architecture is capable of accepting and decomposing task commands with goals into actions that accomplish task goals despite unexpected conditions and dynamic perturbations in the world. The architecture give plan on a model of the world rather than planning directly on processed sensor output. This may be accomplished by real-time sensors, a priori information, or a combination of the two in order to create a picture or snapshot of the world that is used to update a world model.

We note an interesting link between the desirable properties of intelligent control architecture for complex systems requiring a large degree of autonomy and the Multi-agent systems. To fulfill these requirements, we decided to use a multi- agent's formalism that fits naturally our needs. The Multi-Agent System paradigm is one of the most promising approaches to create autonomous, open and dynamic systems, where heterogeneous entities are naturally represented as interacting autonomous agents, who can enter or leave the system at will. In accordance with these proprieties, the multi-agent system is suitable for developing the control architecture of a UAV since it has inherent characteristics that are also desirable for architecture and offer many potential advantages. The fact that the architecture is a multi-agent system provides flexibility in terms of the software level. This architecture will have reactive and deliberative agents at least. The reactive agents will guarantee that simple tasks are achieved under time constraints while deliberative agents will grant planning and reasoning. The whole architecture must assure the safety of the UAV and the environment, so it should provide the mechanisms to deal with hardware and software failures.

Agents' description and Task distribution:-

One recent trend in autonomous robotic architectures has been the focus on multi-agent hybrid layered systems. Unlike the hierarchical and centralized structures, a distributed multiagent system can integrate its component capabilities to solve real-time complex problems.

Fundamental capacities of our architecture encompass modularity, encapsulation, scalability and parallel execution. To fulfill these requirements, we decided to use multi-agent formalism that fits naturally our need for encapsulation in independent, asynchronous and heterogeneous modules. The communication between agents is realized by messages. For us, a multi-agent technique is an interesting formalism and though our architecture could be implemented without them it led to a very convenient and scalable framework.

EAAS architecture consists in six types of agents: Human interface agent, actions selection agent, perception agent, action agent, the expert system agent and hardware link agent [22].

The human interface agent (HIA) is the high level of our control architecture. It must generate a succession of goal, or missions for the actions selection agent, according to the general mission of the robot. It also visualizes the progress of the mission while the UAV is executing it. It can follow up the UAV's position, trajectories, obstacle information, etc. It can show a local map, with the obstacles detected by the UAV and the global map with trajectories and the available readings of the different sensors.

The perception agents manage the processing of incoming data (the sensor measurements) and create representations of the environment. They are passive; they only run upon request, perform a one shot execution and then wait for a new message. A perception agent can activate another agent and build a more specific representation using its complementary data. The perception subsystem agents obtain information about the environment and the internal condition of the UAV (for example the level of the battery). They collect data from the sensors and adapt them to provide the information requested by the other agents in the system [23]. There are as many perception agents as there are sensors or sensor groups in the UAV. Finally, this agent is executed upon request by any other agent that has to use the UAV position and whenever new data are available.

The actions selection agents: The deliberative subsystem is composed of the actions selection agents in charge of carrying out high-level complex tasks which require a certain amount of time.

Five deliberative agents (actions selection agent) have been defined: *the mission planning, the path planner, the navigator for itinerary generation, the guidance and pilot and the localization*. The actions selection agent must choose and activate the UAV behavior suited for the mission according to all information available and necessary to this choice [24].

-Mission planning agent's goal is to plan the sequence of tasks based on the information provided by the interface and localization agents to know the position of the UAV on the global map, and to ensure that the mission is achieved. Also, the mission planning agent receives information from the battery charger agent to judge whether a mission is possible considering the level of battery charge.

-The path planner agent may take a goal as input and give a path for achieving the goal as output. The Navigator agent must translate a path into a trajectory for the guidance and pilot. The path does not take into account physical constraints of the robot, but the trajectory that it delivers must integrate them.

-The navigator for itinerary generation agents has two main goals: the calculation of trajectory to the goal that is free of non-moving obstacles, and the estimation of the energy consumption of the planned trajectory. The optimal trajectory calculation is obtained in two steps: first with a graph method to obtain a general sequence of destinations (considering only rooms and hallways), and second with a grid method to find the path between two consecutive destinations (considering all the non-moving obstacles). In both methods a search algorithm is used, Dijkstra's algorithm for the first and the A* search algorithm for the second one. Once the trajectory is determined, an estimation of energy consumption is made based on the cruising speed. Other alternative methods for path planning can also be used, as for example potential fields [25].

The function of the guidance and pilot agent is to convert this trajectory into orders to be performed by the action agent. This agent may be totally autonomous or constitute the process that runs the human interface.

The localization agent: The localization agent's goal is to locate the UAV on the goal map. It receives the sensor measurements from the GPS, odometer, laser data and encoder agents as well as the map representation as input and outputs a representation containing the current position of the UAV. In order to accomplish the agent's goal, the localization agent collaborates with the mission planning. It also sends information to the encoder agent to prevent wide derivations from the current estimated position. In this way, the encoder agent can correct accumulative errors produced by the encoder's readings.

The action agents consist of a set of behaviors controlling the UAV effectors. It is activated by a specific request coming from the behavior selection agent. The selection orders him to work with a perception agent by sending its reference. The action agent sends in turn a request to the proper perception agent. This agent is responsible for driving the UAV to the goal position. Given a desired position and an orientation, and based on the current position and heading, this agent calculates the linear and angular speeds to drive the UAV to the target position (goal). It

receives the global coordinates of the UAV from the encoder agent and the desired points of the trajectory from the guidance and pilot agent from the battery agent. The basic action agents are the goal position and the avoid agents, although more agents can be added [26].
The Expert System Agent: The strength of our architecture is its expert system whose different diagnostics and recoveries envisaged at its level are the following: Fault diagnosis of sensors, effectors and communication links; Control of batteries (autonomy), accuracy of the trajectory and accuracy / quality of the measured data; Control of the embedded control software (partial failure).

The learning agent: The learning element uses feedback from the "critic" on how the agent is doing and determines how the performance element should be modified to do better in the future. The performance element is what we have considered to be the entire agent: it takes in percepts and decides on actions. The last component of the learning agent is the "problem generator". It is responsible for suggesting actions that will lead to new and informative experiences.

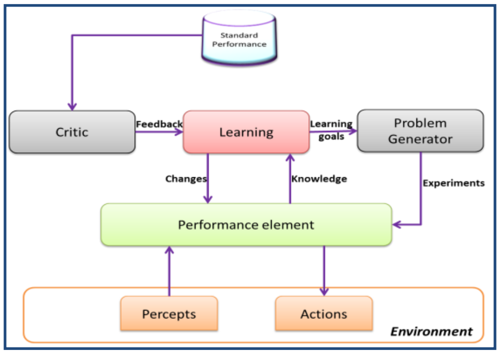

Figure 7:- A general learning agent

The hardware link agent is an interface between the multi-agent architecture and real UAV. Changing the real UAV require the use of a specific agent but no change in the overall architecture. Within a behavior, it is up to the action agent to analyze representations coming from the perception agent and to establish the correct control orders for the platform. Specifically, the hardware link agent communicates with the UAV and obtains its current position, sonar readings and battery charge, and distributes this information to the different sensor agents as required [27].

CONCLUSION:-

This article describes our work concerning the development of effective architecture to control a UAV in hostile environment . We have created a robot control architecture based on a multiagent paradigm that allows various levels of autonomy and interaction between an operator and the UAV.

In this work, the first part presents the three paradigms used to develop UAV control architecture, the reactive, the deliberative and the hybrid paradigm. The significance, advantages and drawbacks of the architectures are described and compared with each other. The hybrid paradigm is the most used since it combines the advantages of planning in deliberative architectures and quick response of reactive architectures in dynamic or unknown environment. In it, we looked at the issue of control architectures for autonomous robot. First, we defined a set of requirements for such architecture, which focus on a different time of cognition (From perception to action to learning), provide a system with several types of intelligence, easy management of the competition, the satisfaction of robustness properties and verifiability, the satisfaction of modularity and composability requirements, and finally giving the ability to autonomous learning expands the variety and diversity of tasks that UAV can perform. Based on these requirements and analyzing the state of the art, we proposed hybrid intelligent control architecture for autonomous navigation of an unmanned aerial vehicle (UAV).

Our architecture consists of a multi-layered multi-resolutional hierarchy of computational modules containing elements of sensory processing, world modeling, Localization, Mission planning & high level decision making, and a Flight control laws.

Our architecture is a real-time intelligent control system for unmanned aerial vehicles operating in the real world. It provides an excellent control in which integrate multiple knowledge representation approaches (ranging from iconic to symbolic and from declarative to procedural,) to build cognitive models and intelligent systems that significantly advance the level of intelligence we can achieve. Sensory processing and planning processes have access to a model of the world that is resident in a knowledge database; this world model enables the intelligent system to analyze the past, plan for the future, perceive sensory information in the context of expectations and thus give, on the one hand, the ability for the UAV to control its own autonomy, and on the other hand the capacity to evolve and to learn.

The strength of our architecture is the expert system module. This module defines the Meta - behavior of the 'UAV'. It manages the data flow, as much as the perturbations. It also manages the asynchronous events coming from the environment. Moreover, it allows adapting an appropriate behavior by aggregating several behavior modules in front of special situations. It also manages the activation of a behavior module, and transfers the control to it.

References:-

1. Boskovic, J. D., Garagic, D., Byrne, J., Cosgrove, M., and Mehra, R. K., "Development of Intelligent Model Predictive Control Algorithms and a Software Design Toolbox for Autonomous Systems," Semi-Annual Report #7 for DARPA Phase II SBIR, Contract No. DAAH01-00-C-R187, July 2004.
2. G.A. Bekey, Autonomous Robots: From Biological Inspiration to implementation and control. The MIT Press, Cambridge, Massachussetts, London, E,gland, 2005.
3. A.Oreback and H.I. Christensen. Evaluation of architectures for mobile robotics. Autonomous robots, 14:33 – 49, 2003.
4. M. Mtshali and A. Engelbrecht, "Robotic Architectures, " Def. Sci. J, vol. 60, no. I, 201 O.
5. R.P. Bonasso and M.Slack, Experiences with an architecture for intelligent, reactive agents. Journal of experimental and theorecal artificial intelligence, 9(2/3): 237-256, 1997.
6. R.R. Murphy. Introduction to AI Robotics. The MIT Press,2000.
7. Laird, Newell, and Rosenbloom, P. (1987) SOAR: An Architecture for General Intelligence, Artificial Intelligence, 33, pp.1-64
8. Hayes-Roth, B. (1995) "An architecture for adaptive intelligent systems." Artificial Intelligence, 72
9. R.Brooks. A robust layered control system for a mobile robot. IEEE journal of robotics and automation, 2(1): 14-23, 1986.
10. Arkin, R.C. and Balch, T. (1997) "AuRA: Principles and Practice in Review", Journal of Experimental and Theoretical Artificial Intelligence, Vol. 9, No. 2, pp. 175189.
11. Albus, J. S., and Barbera, A. J. (2004) "RCS: A Cognitive Architecture for Intelligent Multi-Agent Systems," Proceedings of the 5 Th IFAC/EURON Symposium on Intelligent Autonomous Vehicles, IAV 2004, Lisbon, Portugal, July 5-7
12. Alami, R., Chatila, R., Fleury, S. Ghallab, M. & Ingrand, F. (1998). An architecture for autonomy. The International Journal of Robotics Research, Special Issue on Integrated Architectures for Robot Control and Programming, vol. 17, No 4, pp. 315-337
13. D. A. Handelman and R. F. Stengel. "Rule-based mechanisms of leaming for intelligent adaptive flight control," in Proc. Amer. Cont. Conf., Atlanta, pp. 208-213, June 1988.
14. W. H. Harris and J. S. Levey, eds., New Columbia Desk Encyclopedia. New York: Columbia Univ. Press, 1975.
15. Boskovic, J. D., Bergstrom, S. E., Urnes, Sr., J. M., Mehra, R. K., Hood, M., and Lin, Y., "Performance Evaluation
16. Albus, J.S., "The role of world modeling and value judgment in perception", dans Proc. Fifth Int'l Symposium on Intelligent Control, 1990, p. 154-163.
17. Albus, J.S., "Hierarchical interaction between sensory processing and world modeling in intelligent systems", dans Proc. Fifth Int'l Symposium on Intelligent Control, 1990, p. 53-59. of an Integrated Retrofit Failure Detection, Identification and Reconfiguration (FDIR) System Using High-Fidelity and Piloted Simulations," presented at the *2004 SAE World Aviation Congress*, 2-4 Nov. 2004.
18. Frazzoli, E., Daleh, M. A., and Feron, E., "Real-Time Motion Planning for Agile Autonomous Vehicles," Proceedings of the AIAA Guidance, Navigation and Control Conference, Paper No. AIAA-2000-4056, Aug. 2000.
19. D. Dufourd, "Autonomous construction of indoor maps with a mobile robot" in SPIE 15th Annual Symposium, AEROSENSE'01, UGT III, (Orlando, FL), April 2001.
20. https://en.wikipedia.org/wiki/Intelligent_agent
21. D.Filliat and J.-A. Meyer "Map-Based navigation in mobile robots: I. a review of localization strategies," Cognitive Systems Research 4 (4), pp 243-282, Dec 2003.
22. A.SAYOUTI & H. MEDROMI Autonomous and Intelligent Mobile Systems based on Multi-agent, Book Chapter in the book " Multi-agent Systems – Modeling Control , Programming, Simulations and Applications" .

23. SAYOUTI, H. MEDROMI –“Les Systèmes Multi-Agents : Application au Contrôle sur Internet” Auteurs Éditions universitaires européennes, Août 2012.].
24. Sayouti, F. Moutaouakkil, H. Medromi. "The Interaction- Oriented Approach for Modeling and Implementing Multi-Agent Systems", International Review on Computers and Software (I.RE.CO.S), Vol. 5, N. 2, Mars 2010.
25. Planas R.M., Fuertes J.M., and Martinez A.B. Qualitative approach for mobile robot path planning based on potential field methods. Sixteenth International Workshop on Qualitative reasoning QR 2002. On-line: http://www..upc.edu/web/QR2002/papers/QR2002
26. A.Sayouti, H. Medromi, Book Chapter in the book "Multi- AgentSystems - Modeling, Control, Programming, Simulations and Applications", ISBN 978-953-307-174-9, InTech, April4, 2011.
27. Sayouti, A. (2009). Conception et Réalisation d’une Architecture de Contrôle à Distance Via Internet à Base des Systèmes Multi-Agents. Phd. Thesis, ENSEM, Hassan II University, 2009.