Available online at www.sciencedirect.com

ScienceDirect

Cognitive Systems Research 39 (2016) 58–72

Cognitive Systems RESEARCH

www.elsevier.com/locate/cogsys

Multilayer cognitive architecture for UAV control

Stanislav Emel'yanov, Dmitry Makarov, Aleksandr I. Panov *, Konstantin Yakovlev

Federal Research Center "Computer Science and Control" of RAS, pr. 60-letiya Octyabrya 9, Moscow, Russia

Received 8 April 2015; received in revised form 27 September 2015; accepted 16 December 2015
Available online 21 January 2016

Abstract

Extensive use of unmanned aerial vehicles (UAVs) in recent years has induced the rapid growth of research areas related to UAV production. Among these, the design of control systems capable of automating a wide range of UAV activities is one of the most actively explored and evolving. Currently, researchers and developers are interested in designing control systems that can be referred to as intelligent, e.g. the systems which are suited to solve such tasks as planning, goal prioritization, coalition formation, etc. and thus guarantee high levels of UAV autonomy. One of the principal problems in intelligent control system design is tying together various methods and models traditionally used in robotics and aimed at solving such tasks as dynamics modeling, control signal generation, location and mapping, path planning, etc. with the methods of behavior modeling and planning which are thoroughly studied in cognitive science. Our work is aimed at solving this problem. We propose layered architecture—STRL (strategic, tactical, reactive, layered)—of the control system that automates the behavior generation using a cognitive approach while taking into account complex dynamics and kinematics of the control object (UAV). We use a special type of knowledge representation—sign world model—that is based on the psychological activity theory to describe individual behavior planning and coalition formation processes. We also propose path planning methodology which serves as the mediator between the high-level cognitive activities and the reactive control signals generation. To generate these signals we use a state-dependent Riccati equation and specific method for solving it. We believe that utilization of the proposed architecture will broaden the spectrum of tasks which can be solved by the UAV's coalition automatically, as well as raise the autonomy level of each individual member of that coalition.

Keywords: Cognitive architecture; Intelligent control system; Sign world model; Path planning; Nonlinear control; State-dependent Riccati equation

1. Introduction

One of the obvious recent trends in science and technology is the rapid growth of the R&D areas related to unmanned aerial vehicle (UAV) design. UAVs are getting cheaper and thus more available both to researchers and the general public due to the following factors: First, sensors which are needed in large quantities to build any UAV are getting smaller, cheaper and more energy efficient while the quality of the output signal remains the same or is improving (sensors become less noisy and more robust). Second, other UAV components, such as rotors and carbon bodies are getting more widespread and available at a moderate price. Third, the computational efficiency of modern in-flight controllers has increased significantly. All of these factors gave an impetus to the creation and proliferation of the unified UAV platforms such as Parrot AR.Drone (ardrone2; Bristeau et al., 2011), mikrokopter

* Corresponding authors.
E-mail addresses: emelyanov.stanislav@gmail.com (S. Emel'yanov), makarov@isa.ru (D. Makarov), pan@isa.ru (A.I. Panov), yakovlev@isa.ru (K. Yakovlev).

http://dx.doi.org/10.1016/j.cogsys.2015.12.008
1389-0417/

(mikrokopter), 3DR IRIS (3drobotics), equipped with the sufficient amount of sensors, actuators, peripherals and in-flight controllers, coupled with the core build-in software which automates basic flight maneuvers and modes. This software typically supports easy and seamless integration of the third-party modules via the open data exchange protocols and application programming interfaces (APIs). Thus, a lot of research is now focused on the development of models and methods that can be further implemented as software modules and plugged into existing UAV platforms. The spectrum of the methods under development and investigation is extremely wide: from methods and algorithms for UAV dynamics modeling, identification, and flight controller development to methods of localization, mapping and path planning, to methods of strategic (behavior) planning and UAV coalition formation, etc. An informative recent survey of such methods can be found in Kendoul (2012) for example. Developed methods and algorithms are usually grouped in bundles and implemented as software modules comprising the UAV control system. Thus another direction of research, in which we are more interested, exists in the broad area of UAV design, specifically, studying the methods of interaction between the modules of control systems and the ways of organizing hierarchical relations between them. In other words, we are talking about studying (and developing) the architectures of modern UAV control systems. Control systems which mainly attract researcher's attention nowadays can be considered intelligent control systems (Albus, 2002) (ICS). ICS is a system that is capable of solving non-trivial, intelligent tasks—planning, goal prioritization, coalition formation, etc.—and thus guarantees high levels of UAV autonomy. Under the cognitive approach the ability of the system to solve the abovementioned tasks relies on its ability to model human cognitive behavior and higher psychological functions (and thus only cognitive systems can be characterized as intelligent) (Kurup & Lebiere, 2012). At the same time, researchers of cognitive systems frequently propose such cognitive architectures as can hardly be implemented as software control systems for real-world technical objects due to the lack of interfaces between the proposed methods and modules for solving high-level, intelligent tasks, and the methods for dealing with such lower level tasks as localization, mapping, path planning, control signal generation, etc. Our work aims at filling this gap. On the one hand we are dealing with the non-abstract technical objects involving complicated dynamics and kinematics, such as multirotor UAVs, and creating an architecture for the control system which takes this into account. On the other hand, we are not limiting ourselves to dealing only with low- and mid-level control tasks (UAV stabilizing, performing standalone flight maneuvers, localization and mapping, path planning, etc.), but also trying to automate high-level functions (distribution of roles in the group, coalition formation, goal setting and behavior planning) using cognitive experimental data and psychological methods. As a result, we present the multi-layered cognitive architecture—STRL (from Strategic, Tactical, Reactive, Layered)—of the intelligent control system which automates the control of the coalition of UAVs performing complex tasks in a wide range of scenarios.

2. Related works

Numerous approaches to the creation of the UAV's intelligent control systems exist and the architectures of such systems thus can be classified in many different ways. One of the most advanced ways to do so is to use a hierarchy along with the type of functional specification (implicit or explicit) as a categorization factor. In that case, at one extreme on the spectrum lie ICSs which use simple, flat architectures based on explicit functional decomposition (i.e. the control system is considered to be a bundle of modules without any hierarchy and each module is presumed to solve some functionally specific task). Within this approach the following tasks are typically distinguished: behavior planning, interaction management, contingency management, situation awareness, communication management, navigation (including localization, mapping and path planning) and others. Cognitive functions in that case are dispersed over the whole system, so that each module can implement some of them. One can see (Jameson, Franke, Szczerba, & Stockdale, 2005) as an example of such system (architecture).

On the other extreme there are layered architectures (with possibly infinite number of layers) based on implicit functional decomposition. Each level of the architecture is composed of the elements which abstract specific controllable entities (vehicle subsystems, vehicles, groups of vehicles, etc.) and each element is composed of fixed number of identical modules (groups of modules) having implicit specification. The most obvious example of such an architecture is 4D/RCS developed by the research group of professor Albus (2002). Within 4D/RCS the following 4 implicitly specified modules ("functional processes") compromise each element ("node") of the architecture: behavior generation, world modeling, sensory processing, value judgement. At the higher levels of the 4D/RCS system, behavior generation is meant to be situation planning (i.e. planning in the context of actions, capabilities and high-level goals and constraints) while on the lower levels behavior generation becomes, for example, path planning (planning in the context of spatial constraints) or control signal generation (planning in the space of UAV control inputs). Within such an approach, cognitive functions of the system are concentrated mainly on its highest levels and are specified implicitly.

In between those two extremes lie a vast number of multilayered architectures with explicit module specification. In that case each module is considered to be in charge of solving some specified task(s) and the modules are grouped into layers which encapsulate the level of abstraction: the higher the level, the more abstract representation of input

signals it uses to solve a given task. The tasks being solved at the highest levels of the system are considered more sophisticated and complicated than the tasks of the lower levels. Typically in the areas of robotics and unmanned vehicles 3 levels of control are distinguished and the corresponding 3 level architectures are proposed. Among the most widespread examples of such architectures one can name ATLANTIS (Gat, 1992), 3T (Bonasso et al., 1995), and Aura (Arkin, 1987), among others. Typically within these and other architectures, modules of the high level are considered to automate deliberative behavior functions; modules of the low level automate reactive behavior; and modules of the middle provide interfaces. One of the main differences between the architectures is the mechanism for altering the behavior (either high-level or low-level one). 3T, for example, supports re-planning only on the deliberative level while ATLANTIS uses a mid-level sequencer to re-plan activities both on high and low levels. We believe that the re-planning process should be a pass-through process and propose the architecture which implements this idea. We also would like to pay much more attention to such tasks as collaborative behavior planning and goal distribution, which are usually left out of the focus of well-known and widespread robotics architectures. At the same time we consider that the idea of splitting the control system (at least—for the UAVs) into three levels has been proven successful by the positive experience of researchers in the robotics domain. Thus, we will now propose an original 3-level cognitive architecture.

We also should mention cognitive architectures, which are standard for simulating cognitive functions and in application this simulation to explain the results of cognitive tests and experiments. They are not well suited for the classification by above-mentioned schema, rarely used for control of real hardware and cover only top level of proposed STRL architecture of the UAV control system.

One such system with flat architecture is ACT-R (Anderson et al., 2004; Langley, Laird, & Rogers, 2009). Its last versions are organized into a set of modules communicating with central modules of productions and processing a different type of information. These include a visual modules for sensor processing, a motor module for action, an intentional module for goals, and a declarative module for long-term declarative knowledge. Each module has an associated buffer that holds a relational declarative structure. These buffers comprise ACT-R's short-term memory. The main production subsystem of ACT-R includes realization of matching, selection and execution procedures. In this approach hierarchical action and knowledge representation cannot be constructed for different levels of abstraction. All parts of ACT-R are equal and there is no control links either top-down or bottom-up between modules.

There are some examples of architectures with elements of multilayer approach that particularly focus on memory organization. The representative of this direction of cognitive modeling is the Soar system (Derbinsky & Laird, 2010; Laird, 2012). Procedural long-term knowledge in Soar takes the form of production rules, which are in turn organized in terms of operators associated with problem spaces. Some operators describe simple, primitive actions that modify the agent's internal state or generate primitive external actions, whereas others describe more abstract activities. Recently separate episodic (a history of previous states) and semantic (a history of previously known facts) memories have been added also. In Soar it does not usually take into account features of the control object model. It does not include separate layers for productions and memories of other level of abstraction to execute and process more accurate control as in ACT-R.

ACT-R, Soar and other cognitive architectures have not developed solution for symbol grounding problem (Barsalou, 1999; Harnad, 1990) and therefore do not have flexible instruments for organizing connections with low level control approaches that use physical model of control object. They make particular emphasis on memory organization and several types of workflows between other memory types and do not take into account some useful concepts of control theory such as positive feedback and prediction. It is therefore to improve control quality and degree of autonomy of robotics systems we develop and propose new multilayer architecture in which all these defects have been remedied.

3. General view of architecture

In this work we propose the control system architecture consisting of three levels: strategic, tactical and reactive (STRL architecture). The system controls the behavior of the UAV, which is an individual member of the coalition taking part in joint activity aimed at the solution of the coalition task. The strategic level of STRL is the core cognitive level of the system and is in charge of solving high-level cognitive tasks (coalition formation and behavior planning) and using sign knowledge representation (Osipov, Panov, & Chudova, 2014) and extensive inter-coalition communication to accomplish them. Tactical and reactive levels contain modules that support these activities and translate them into the UAV low-level control signals which serve as the actuators' input. Overall architecture is shown on Fig. 1 (modules are depicted as boxes, intermodule interfaces as arrows, different world model representations as ovals).

The strategic level's main task is to build a plan of the behavior for a member of the coalition coordinated with the plans of all the others. Each participant has its own world model constructed in accordance with the psychological theory of group activity. The world model consists of the interlinked knowledge elements (signs) referred to as domain objects or actions. Each element contains both procedural and declarative knowledge about the object (or the action). Thus, we suggest not to operate with several types of memories, such as procedural, episodic and others, but to have several components of the element of

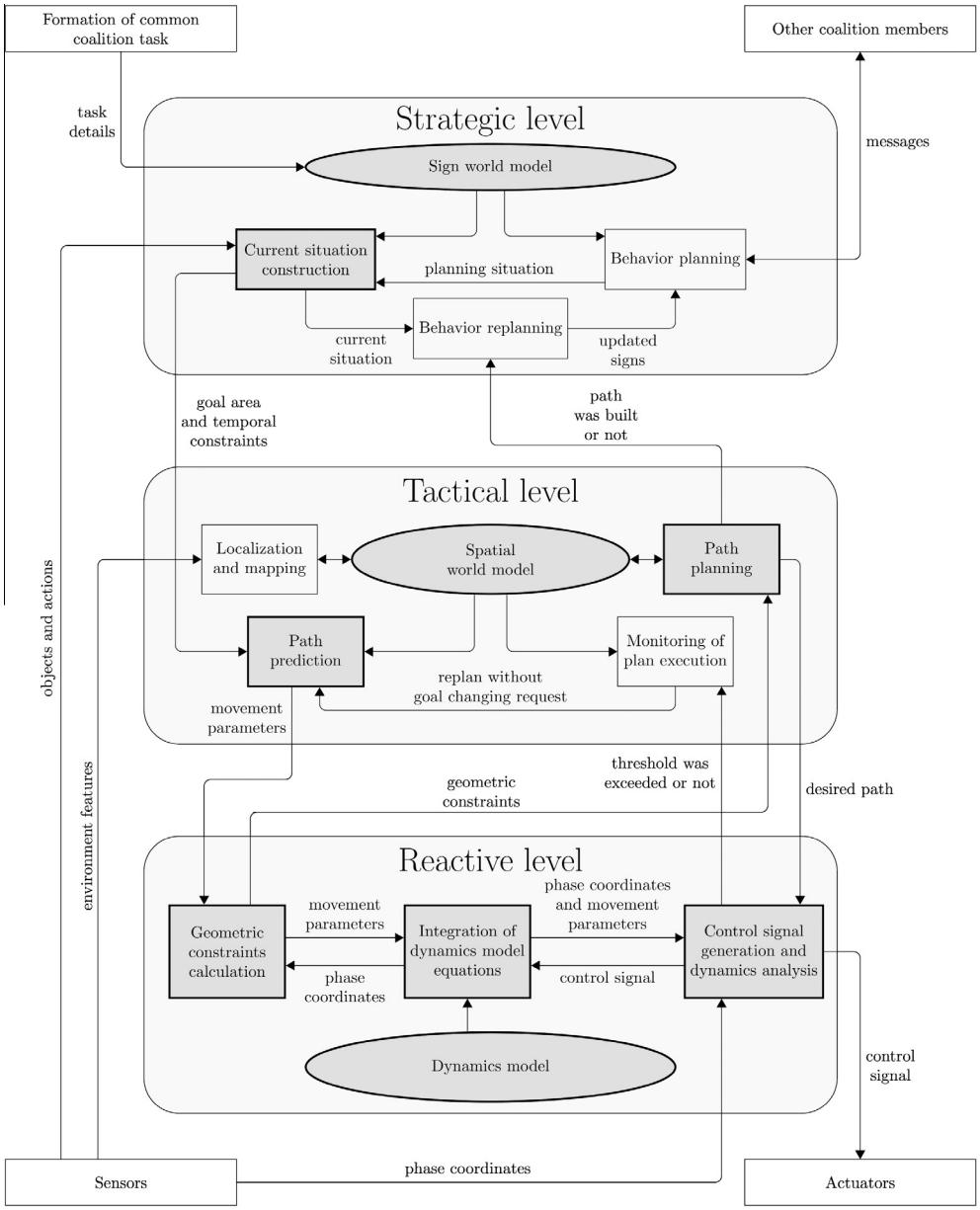


Fig. 1. Schema of proposed cognitive architecture. Bold bordered nodes represent parts of the architecture realized in the program system.

the world model, such as name, image, personal meaning and significance (Leontiev, 2009). For example, for a sign with the word "table" the image component presents characteristic features and constituent parts of the mediated object (real table): "square tabletop", "four table legs", "brown colour" and that it "is made of wood". The significance of "table" sign is the common abstract actions associated with the mediated object: "take food on the table", "playing cards at the table", etc. The personal meaning component reflects personal actions associated with the mediated object for the actor: "I can dance on the table", "I can eat on the table", etc.

One of the sign components—significance—is the same for all members of the coalition. Therefore, if two or more members share the name of some knowledge element (i.e. this element is present in their world models) the significance component will be the same while image and personal meaning can be different. Thus only significance component is used in the communication between coalition members (via the designated protocol). Significance based communication is used to share individual knowledge and abilities as well as to avoid conflicts while generating the behavior plan (i.e. it can make an individual plan consistent with the plans of the others). At each stage of plan execution, the description of the current state of the sign world model (situation) is updated using the data coming from the sensors. The temporal-spatial aspect of the situation description is extracted to form the task which is passed to the tactical control level. This task contains a spatial description of the goal area (location) and its achievement. In case the task cannot be accomplished, as informed by the tactical level planner, the strategic

re-planning procedure is invoked and the corresponding adjustments are made to the individual behavior plan upon coordination with the other members of the coalition.

The tactical level of the STRL architecture is the mediator between high-level activities (as described above) and low-level activities (control signals production), and the tasks over which it is in charge are the navigational. We split the navigation activities into 3 major categories: mapping (construction, updating, refinement of the spatial model of the world), localization (binding the UAV state to the existing map) and path planning. The latter is divided into three phases: prediction, generation and monitoring. The planning process is used when the information on the goal location (some area bound to the map) and time limits on its achievement is received from the strategic control level. Then the prediction module performs preliminary calculations of the necessary movement parameters (e.g. speed) which are needed to achieve goal area in time. These parameters are then transferred to the lower (reactive) control level of STRL, which uses them to create a spatial (geometric) model of the UAV movement dynamics constraints and then sends these constraints back to the path planning module. The latter tries to generate the path under given constraints. Depending on the result (either successful, which means that the path is generated, or unsuccessful, which means that the path to the desired goal location cannot be achieved in the allotted time) the corresponding message is sent to the higher (strategic) level. In response to a failure message the strategic level is supposed to generate an alternative navigation task (location-time pair). Thus, path planning is an iterative process supported with the feedback both from upper and lower levels of control. We suppose that the idea of finding spatial representation for the UAV dynamics constraints and the use of an iterative loop "prediction - calculation geometric constraints - planning" will substantially improve the computational performance of the path planning process and thus improve the overall performance of the control system (as more navigation tasks will become solvable in the allotted time). For more details, see Section 5.

The main task at the reactive level of control is to generate the control signals needed to follow the trajectory, provided by the tactical layer as a path, under specified movement constraints (e.g. speed), also provided by the tactical layer. These signals are then fed to the actuators and the control error is analyzed via sensory feedback. For example, measured phase coordinates (current position and speed) of the control object (UAV) are compared with the desired ones. If the difference exceeds the predefined threshold, the tactical layer is informed and, potentially, a re-planning process as described above is triggered. The control signals generation module can also work in modeling mode, which is needed while path geometry constrains calculation. To calculate these constraints the input data is received from the path prediction module residing at the tactical layer and then integration of dynamic model equations is triggered, coupled with the control signal generation. As a result, phase coordinates are obtained which are analyzed in a special way to form the needed geometric constraints of the path, which are then pushed back to the tactical layer.

The three-tiered approach is also reflected in the psychological Stanovitch's tripartite framework (Stanovich, 2009), where it is suggested a tri-partition of architectural levels in human cognition corresponding to what is called the "Algorithmic Mind", responsible for cognitive control, the "Reflective Mind", responsible for more higher order deliberative processes and the "Autonomous Mind". Mentioned levels correspond to tactical, strategic and reactive levels of STRL architecture indicating the psychological reliability of the STRL.

Proposed multi-layered cognitive architecture of the intelligent control system has quite a wide range of functional capabilities. It uses methods of modeling human cognitive activities at the highest level of control to solve the tasks of behavior planning and re-planning, group formation and others. It utilizes novel path planning methodology which acts as the mediator between high-level and low-level control tasks. This methodology includes geometry constraints calculation, which is done at the lower (reactive control) level of the system. The architecture is not strictly tailored to the UAV domain. With insignificant changes it can be used to control other types of unmanned vehicles (or, in general, complex technical objects).

4. Details of the organization on strategic level

4.1. Knowledge representation

As the basis for psychological theories, not only a qualitative description of the properties of cognitive functions, but also the structural description of the underlying mental formations, cultural-historical approach of Vygotsky-Luria (Vygotsky, 1986), the theory of activity (Leontiev, 2009) and the model of mind (Artemieva, 1999) were used. According to these theories, the higher cognitive functions are carried out within the framework of the so-called motivated objective activity when objects and processes are mediated by the external environment for the subject of special education called signs. The process of engaging the sign in a particular cognitive function has three generators: an image, a significant and a personal meaning (see Fig. 2) (Osipov et al., 2014). The image component is responsible for playback and discernment of the mediated object or process during the activity. The significant component defines the place of the sign in some psychological sign system. This place reflects in the functional sense the ways of using a mediated object or a process and is determined by general historical practice of the collectivity that is owner of the sign system. Finally, the personal meaning component carries its own experience of action between the subject and the denotation of the sign, which is expressed by the integrated estimate of the role of denotation in its

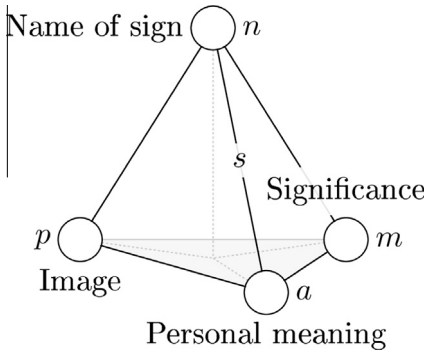


Fig. 2. Structural components of the element of the world model.

current activities: if the process or object satisfies the current motive.

The three-component structure of individual knowledge, which, as mentioned above, is called the sign in psychology and confirmed by the work of neuroscientists, is an attempt to construct a general theory of operation of the human brain. Thus, in the re-entry theory of Edelman (1987) and Ivanitsky (1996) it is approved that the formation of conscious sensation or fixing input flow occurs only when the excitation has been activated by sensory input via the associative cortex from the hippocampus, and then imposed on the sensory track on the projection cortex from hypothalamus. This "circle of sensations" that passes over a characteristic time in the 150–300 ms sequentially activates the three components of individual knowledge: the image (primary and sensory cortex), the significant component (hippocampus) and the personal meaning (the hypothalamus). In addition, the structure of the cerebral cortex, based on modern neurophysiological concepts, is almost uniform in its entirety (the existence of neocortical columns). Hereby the plurality of links between small enough areas of the cortex (the so-called connectome (Zador, Dubnau, & Oyibo, 2012)) clearly indicates its hierarchical structure and the presence of both uplink and back downlink. It follows that the components of individual knowledge elements should have a hierarchical homogeneous structure with ascending information flows and descending feedback. Furthermore, the significant component should have such recognition function, except for the categorization of static objects and dynamic processes, which uses a feedback signal for predicting a sign the next time.

Presented knowledge representation on the strategic level has common points with existing works in this area. So, an approach that can be represented in terms of "heterogeneous proxytypes" (Lieto, 2014) should be noted. Knowledge base contains heterogeneous pieces of information semantically pointing to the same object and conveying different types of information. Such pieces of information may be contextually activated by an agent according to the external stimulus being categorized. In these categories we can also describe partitive information communication between agents but cannot get grounding in other levels of knowledge representation.

To define the mathematical model of components of individual knowledge elements infinite state machine with variable structure and finite memory (recognizing state machine or R-SM):

$$R_i^j = \langle X_i^j \times \hat{X}_i^{j+1}, 2^{Z_i^j}, X_i^{*j} \times \hat{X}_i^j, \varphi_i^j, \vec{\eta}_i^j \rangle,$$

where i—the index of R-SM through the hierarchy, j—the level of the hierarchy, X_i^j—the set of input signals, X_i^{*j}—the set of output signals, X_i^{j+1}—the set of control signals from the upper layer of the hierarchy, $\hat{X}_i^j$—the set of control signals to the lower layer of the hierarchy, $2^{Z_i^j}$—the set of states (Boolean of prediction matrices (see below)), $\varphi_i^j : X_i^j \times \hat{X}_i^{j+1} \to 2^{Z_i^j}$—the transition function, $\vec{\eta}_i^j : 2^{Z_i^j} \to X_i^{*j} \times \hat{X}_i^j$—the vector-function of outputs. Input, output and control signals are vectors of real numbers. Each component of these vectors is a weight of recognizing or input feature.

As a recognition function $\hat{f}_k$ of the k the output feature in R-SM it is convenient to use the set of bit prediction matrices $Z_k = \{Z_1^k, Z_2^k, \ldots, Z_m^k\}$. In these matrices each column $\bar{z}_u^r$ is the prediction vector of input features in the moment $\tau_s + u$, where τ_s is a start of the calculation circle (the moment of operation of the control signal $\hat{x}_i^{j+1}$). The matrix Z_r^k specifies the sequence of bit vectors where each bit indicates presence of a feature recognized by the function $\hat{f}_k$. The algorithm $\mathfrak{A}_{th}$ that calculates the transition function φ_i^j and the vector-function of outputs $\vec{\eta}_i^j$ by the initial moment τ_s, the control effect $\hat{x}_i^{j+1}(\tau_s)$ and the input effect ω_i^j is shown on Fig. 3.

Introduction of that R-SM and several relationships on the set of R-SM allows defining all components of the sign. The image of the sign s corresponding to the feature f_1 is a subset $p(s)$ of features where $\forall f_i \in p(s) f_i \sqsubset f_1$. Here the relationship $\sqsubset$ is the relationship of subsumption of one feature by another. If the set of columns of a prediction matrix is divided into two subsets, columns of conditions and columns of effects, then each feature that has such prediction matrices is named as a procedural feature. If f_1 is a feature corresponding to the sign s, f_2 is a procedural feature and $f_1 \sqsubset^c f_2$ then f_2 is named as an element of significance of f_1. If F_I is the subset of features where each feature describes characteristics of the control system then the definition of personal meaning will be as follows: f_1—the feature corresponding to the sign $s; \sqsubset$—the procedural feature, $f_1 \sqsubset^c f_2, \exists f_I \in F_I : f_I \sqsubset^c f_2, f_2$ is named as an element of personal meaning of f_1.

The proposed approach to the description of the image component of the sign using state machine representation has been applied to the wire toy recognition task. The input dataset for this task contains sequences of shifted wire toy pictures comprising of variations of eight types of figures (see Fig. 4). Hierarchy of R-SM contains three-level

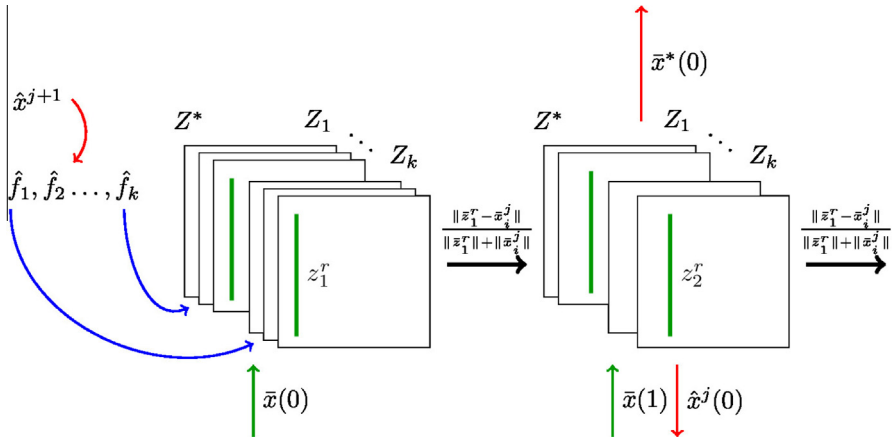


Fig. 3. Schema of the R-SM's function. From the set of measure functions $\hat{F}_i^j$ subset is selected by prediction vector $\hat{x}^{j+1}$ entering from the upper level. This subset of active functions forms the set of active prediction matrixes Z^*. This subset is filtered on each time step according to the input vector $\bar{x}^j$ and the simple distance metric. On each time step the output vector $\bar{x}^*$ and prediction vector $\hat{x}^j$ into the lower level are constructed.

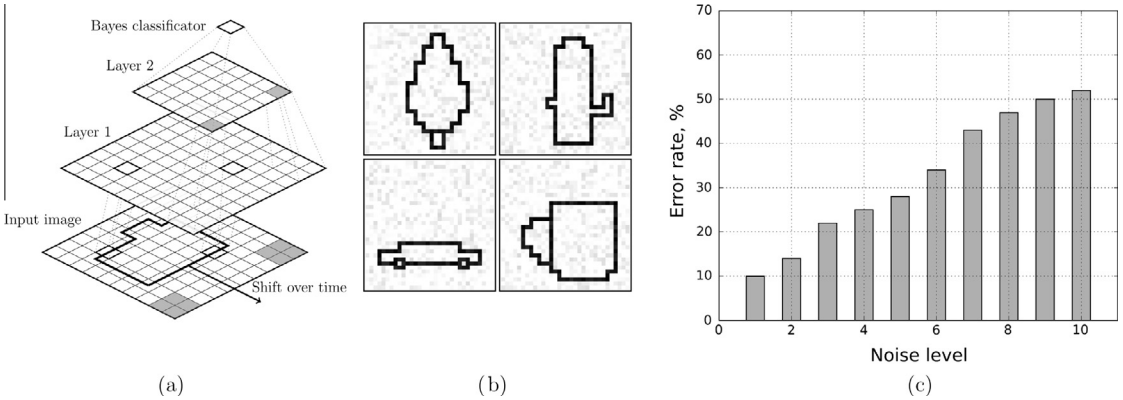
counting simple Bayes classifier on the highest level. On the lowest level of the hierarchy the set X_i^j of the first R-SM was presented by a sequence of vectors, each component of which represented the gray scale value of the corresponding pixel of the input image. The set X_i^{*j} of the R-SM was the set of vectors of which each component was the likelihood of a presence of memorized patterns on the parts of the image. The set $\hat{X}_i^{j+1}$ contained predicted patterns at the next time from the second layer. On the next level, R-SM encoded a sequence of patterns for the corresponding part of an image consisting of several pixels. The top-level Bayes classifier took the decision about the name of the input sign. Fig. 4 depicts the schema of the experiment, test images and accuracy of recognition, dependent on the level of noise on the image.

4.2. Self-organized processes

Within the process of the actor activity specific relationships occur on the sets of sign components. In its turn it leads to the formation of the world model of the UAV. Three types of semantic networks were used as a model of the UAV's knowledge about the world. These are the network based on the set of sign images, the network based on the set of sign significances and the network based on the set of personal meanings. Self-organized processes on these networks involve the supplementation of the relationships' collection as well as the formation of new nodes of the network, which corresponds to the formation of a new element of individual knowledge.

The process of the formation of a new sign includes establishing connections between the sign components and the naming generation structure. Until the name is obtained, this structure is called a protosign and its components are called percepts (evolved into the sign image), a functional significance (evolved into the sign significance), and a biological meaning (evolved into the personal meaning after the sign formation completion). Common schema of a new sign formation (Osipov et al., 2014).

1. The formation of a percept.
2. Generation of the set of pairs "percept—functional significance" of the functional significance of the object based on the previous experience or precedents.



(a) (b) (c)

Fig. 4. Experimental results of R-SM recognition tests. (a) a schema of R-SM hierarchy used in the task, (b) an example of test images with noise, (c) accuracy of recognition, dependent on the image noise level.

3. From the cultural environment based on the external collective information objects (for example, texts) the actor obtains the pair "sign name—significance", accumulated in the natural language system, and evaluates the degree of closeness of the functional significance obtained in the phase 2 to the significance obtained from the cultural environment. If these significances are not close enough, then the percept formation continues by returning to phase 1.
4. Linking the name from the pair "sign name—signif icance" to the percept constructed after the completion of phases 1–3. At this time, the percept turns into an image.
5. Formation of personal meanings of the sign based on precedents of actions with the object.
6. Linking the name from the pair "sign name—signif icance" to each personal meaning. From this time on, the functional significance turns into the significance, and the biological significance turns into the personal meaning.
7. Continuing to map the biological significance—percept by including the personal meaning (formed in the preceding phase) in the domain and by including the image formed in phase 4 in the set of values.

Consideration of procedural features in the form of rules with defined sets of added and removal features allows algorithm construction of the main iterative process (phases 1–3) of the described sign formation schema with R-SM. To define conflictness and applicability properties of rules on the set of procedural features special operations are introduced. These are the cast of the column $\bar{z}$ to the R-SM $R(\bar{z} \rightarrow R)$ and the cast of the column $\bar{z}$ to the R-SM R by the column $\bar{z}$ $(\bar{z} \xrightarrow{\bar{z}'} R)$.

5. Details of organization on tactical level

The tactical level lies in between a strategic level and a reactive level and serves as a bridge which connects high-level, intelligent activities (cognitive world modeling and behavior generation) with the low-level ones aimed at forming control signals needed to operate the UAV and perform flight maneuvers. Our main idea is that both of these classes of activities can be meditatively connected via the spatial world model. The strategic level can output the time-spatial reference (associated with some high-level task the UAV has to perform) as an input to the tactical level modules. At the same time the reactive level is able to transform the model of UAV dynamics constraints into spatial models of constraints (either geometry or metric or, as we propose, both) and provide them to spatial planners residing on the tactical layer. In that case the core activity of the tactical level (which is the main mediator within the architecture) is to maintain the process of spatial reasoning. The latter is split into three main sub-processes: localization—registering the state of the UAV in the respect to the spatial world model (SWM); mapping—maintaining the consistency of the spatial model of the world; and path planning—finding a sequence of states of the SWM ending with the goal state provided from the strategic level.

In modern robotics and AI the first two abovementioned processes are usually tied together in what is called simultaneous localization and mapping (SLAM). There exist dozens of methods and algorithms nowadays to solve the SLAM task, each of which differs from the others mainly in the input data it can process (but not in the way it processes these data). The input data include at least the following components: a mathematical model of UAV dynamics and the sensory data. So, practically speaking, different SLAM methods are needed for different robotic systems (as different robotic systems are equipped with different types of sensors and have different dynamic models). If we narrow ourselves to the vertical take-off and landing of multirotor UAVs (multicopters), the following SLAM algorithms can be named. First of all, if the UAV is equipped with laser range-finder SLAM algorithms, initially developed for ground robots, these can be used successfully (with some modifications); see (Borrmann, Elseberg, Lingemann, Nuchter, & Hertzberg, 2008) for example. When the stereo-camera is used, the SLAM algorithms based on the computational geometry and computer vision are used (Sim, Elinas, Griffin, & Little, 2005). If the UAV is equipped only with a single (forward-looking camera), then such methods as Saxena, Sun, and Ng (2007), Bills, Chen, and Saxena (2011) can be used. Needless to say, in many cases there exists a prior map which is also an input to the SLAM algorithm, so that mapping is often not meant to be the process of building the spatial world model from scratch (although it can be so) but of updating and refining it. Also, in some cases a localization task becomes trivial; for example, when the UAV is performing outdoor flight and is equipped with global positioning sensors (GPS receivers).

Path planning methods may be presumed to be more universal than the SLAM ones as they do not deal directly with sensor data. In effect, one path planning algorithm can be used for many different types of robotic systems. Typically the input to the path planning module (method) is the spatial world model (along with the references to the desired goal state(s) and current start state), and that is the case within our architecture. Usually, the spatial world model is a graph where each node corresponds to some 3D (or 2D) point (location) and each edge corresponds to so-called elementary trajectory (typically the segment of a straight line). This is a trajectory that can be followed in a reactive way by the use of lower level methods and algorithms—the modules of the reactive level of the control system—so that path planning is viewed as a graph search (Likhachev, Ferguson, Gordon, Stentz, & Thrun, 2008).

In our work, we assume that 3D (or 2D) grid (Yap, 2002) is used as the spatial world model as it is a simple and powerful graph model successfully used both in ground (Elfes, 1989) and aerial robotics (De Filippis,

Guglieri, & Quagliotti, 2012). Grids are so widespread because they can be extracted from the output of the SLAM module(s) by naive, fast and efficient algorithms (in most cases only post-processing is done). To find a path on a grid, one can use one of the numerous algorithms of heuristic search: A* (Hart, Nilsson, & Raphael, 1968), R* (Likhachev & Stentz, 2008), JPS (Harabor & Grastien, 2011), Theta* (Nash, Daniel, Koenig, & Felner, 2007), etc. The main problem here is that the result—path found as the sequence of free grid cells and sections—is not guaranteed to be executable by the lower level of the control stack, i.e. corresponding control signals to follow the path cannot be generated. The most obvious example of such a path is a path containing a sharp turn when the corresponding maneuver cannot be performed due to the vehicle's dynamics constraints. One way to take these constraints into account is to extend the graph model used to search for a path. In that case we are talking about finding a path as a sequence of states which are referred not to the elements of spatial world model but to the elements of some extended graph model which incorporates also information on UAV dynamics. As an example of the algorithm exploiting such an idea one can name (Kuwata et al., 2009; Kothari & Postlethwaite, 2013).

As mentioned above, one of the core intuitions which underlies the proposed architecture of the control system is that the reactive level of control can be separated from higher levels and, for this, we are proposing the mechanism of such separation is to find a model of geometry constraints which can easily be incorporated into the spatial world model, on the one hand, and take into account UAV dynamics, on the other. We propose the following model of geometry constraints: we assume that the executable trajectory corresponds to the grid path which can be presented as the sequence of such sections that the angle in between each pair of them does not exceed predefined threshold α_m. On Fig. 5 two paths are depicted: the one on the left violates maximum level of alteration constraints and the one on the right does not.

The exact value of the threshold—α_m—is estimated by the module which resides at the reactive level and takes into account all constraints imposed by the vehicle dynamics, control laws, etc. For more detail, the method of the constraints transformation is described in Section 6 of this paper.

To the best of the authors' knowledge, there are not many grid path planning methods capable of taking angle-change constraints into account, although methods based on similar ideas of constraints modeling do exist (see Kim, Kim, Shin, Kim, & Myung, 2014 for example). Thus, we have developed and studied both theoretically and empirically a new path planning method for the proposed model of geometry constraints—LIAN (from "limited angle"). It is pretty similar to a classic A* search as it uses the same heuristics and strategies to focus the search. The main difference is that at each step A* is investigating only grid cells that are immediate successors of the current one (i.e. they are adjacent to the current one) while LIAN "jumps" to the cells that lie at some distance from the current cell (exact distance value—Δ—is an input parameter of the algorithm). By doing this we implicitly present each partial path (in the set of all paths under investigation) as the sequence of sections. Thus, paths which do not satisfy "in-between-section angle" constraint are easily pruned. It is proved that the proposed algorithms are sound and complete (with respect to the input parameter Δ). More details on the proposed algorithm can be found in Yakovlev, Makarov, and Baskin (2015) and Yakovlev, Baskin, and Hramoin (2015).

An example of the path generated by the proposed algorithm can be seen in Fig. 6. The start cell is located on the left edge of the grid and the goal is located on the right. The path is shown as a curved line. Cells that have been visited

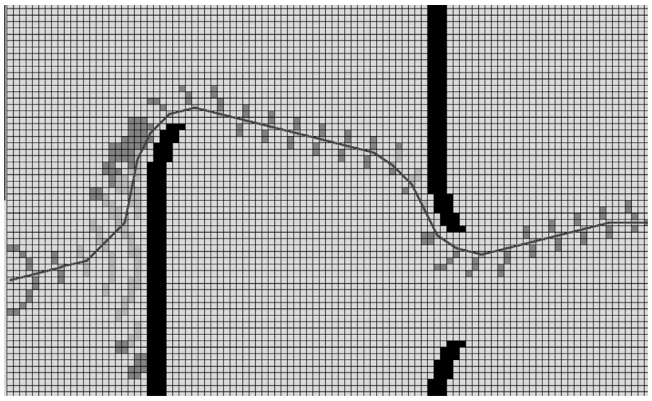


Fig. 6. The path found by the LIAN algorithm as well as the explored areas (shaded in light gray).

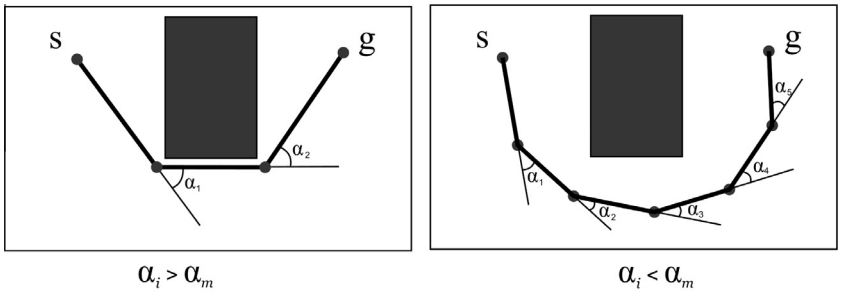


Fig. 5. Angle constrained path (on the right) and a path violating maximum angle of alteration constraint (on the left).

by the algorithm (so called OPEN and CLOSE lists in heuristic search literature) are shaded in light gray. Obstacles are shaded in dark gray. One can clearly see the "jumps" as the caverns of unvisited cells in between the visited ones (center and right of the picture). The most explored area is located on top of the left-most obstacle: LIAN has investigated a handful of variants of passing that obstacle, in such a way that the maximum angle of alteration constraint is not violated, before finally finding a suitable path fragment.

We have conducted an empirical study of LIAN and compared its ability to solve angle-constrained path planning tasks and its performance to the competitors: nave modification of Theta* which takes into account angle constraints—Theta*-LA; and its modification implementing grid-weighting techniques—wTheta*-LA (for the original description of algorithms see Kim et al. (2014), for the details of authors implementation of algorithms and experimental setup see Yakovlev, Baskin, & Hramoin (2015)). We have used 80 different fragments of city maps (retrieved from OpenStreetMaps (wiki.openstreetmap) database) sized 1347 m × 1347 m each and discretized to 501 × 501 grids, and we have chosen 5 different start-goal locations on each grid. Various angle constraints were targeted. For example $\alpha_m = 25$ corresponds to AscTech UAV flying at the air speed of 7 m/s, the typical cruise speed in urban environments. Results are shown in the Table 1.

Here *sr* (success rate) is the number of accomplished angle-constrained path planning tasks divided by the number of all tasks (time limit for each task was set to 60 s); PAR-10 (penalized average runtime) is the metric that averages the runtime and takes failures into account (Hutter, Hoos, Leyton-Brown, & Stutzle, 2009); *t* (time) is the time (in seconds) needed for an algorithm to produce solution averaged without taking failures into account; *m* (memory) is the number of intermediate grid elements (in thousands) explored by the algorithm and stored in memory (the memory consumption of the algorithm); *pl* is the length (in meters) of the resulting angle-constrained path.

Results of the experiments prove that the proposed algorithm significantly outperforms existing analogs (at least for the considered outdoor navigation scenarios): LIAN solves more angle-constrained path planning tasks than the competitors while using less memory and processing time. In Yakovlev, Baskin, and Hramoin (2015) the reader can find more details on the algorithms experimental assessment.

In order to form a geometrical model of the executable trajectory constraints, modules of the reactive level need to be provided with the input that defines the values of basic UAV movement parameters (for example—speed). To calculate these parameters, a path prediction module residing on the tactical level is used. Basically path prediction can be considered as fast, constraint-free version of path planning. In the simplest case, when the goal location and time of the arrival is received from the higher level, the length of the straight line connecting the current UAV location and the goal is calculated and used to determine the desired (minimum) speed of traversal, which is then passed to the "geometry model generation" module of the reactive level. Furthermore, with such a straightforward computation more complex algorithms can be used (for example, path planning algorithms utilizing over-informative heuristics or taking no spatial constraints (obstacles) into account).

The last module of the tactical level not described so far is a path monitoring module. It receives the current location of the UAV and matches it up with the planned path. If the discrepancy exceeds a predefined threshold it triggers again the "path prediction—geometry constraints generation—path planning" loop to find a new path to the goal state. If such a path cannot be found, the corresponding module at the strategic level of the control system is informed, and the global goal-change process is triggered).

6. Details of organization on reactive level

As already noted, the main challenge for the reactive level is to provide specified dynamic object characteristics (speed, location, etc.), received from the tactical level, by the means of a control signal generated by the controller. The control signal is applied to the UAV actuators: screws engines, ailerons, etc. Additionally, using the appropriate procedure, the problem of calculation of geometric constraints on the admissible trajectory of movement is solved at this level. Let us describe the methods of controller synthesis and constraints calculation that are proposed in our architecture.

6.1. Method of controller synthesis

In the area of automatic control there are many approaches and techniques to the construction of the controller. Some of them are indirectly related to a cognitive human activity at the level of implementation and control of motor actions. For example, the control based on artificial neural networks (ANN) uses the principle of learning by examples, and control based on fuzzy logic uses the formalized intuitive knowledge of domain experts. The

Table 1
Comparative study of LIAN, Theta*-LA and wTheta*-LA path planning algorithms.

	$\alpha_m = 25$					$\alpha_m = 30$				
	sr (%)	PAR-10	*t*	*m*	*pl*	*sr* (%)	PAR-10	*t*	*m*	*pl*
LIAN-5	98	12	0.5	6.3	1617	98	11	0,5	6.1	1611
Theta*-LA	12	536	2.1	37.5	1574	31	421	2.2	47.4	1580
wTheta*-LA	55	277	2.76	58.3	1598	73	165	2.7	61.0	1567

application of these techniques is particularly useful if there is no possibility to build an adequate mathematical model of UAV. A few works on the use of ANN and fuzzy logic to automate the UAV's flights are outlined in Garcia and Valavanis (2009) and Buskey, Wyeth, and Roberts (2001). The main disadvantage of the noted approaches is the lack of a rigorous proof of operability of such controllers. On the other hand, there are many rigorous techniques free from this disadvantage that do not rely on human motor activity, such as the synthesis of linear-quadratic controller, H-infinity methods, feedback linearization and backstepping methodology. Limited area of application is a disadvantage of such techniques: they deal with either a linear model of the UAV or a nonlinear model of a special kind.

In the proposed architecture one of the promising approaches used is based on the Riccati equation with state-dependent coefficients (SDRE—State Depended Riccati Equation). Work in this area has been carrying out actively since the mid-90s of the last century (Cimen, 2008; Mracek & Cloutier, 1998). On the one hand, development and application of this technique provide a fairly general methodology for constructing suboptimal, smooth, nonlinear, state dependent controllers for nonlinear systems. On the other hand, the SDRE control allows us to use some of the principles that are utilized in the realization of lower cognitive human functions, associated with the control and execution of motor actions. Namely, it is possible to adapt the control signal depending on the mode and conditions of operation, and constraints on the control. Let us describe the approach.

The SDRE technique is based on the representation of the original nonlinear system in the quasi-linear form. This allows us to apply the procedure for stabilizing control constriction similar to a procedure for optimal synthesis for linear systems (linear-quadratic regulator synthesis) by means of considering the corresponding algebraic Riccati equation, whose coefficients are already dependent on the state variables of the original system.

The Riccati equation, as in the linear case, is given by the linear quadratic cost functional, reflecting the quality requirements of the transition process by entering two weight matrices, for state and control. The objective of control is to minimize the functional. However, the elements of these two matrices are also non-linear functions of the state. This fact allows the specification of different requirements for the transition process, depending on the operating mode of the system (areas of phase space), as well as taking into account the existing control constraints. For example, the requirements for the trajectory accuracy is to be increased at the final stage of a missile flight or at aircraft landing, so that values of the respective elements in the weighting matrix of the system state increases to modify the control law. Besides, one can modify a control weight matrix to create an area in a state space, where the gain should be lowered because of the probability of the control saturation.

Thus, the controllers synthesized in this way can compensate nonlinearity of control systems by means of the SDRE numerical solution, taking into account (Cloutier & Cockburn, 2001; Cloutier & Stansbery, 2002) constraints both on control and the system state, and can continuously change their algorithm depending on the scenario of the task.

In contrast to the well-known linear-quadratic regulator synthesis, the SDRE technique requires real-time performing of a number of quite computationally difficult operations to solve the matrix Riccati equation for each of the current states of the system. This may require significant computational resources. Technically it can be realized only with a certain time step, varying depending on the computational complexity of a particular problem-solving task.

To overcome this difficulty, the authors proposed an original method for the approximate analytical solution of SDRE, which significantly reduces the required computations. The asymptotic stability of the resulting control system can be rigorously proved by means of Lyapunov function.

The performance of proposed method was confirmed by the "classical" task of stabilizing a nonlinear second-order system: the inverted pendulum from paper (Dutka, Ordys, & Grimble, 2005). Constructed nonlinear control is up to 70% more effective on the considered quadratic cost function than linear control, which is optimal at the equilibrium point of the system. However, the experiments showed that the nonlinear control effectiveness significantly depends on the initial conditions. It decreases when the initial conditions tend toward the equilibrium point of the system. In a small neighborhood of equilibrium point linear control, even slightly better (up to 10%) results are produced than in a constructed nonlinear one. In general, the estimation of proposed analytical solution accuracy is the subject of the further investigations.

Thus, in our architecture we use the algorithm for constructing a nonlinear controller, which greatly reduces the amount of computation compared with traditional procedures of the SDRE control approach. Another distinctive feature of the approach is sufficient mathematical and rigorous proof of closed system stability, as well as the opportunity to use some principles which affect human motor actions: it is possible to adapt the control signal depending on the mode and conditions of operation and constraints on the control. More details on the proposed method can be found in Dmitriev and Makarov (2014).

6.2. Method of constraints calculation

In the case of horizontal flight, the tactical level algorithms are trying to construct a trajectory in the form of a straight line sequence in which the angle between any adjacent lines of the sequence does not exceed (in absolute value) a fixed value. It is assumed that satisfaction of this condition ensures the feasibility of the resulting trajectory,

i.e. the possibility of constriction of an admissible control signal to follow the trajectory with the specified error. The proposed method of geometric constraints calculation is based on numerical analysis of an attainability domain of a dynamic system. The exact solution of this problem usually requires a large computational cost. Let us confine ourselves to a simplified approach based on certain plausible assumptions. Nevertheless, the proposed approach is rather general for a lot of applications.

Let us suppose that a UAV must follow the trajectory tr. Our basic assumption consists in the idea that the flight conditions and precise control within its constraints make it possible to construct admissible control, which guarantees that the UAV is located in some admissible neighborhood of the desired straight-line trajectory. This neighborhood is defined by the "tube" with a radius r_d (see Fig. 7). The specific value of rd depends on the type of UAV, conditions and mode of flight, etc. It is assumed that values of rd are known (e.g. based on the operating experience of the selected UAV).

Obviously, in the case of trajectory breaking, shown in Fig. 7, the location of the UAV at the point P did not belong to the "tube", which is the worst challenge for control. Let us consider an additional area to be defined by a circle with a given radius R_d and to be centered at the point P. It is assumed that any point in does not contain obstacles. In fact, the radius R_d determines the turning maneuver area.

It is assumed also that in the worst case, which is considered below, the velocity vector V_g of the UAV is deflected from desired flight trajectory before breaking in the point P at a maximum angle α_v (see Fig. 7). Then the problem of geometric constraints calculation may be formulated as follows. It is necessary to find the maximum angle α, so that the UAV trajectory, without leaving the circle, will return again in the admissible "tube" and will no longer leave it (because of the basic assumption above, there is always an admissible control which guarantees it).

Next, it is assumed that R_d value is given (for example, by means of location map analysis) and the value of the velocity V_g is supplied as a parameter from the tactical level. Besides, it is assumed that the maximum of the UAV allowable control is applied. Let axis Ox_g, Oz_g of Earth coordinate system $Px_gy_gz_g$ to be oriented as shown in Fig. 7, and the axis Oy_g is oriented vertically upward. Then the following algorithm for angle α determination is correct.

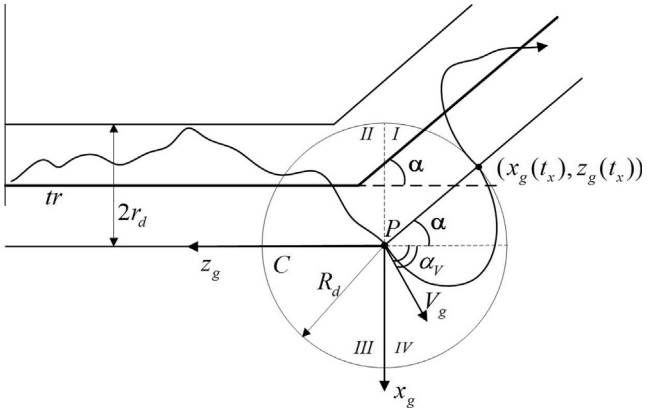


Fig. 7. To the method of geometric constraints calculation.

Make a numerical simulation of the UAV motion dynamics, starting at the point P with the velocity V_g, up to the time t_x at which one of the following conditions holds:

1. The trajectory crosses the circumference corresponding to circle C in sector I or II. Then using the geometric considerations (see Fig. 7), one may calculate the desired angle by the formula

$$\alpha(t) = \arccos\left(\frac{-z_g(t_x)}{\sqrt{x_g(t_x)^2 + z_g(t_x)^2}}\right),$$

   where $x_g(t_x), z_g(t_x)$ represents UAV coordinates in $Px_gy_gz_g$ at time t_x.
2. The trajectory crosses the circumference corresponding to the circle C in the sector IV. This case means that under the current flight parameters the UAV does not have time to make the maneuver, and the angle $\alpha = 0$. It is necessary to change the flight parameters or expand the area of the maneuver, i.e. increase the R_d value.
3. The trajectory comes to the sector III, passing through the sectors I and II. This means that $180^\circ \leqslant \alpha \leqslant 270^\circ$ for both given flight conditions and R_d value, i.e. geometric constraints are absent. It is necessary to decrease the R_d value.
4. The maximum simulation time is reached. It corresponds to the case when the UAV behavior dynamics are rather complicated: the trajectory does not leave sectors I, II or IV of the circle C. For sufficiently larger simulation time it may be concluded that the restrictions on geometry of the trajectory are also absent. It is necessary to decrease the R_d value.

The proposed method was tested for the mathematical model of the AscTec Hummingbird quadrocopter (www.asctec). Results for different flight conditions and $\alpha_v = 45^\circ$ are shown in Table 2.

Table 2
Calculated angle α for different flight conditions.

R_d (m)	V_g (m/s)	α (°)
2.7	6.5	$\alpha \geqslant 180$
2.7	6.9	31.7
2.7	7	24.4
2.7	7.5	7.8
2.7	8	0
5.4	8.5	$\alpha \geqslant 180$
5.4	9	34.5
5.4	9.5	17.5
5.4	10	0

Thus, a quite general method of geometric constraints calculation is proposed. It's based on the model of a flight dynamics, as well as some plausible assumptions. The method can be improved by taking into account the external disturbance, such as wind (see the paper Yakovlev, Makarov, and Baskin (2015) for details).

7. Conclusion

One of the obvious recent trends in science and technology is the rapid growth of the R&D areas related to UAV design. UAVs themselves are getting cheaper and thus more available both to researchers and the general public. Nowadays there exist numerous unified, moderately priced UAV platforms equipped with all the needed hardware and software to perform basic flight maneuvers. Typically this software provides seamless integration of third-party modules, which has lead to the currently observed situation, in which a lot of research is focused on the development of methods and algorithms for existing UAV platforms (rather than creating UAVs from scratch).

Thus, the direction of research that has a great importance now is studying the methods of interaction between the modules of control systems and the ways of organizing hierarchical relations between them. That is to say, studying and developing the architectures of modern UAV intelligent control systems. Such intelligent control systems are able to solve non-trivial, intelligent tasks such as planning, goal prioritization and coalition formation and thus guarantee high levels of autonomy for the UAV.

We follow the cognitive approach and believe that the ability of the system to solve the abovementioned tasks relies on its ability to model human cognitive behavior. We have proposed a multi-layered cognitive architec ture—STRL—for the intelligent control system using cognitive experimental data and psychological methods. At the same time, the STRL-based control system is also meant to deal with the non-abstract technical objects (multirotor UAVs) and take complicated dynamics and kinematics of such objects into account.

Each module of the proposed architecture is supposed to solve some specified task(s) (the architecture can be referred to as explicitly specified) and the modules are grouped into levels which encapsulate the level of abstraction: the higher the level, the more abstract the representation of input signals it uses to solve a given task. The tasks being solved on higher levels of the system are considered more sophisticated and complicated than the tasks of the lower levels. We distinguish 3 levels of control and thus separate the following 3 levels within the architecture: strategic, tactical and reactive.

The main task at the strategic level is to plan the behavior of the single member (UAV) of joint activity and coordinate it with the other members of the coalition. The distinguishing feature of the proposed strategic level is the use of knowledge representation model based on neurophysiological and psychological studies of human cognitive functions: the sign world model. Such a representation allows us to construct sophisticated algorithms for coalition behavior planning, goal-setting and communication.

UAV navigation tasks are solved at the tactical level, which is the mediator between high-level activities with the low-level ones. The main distinguishing features of the tactical level are the utilization of spatial representation of the UAV dynamics constraints and the use of the following iterative loop to perform path planning: "prediction"—"calculation geometric constraints"—"planning". We believe that such an approach (splitting planning into easily manageable stand-alone subroutines) would substantially improve the computational performance of the path planning process and thus improve the overall performance of the control system (as more navigation tasks will become solvable in the allotted time).

The main challenge for the reactive level is UAV trajectory tracking control. Both desired trajectory and UAV speed are received from the tactical level. We use the approach of nonlinear control based on a special method of solving the state-dependent Riccati equation. It greatly reduces the amount of computation and increases the control accuracy.

We suppose that any intelligent control system that implements the proposed architecture will be capable of solving a broad range of tasks and will significantly raise the degree of autonomy of the control object. STRL architecture is especially tailored to the solution of the non-trivial, complex tasks when the role distribution and coalition behavior planning is obligatory. The architecture provides a full stack of translation mechanisms: from high-level behavior plans to spatial plans to control signals needed to follow these spatial plans (trajectories). The overall computational effectiveness of the control system implementing the proposed architecture relies on the use of the original methods of interaction between the planning modules residing on different levels of the STRL hierarchy.

Acknowledgment

This work was supported by the Russian Science Foundation (Project No. 14-11-00692).

References

Albus, James S. (2002). 4D/RCS: A reference model architecture for intelligent unmanned ground vehicles. AeroSense 2002, International Society for Optics and Photonics.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review, 111*(4), 1036–1060. http://dx.doi.org/10.1037/0033-295X.111.4.1036.

Arkin, R. (1987). Motor schema based navigation for a mobile robot: An approach to programming by behavior. In *Proceedings of IEEE international conference on robotics and automation.*

Artemieva, E. U. (1999). The bases of the psychology of the subjective semantics. In I. B. Khanina (Ed.), Smisl (in Russian).

Barsalou, L. W. (1999). Perceptual symbol systems. *The Behavioral and Brain Sciences, 22*(4), 577–609. http://dx.doi.org/10.1017/S0140525X99252144, discussion 610–660.

Bills, C., Chen, J., & Saxena, A. (2011). Autonomous MAV flight in indoor environments using single image perspective cues. In *Proceedings of international conference on robotics and automation* (Shanghai, China) (pp. 5776–5783).

Bonasso, R. P., et al. (1995). Experiences with an architecture for intelligent, reactive agents. In *International joint conference on artificial intelligence*.

Borrmann, D., Elseberg, J., Lingemann, K., Nuchter, A., & Hertzberg, J. (2008). Globally consistent 3D mapping with scan matching. *Robotics and Autonomous Systems, 56*, 130–142.

Bristeau, P.-J., Callou, F., Vissiere, D., & Petit, N. (2011). The navigation and control technology inside the AR.Drone micro UAV. Preprints of the 18th IFAC world congress (Milano, Italy) (Vol. 18, Part 1, pp. 1477–1484). Milano: IFAC.

Buskey, G., Wyeth, G., Roberts, J. (2001). Autonomous helicopter hover using an artificial neural network. In *Proceedings of the IEEE conference on robotics and automation* (pp. 1635–1640).

Cimen, T. (2008). State dependent Riccati equation (SDRE) control: A survey. In *Proceedings of the 17th world congress the international federation of automatic control* (pp. 3761–3775). Seoul, Korea, July 6–11.

Cloutier, J. R., & Cockburn, J. C. (2001). The state-dependent nonlinear regulator with state constraints. In *Proc. of the American control conference* (pp. 390–395). Arlington.

Cloutier, J. R., & Stansbery, D. T. (2002). Dynamic conversion of flight path angle commands to body attitude commands. In *Proc. of the American control conference* (pp. 221–225). Anchorage.

De Filippis, L., Guglieri, G., & Quagliotti, F. (2012). Path planning strategies for UAVS in 3D environments. *Journal of Intelligent & Robotic Systems, 65*(1–4), 247–264.

Derbinsky, N., & Laird, J. E. (2010). Extending soar with dissociated symbolic memories. In *Proceedings of the remembering who we are – Human memory for artificial agents symposium, AISB 2010* (pp. 31–37).

Dmitriev, M. G., & Makarov, D. A. (2014). Smooth nonlinear controller in a weakly nonlinear control system with state depended coefficients. *Trudy Instituta Sistemnogo Analiza Rossijskoj Akademii Nauk, 64*(4), 53–58 [in Russian].

Dutka, A. S., Ordys, A. W., & Grimble, M. J. (2005). Optimized discrete-time state dependent Riccati equation regulator. In *Proceedings of the American control conference (ACC 2005)* (pp. 2293–2298). IEEE.

Edelman, G. M. (1987). In *Neural darwinism: The theory of neuronal group selection* (pp. 400). New York: Basic Books.

Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation. *Computer, 22*(6), 46–57.

Garcia, R., & Valavanis, K. (2009). The implementation of an autonomous helicopter testbed. *Journal of Intelligent and Robotics Systems, 54*, 423–454.

Gat, E. (1992). Integrating planning and reacting in a heterogenous asynchronous architecture for controlling real-world mobile robots. In *National conference for artificial intelligence*.

Harabor, D., & Grastien, Al. (2011). Online graph pruning for pathfinding on grid maps. In *25th Conference on artificial intelligence (AAAI-11)*.

Harnad, S. (1990). Symbol grounding problem. *Physica, 42*, 335–346. http://dx.doi.org/10.4249/scholarpedia.2373.

Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics, 4*(2), 100–107.

http://3drobotics.com/iris/.

http://ardrone2.parrot.com/.

http://mikrokopter.de.

http://wiki.openstreetmap.org/wiki/Database.

http://www.asctec.de/en/uav-uas-drone-products/asctec-hummingbird/.

Hutter, F., Hoos, H. H., Leyton-Brown, K., & Stutzle, T. (2009). ParamILS: An automatic algorithm configuration framework. *Journal of Artificial Intelligence Research, 36*(1), 267–306.

Ivanitsky, A. M. (1996). Brain basis of subjective experience: Information synthesis hypothesis. *Neuroscience and Behavioral Physiology, 46*(2), 251–252.

Jameson, S., Franke, J., Szczerba, R., & Stockdale, S. (2005). Collaborative autonomy for manned/unmanned teams. AHS International Forum 61, Grapevine, TX.

Kendoul, Farid (2012). Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems. *Journal of Field Robotics, 29*(2), 315–378.

Kim, H., Kim, D., Shin, J. U., Kim, H., & Myung, H. (2014). Angular rate-constrained path planning algorithm for unmanned surface vehicles. *Ocean Engineering, 84*, 37–44.

Kothari, M., & Postlethwaite, I. (2013). A probabilistically robust path planning algorithm for UAVs using rapidly-exploring random trees. *Journal of Intelligent & Robotic Systems, 71*(2), 231–253.

Kurup, U., & Lebiere, C. (2012). What can cognitive architectures do for robotics? *Biologically Inspired Cognitive Architectures, 2*, 88–99.

Kuwata, Y., Karaman, S., Teo, J., Frazzoli, E., How, J. P., & Fiore, G. (2009). Real-time motion planning with applications to autonomous urban driving. *IEEE Transactions on Control Systems Technology, 17*(5), 1105–1118.

Laird, J. E. (2012). *The soar cognitive architecture*. MIT Press.

Langley, P., Laird, J. E., & Rogers, S. (2009). Cognitive architectures: Research issues and challenges. *Cognitive Systems Research, 10*(2), 141–160. http://dx.doi.org/10.1016/j.cogsys.2006.07.004.

Leontiev, A. N. (2009). *The development of mind*. Erythros Press and Media.

Lieto, A. (2014). A computational framework for concept representation in cognitive systems and architectures: Concepts as heterogeneous proxytypes. *Proceedia Computer Science, 41*, 6–14. http://dx.doi.org/10.1016/j.procs.2014.11.078.

Likhachev, M., Ferguson, D., Gordon, G., Stentz, A., & Thrun, S. (2008). Anytime search in dynamic graphs. *Artificial Intelligence, 172*(14), 1613–1643.

Likhachev, M., & Stentz, A. (2008). R* search. In *Proceedings of the twenty-third AAAI conference on artificial intelligence*. Menlo Park, CA: AAAI Press.

Mracek, C. P., & Cloutier, J. R. (1998). Control designs for the nonlinear benchmark problem via the state-dependent Riccati equation method. *International Journal of Robust and Nonlinear Control, 8*, 401–433.

Nash, A., Daniel, K., Koenig, S., & Felner, A. (2007). Theta*: Any-angle path planning on grids. *Proceedings of the national conference on artificial intelligence* (Vol. 22 (2), pp. 1177). Menlo Park, CA; Cambridge, MA; London: AAAI Press; MIT Press, 1999.

Osipov, G. S., Panov, A. I., & Chudova, N. V. (2014). Behavior control as a function of consciousness. I. World model and goal setting. *Journal of Computer and Systems Sciences International, 53*(4), 517–529.

Saxena, A., Sun, M., & Ng, A. (2007). Learning 3-D scene structure from a single still image. *IEEE Transactions of Pattern Analysis and Machine Intelligence (PAMI), 30*(5), 1–8.

Sim, R., Elinas, P., Griffin, M., & Little, J. (2005). Vision-based SLAM using the rao-blackwellised particle filter. In *Proc. IJCAI-2005 workshop reasoning with uncertainty in robotics* (pp. 9–16).

Stanovich, K. E. (2009). Distinguishing the reflective, algorithmic, and autonomous minds: Is it time for a tri-process theory? In J. Evans & K. Frankish (Eds.), *In two minds: Dual processes and beyond* (pp. 55–88). Oxford University Press, http://doi.org/10.1093/acprof:oso/9780199230167.003.0003.

Vygotsky, L. (1986). *Thought and language*. MIT Press.

Yakovlev, K., Baskin, E., & Hramoin, I. (2015). Grid-based angle-constrained path planning. In *Proceedings of 38th annual German*

conference on AI, Dresden, Germany, September 21–25, 2015. Springer International Publishing.

Yakovlev, K. S., Makarov, D. A., & Baskin, E. S. (2015). Automatic path planning for an unmanned drone with constrained flight dynamics. *Scientific and Technical Information Processing, 42*(5), 2015.

Yap, P. (2002). Grid-based path-finding. In *Proceedings of 15th conference of the Canadian society for computational studies of intelligence* (pp. 44–55). Berlin, Heidelberg: Springer.

Zador, A. M., Dubnau, J., Oyibo, H. K., et al. (2012). Sequencing the connectome. *PLoS Biology, 10*(10).