World Academy of Science, Engineering and Technology
International Journal of Mechanical and Mechatronics Engineering
Vol:14, No:1, 2020

Real-Time Episodic Memory Construction for Optimal Action Selection in Cognitive Robotics

Deon de Jager, Yahya Zweiri, Dimitrios Makris

***Abstract*—**The three most important components in the cognitive architecture for cognitive robotics is memory representation, memory recall, and action-selection performed by the executive. In this paper, action selection, performed by the executive, is defined as a memory quantification and optimization process. The methodology describes the real-time construction of episodic memory through semantic memory optimization. The optimization is performed by set-based particle swarm optimization, using an adaptive entropy memory quantification approach for fitness evaluation. The performance of the approach is experimentally evaluated by simulation, where a UAV is tasked with the collection and delivery of a medical package. The experiments show that the UAV dynamically uses the episodic memory to autonomously control its velocity, while successfully completing its mission.

***Keywords*—**Cognitive robotics, semantic memory, episodic memory, maximum entropy principle, particle swarm optimization.

I. Introduction

COGNITIVE robotics is described as the study of robots with cognitive functions, such as perception, attention, anticipation, planning, memory, learning, and reasoning, inspired by human cognition. Human cognition is not trivial and to study and understand it, various computational models and architectures have been devised. For example, statistical models for cognition [1], and cognitive architectures, such as Adaptive control of thought (ACT) [2], State operator and result (SOAR) [3] and Neural Engineering Objects (Nengo) [4]. These architectures show the complexity in the interaction between neuro-cognitive processes which, inevitably, apply to robo-cognitive processes as well. Arguably, the most important cognitive function is the working memory, which is a collection of neuro-cognitive processes: memory representation, memory recall, action selection and execution. Memory representation can be further described in terms of episodic (short-term) and semantic (long-term) memory.

In many robo-cognitive architectures today, control models are learned through methods, such as artificial neural networks (ANNs), to simulate memory representation, memory recall and the executive functions of the brain. The models represent memory through synaptic weight assignment, which is adjusted during a learning process. When presented with an input stimulus, the model "recall" learned facts by applying the synaptic weight and input stimulus to an activation function. Unfortunately, for many real-world cognitive robotic applications, the approach of a priori learning of behavioral models is not always effective. Robots deployed in remote, unknown and dynamic locations, cannot risk catastrophic failure. They do not have the time to learn new complex solutions from the start, every time the environment changes.

The robo-cognitive architecture of a remotely deployed robot must provide an efficient and simple means of updating the semantic and episodic memory. Action selection and execution, based on these memories must automatically adapt according to the new memories. Moreover, when the memory items are complex structures with multiple characteristics (such as the logically conditioned state-transitions used in this study), optimal memory recall becomes extremely complex and computationally expensive.

This study examines real-time episodic construction using a set-based particle swarm optimization (SPSO) algorithm, which evaluates the fitness of semantic memory items using an adaptive entropy-based memory quantification (AEMQ) algorithm. The algorithm uses real-time environmental stimuli and cues to statistically quantify and evaluate the semantic memory. The AEMQ algorithm employs the maximum entropy principle (MEP) [5] to provide a probability distribution over all the characteristics of the semantic memory item, for fitness evaluation. The result of the SPSO is an optimal set of memory items, i.e. the episodic memory from which the executive uses the probability distribution of each item to select the best memory item and execute a suitable action.

The performance of the SPSO and AEMQ algorithms are experimentally evaluated with an unmanned aerial vehicle (UAV) benchmark mission: collecting and delivering a package, before returning to a charging station.

II. Working Memory in Cognitive Architecture

A. Common Cognitive Architectures

Although there are still many unanswered questions regarding the functions of the human brain, there seems to be a common understanding about the basic architectures of human cognition.

Computational architectures, which mimic cognitive processes in the human brain, have been developed to examine and understand human cognition. These include the Adaptive control of thought–rational (ACT-R) architecture [7], the Semantic pointer architecture unified network (SPAUN) [4] and the SOAR architecture [3].

Deon de Jager is with the Science, Engineering and Computer Science Department at Kingston University, River House, 53-57 High street, Kingston-upon-Thames, Surrey, KT1 1LQ (corresponding author, phone: +44 798-502-5517; e-mail: k0952100@ kingston.ac.uk).

Yahya Zweiri and Dimitrios Makris are with the Science, Engineering and Computer Science Department at Kingston University, River House, 53-57 High street, Kingston-upon-Thames, Surrey, KT1 1LQ (e-mail: Y.Zweiri@kingston.ac.uk, D.Makris@kingston.ac. uk).

The *central executive* is responsible for the cognitive processes of memory classification, memory representation, recall, action selection and action execution. Collectively, these processes constitute the *working memory*.

B. Working Memory Models

A number of working memory models have been defined over the years. Baars and Nicole [6] presents a functional framework for human cognition, where working memory is composed of the central executive and working storage. The working storage is created from sensory memory (verbal and visuospatial) and long-term, (stored) memory. The working memory is used in the action selection and execution process. Long-term memory will be referred to as sematic memory

Tulving [8] classifies memory as non-declarative, semantic and episodic.

Arguably, the most widely used model is Baddeley's model of working memory [9]. In Baddeley's model, the central executive processes visuospatial, phonological and long-term semantic memory, and creates the episodic buffer. The episodic buffer performs the same role as the working storage memory in [6].

A different approach is presented in Cowan's attentional focus theory [9] model. In Cowan's model, instead of types of memory being classified separately and distributed according to the cognitive functionality, all memory is stored as long-term memory. When memory receives attention, it becomes salient and closely stored memory is activated at the same time. Activated memory is of interest, as it is potentially relevant to the context and may become attentional. The central executive uses the memory with attentional focus for action selection and execution.

For remotely-deployed exploratory robots, Cowan's model provides a simpler architecture, with fewer possible points-of-failure. This study proposes a robo-cognitive architecture, which combines some features from both Baddeley and Cowan's models, particularly the episodic buffer. However, for computational practicality, memory representation is abstracted from the central executive.

C. Real-Time Episodic Memory Construction

In this study, a robo-cognitive architecture is proposed for the real-time construction of episodic memory. An UAV is used for illustration.

In this study, long-term memory will be referred to as sematic memory and short-term memory will refer to sensory and episodic memory.

Semantic memory is the long-term memory provided by a domain expert. The domain expert also provides cues (or missions) which defines the objectives of the robot. The central executive recalls, quantifies and optimizes semantic memory, in real-time, subject to the cues and stimuli. Since the process is dynamic and real-time, the optimal memory constructed by the central executive is episodic, and used for selecting and executing the optimal action.

Memory optimization is done using a particle swarm optimization (PSO) approach. The standard PSO (StdPSO) [10] algorithm is mostly used for the optimization of real problems. For discrete set-based optimization problems, the StdPSO operators were modified and the SPSO algorithm was created [11], [12].

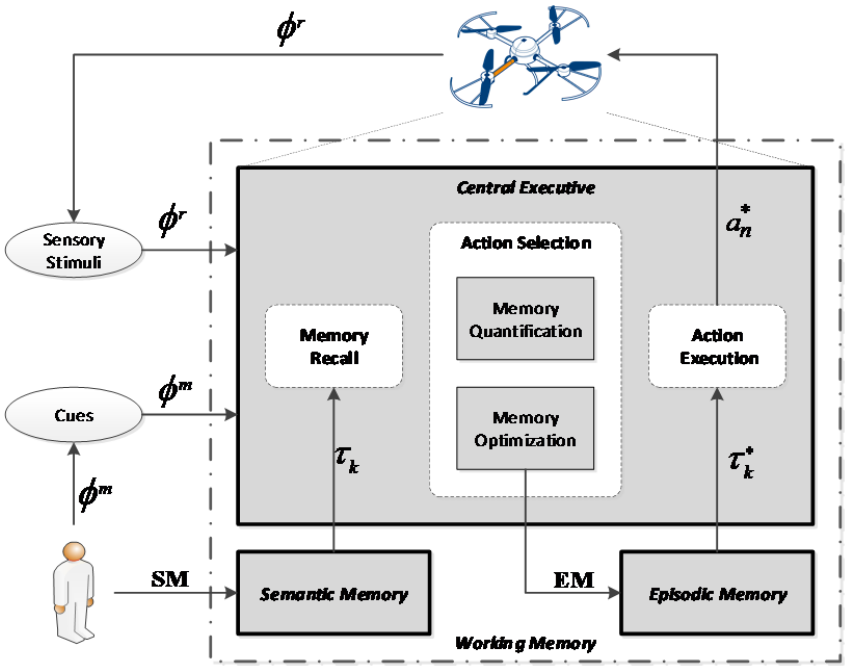

Fig. 1 Real-time episodic memory optimization

Since semantic memory in this study is defined as a set of discrete memory items and the SBPSO is used for the memory

optimization process, the SPSO uses the AEMQ algorithm (discussed in detail in Section III) to evaluate the fitness of each semantic memory item. The optimal set of semantic memory items forms the episodic memory, from which the optimal episodic memory item is selected. Finally, the action corresponding to the selected episodic memory item is executed.

III. Methodology

The AEMQ algorithm is described in terms of a UAV, executing a mission. The AEMQ uses sensory stimuli, received from the UAV, along with a set of cues and semantic memory, provided by a domain expert, to quantify a semantic memory item. Assume that the semantic memory of UAV is defined as the set of all valid state-transitions, then it is the task of the executive to construct the optimal episodic memory from which it is to select and pass the optimal control command to the UAV flight controller.

Note that, since the episodic memory generated is temporal and will become obsolete when either the sensory stimuli or cues change, the approach is more similar to Cowan's attentional focus theory model.

A. Sensory Stimuli

Input stimuli is defined as:

$$\Phi^r = \{\varphi_1^r, \varphi_2^r, \dots, \varphi_{n_{\Phi^r}}^r\} \tag{1}$$

where φ_i^r, $i = 1,..,n_{\Phi^r}$ is the evidence parameter representing the evidence received from the environment.

B. Cues

The cues are defined are defined as:

$$\Phi^m = \{\varphi_1^m, \varphi_2^m, \dots, \varphi_{n_{\Phi^m}}^m\} \tag{2}$$

where $\varphi_j^m \in [lb^{m_j}, ub^{m_j}]$, $j = 1,..,n_{\Phi^m}$ defines a cue, constrained to specified lower and upper boundaries.

C. Semantic Memory

The semantic memory (SM) is defined as the set of state-transitions which governs the behavior of the UAV:

$$\mathbf{SM} = \{\tau_1, \tau_2, \dots, \tau_{n_{LTM}}\} \tag{3}$$

where $\tau_k \in \mathbf{SM}$, k = $(1, \dots, |\mathbf{SM}|)$ is a memory item, representing a state-transition in the **SM**. The state-transition is a tuple,

$$\tau_k = (\nu, \eta, \mathcal{S}_\alpha, \mathcal{S}_\beta, \mathcal{A}, \mathcal{F}) \tag{4}$$

where $\nu = \{0,1\}$ indicates whether the transition is valid, $\eta \in \mathbb{Z}^+$ is an objective identifier assigned to the transition, $\mathcal{S}_\alpha$ and $\mathcal{S}_\beta$ are the start and end states of the state-transition, respectively, $\mathcal{A} = \{a_1, \dots, a_{n_\mathcal{A}}\}$ is a set of actions and $\mathcal{F} = \{p_1, p_2, \dots, p_{n_\mathcal{F}}\}$ is the trigger formula for the transition and consisting of a set of simple logic propositions.

Each proposition $p_l \in \mathcal{F}, l = (1, \dots, n_\mathcal{F})$ is defined by a domain expert and is a tuple,

$$p_l = (\varphi_i^r,\ logical\_operator, \varphi_j^m\ ) \tag{5}$$

where the sensory stimuli and cues are related by a logical_operator, from the set $\{>, <, =\}$, to form simple propositions of the form:

$$(\varphi_i^r > \varphi_j^m), (\varphi_i^r < \varphi_j^m) \text{ and } (\varphi_i^r = \varphi_j^m) \tag{6}$$

Any non-numeric argument is discretized to a numeric value, prior to quantification of $\mathcal{F}$.

The indicator ν, the objective identifier η, the actions $\mathcal{A}$ and all the propositions p_l are defined and maintained by the domain expert.

D. Memory Quantification Preparation

In order to perform the quantification of a state-transition τ_k, a problem-specific model is constructed before it is presented to the MEP equation for quantification. Given a state-transition $\tau_k \in \mathbf{SM}$ the model is formally defined as a tuple,

$$\mathcal{M}_{\tau_k} = (\mathbf{V}, \mathbf{X}, \mathbf{F}, \mathbf{\Lambda}) \tag{7}$$

The set of variables are represented by $\mathbf{V} = \left\{\{v^{\mathbb{Q}}\} \cup , \{v_1^{\mathbb{P}}, v_2^{\mathbb{P}}, \dots v_{n_\mathbb{P}}^{\mathbb{P}}\} \cup \{v_1^{\mathbb{A}}, v_2^{\mathbb{A}}, \dots v_{n_\mathbb{A}}^{\mathbb{A}}\}\right\}$ where $v^{\mathbb{Q}}$ is the query variable, $v_p^{\mathbb{P}}$, $p = 1, \dots, n_\mathbb{P}$ is a predictor variable, representing a proposition in the trigger formula and $v_l^{\mathbb{A}}$, $l = 1, \dots, n_\mathbb{A}$ is an association variable. Note that, since the propositions are independent, they will not have any effect on the query variable, unless relevant associations are defined between the query variable and appropriate predictor variables. The associations are problem-specific and are defined by the user.

Let $m_{\tau_k} = |\{v^{\mathbb{Q}}\} \cup \{v_1^{\mathbb{P}}, v_2^{\mathbb{P}}, \dots v_{n_\mathbb{P}}^{\mathbb{P}}\}|$, and $n_{\tau_k} = 2^{m_{\tau_k}}$, then a $m_{\tau_k} \times n_{\tau_k}$ constraint matrix, $\mathbf{X}$ is the state space of the trigger formula and defines all the joint statements of $\{v^{\mathbb{Q}}\} \cup \{v_1^{\mathbb{P}}, v_2^{\mathbb{P}}, \dots v_{n_\mathbb{P}}^{\mathbb{P}}\}$. A binary constraint function, $F(X = x_{ij})$, $i \in n_{\tau_k}$ and $j \in m_{\tau_k}$ assigns a boolean constraint to each variable in the state space. Let $n_V = (1 + n_\mathbb{P} + n_\mathbb{A})$, then vector $\mathbf{F} = (\langle F_1 \rangle, \langle F_2 \rangle, \dots, \langle F_{n_F} \rangle)$, $n_F = n_V$ are constraint averages for each of the variables in $\mathbf{V}$. The vector $\mathbf{\Lambda} = (\lambda_1, \lambda_2, \dots \lambda_{n_\Lambda})$, $n_\Lambda = n_V$, represents the Lagrange multipliers, calculated for each variable in $\mathbf{V}$, using (17).

E. Representing Real-Time Stimulus and Cue Relationship

The degree of belief in a memory item being recalled is influenced by current stimuli and current cue. Therefore, the quantification of the memory item must consider in both. Each constraint average $\langle F_{n_F} \rangle \in \mathbf{F}$ in (7) represents the degree of belief in a proposition, given the real-time stimuli (evidence) received from the environment. In this study, the constraint average is calculated by interpreting a proposition as a degree of believe, (probability), derived from a distance calculation. Fig. 2 illustrates two example state-transitions with their corresponding transition rules (propositions). A constraint

average for the proposition is calculated by measuring the progress of the current sensory stimulus φ_i^r, relative to the operational bounds of the mission task. The result is a probability assigned to the proposition.

The constraint averages, **F** are calculated as follows: Firstly, given the proposition p_l, calculate the total operation distance d_j^m, using the upper and lower bounds of the mission argument:

$$d_j^m = ub_j^m - lb_j^m \tag{8}$$

Calculate the current distance d_i^r of the sensory stimulus, φ_j^r with respect to the upper and lower bounds of the cue, φ_i^m, according to the logical operation of the proposition:

$$d_i^r = \begin{cases} \varphi_i^r - lb_j^m \; ; if \; p_l = (\varphi_i^r > \varphi_j^m) \\ ub_j^m - \varphi_i^r \; ; if \; p_l = (\varphi_i^r < \varphi_j^m) \\ 1 \qquad\quad ; if \; p_l = (\varphi_i^r \neq \varphi_j^m) \\ 1 \qquad\quad ; if \; p_l = (\varphi_i^r = \varphi_j^m) \end{cases} \tag{9}$$

Use (8) and (9) to calculate a real valued distance, in the range $[0,1]$, for the proposition:

$$Pr(p_l) = \frac{d_i^r}{d_j^m} \tag{10}$$

where $Pr(p_l)$ represent the relative remaining distance of φ_i^r, within the boundaries lb_j^m and ub_j^m as a probability. Once the distances for each proposition have been calculated, the distances for each of the joint statements can be calculated. To illustrate, let $v^{\mathbb{Q}} = p_0$, $v_1^{\mathbb{P}} = p_1$ and $v_2^{\mathbb{P}} = p_2$, then the state space consists of $2^3 = 8$ joint statements. The joint distances, for the predictor variables are calculated as follows:

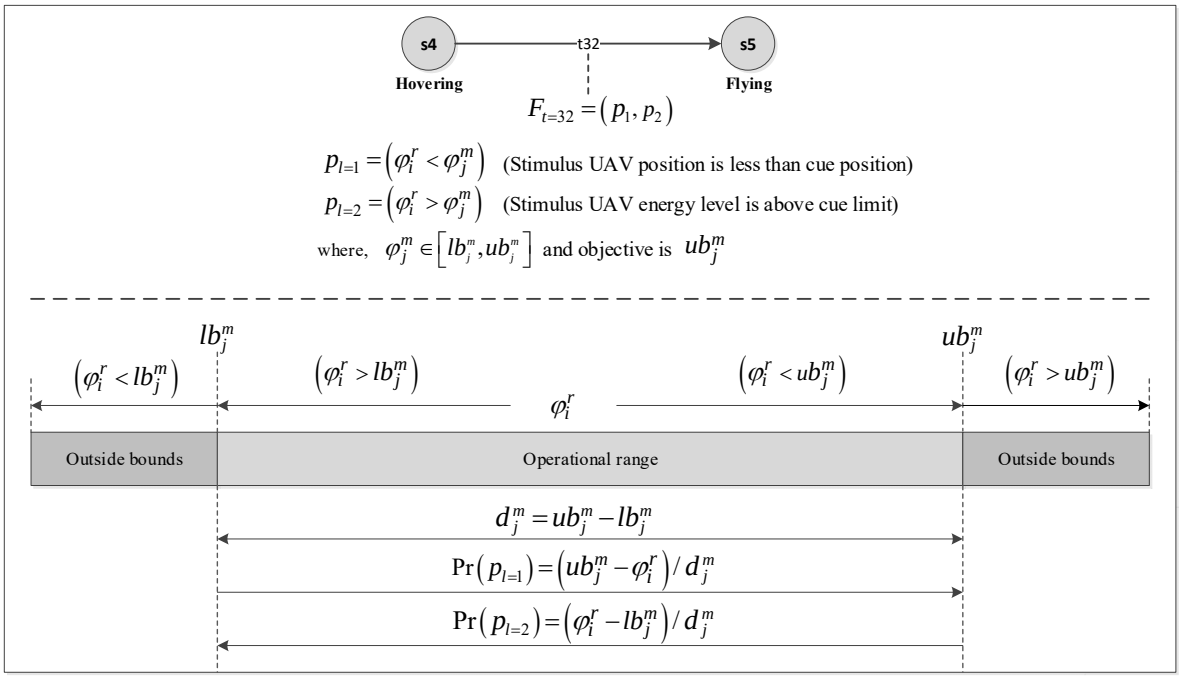

Fig. 2 Interpreting stimuli of two state transitions as probabilities

$$d_{p_1 p_2} = d_{p_1} + d_{p_2} \tag{11}$$

$$d_{p_1 \overline{p_2}} = d_{p_1} + (1 - d_{p_2}) \tag{12}$$

$$d_{\overline{p_2} p_2} = (1 - d_{p_1}) + d_{p_2} \tag{13}$$

$$d_{\overline{p_1} \overline{p_2}} = (1 - d_{p_1}) + (1 - d_{p_2}) \tag{14}$$

The overall distance d_f, represented by the probability distribution over all the propositions of the trigger formula, is calculated by:

$$d_f = (d_{p_1 p_2} + d_{p_1 \overline{p_2}} + d_{\overline{p_2} p_2} + d_{\overline{p_1} \overline{p_2}}) \tag{15}$$

With all the joint distances of the joint statements available, the respective constraint averages can now be calculated. Firstly, the constraint average $\langle F_1 \rangle$ of the query variable p_0 is set to 1.0. The constraint averages for the predictor and association variables are then set as follows:

$$\mathbf{F} = \left(d_{p_0}, \frac{(d_{p_1 p_2} + d_{p_1 \overline{p_2}})}{d_f}, \frac{(d_{p_1 p_2} + d_{\overline{p_1} p_2})}{d_f}, \frac{(d_{p_1 p_2} + d_{p_1 \overline{p_2}})}{d_f}, \frac{(d_{p_1 p_2} + d_{\overline{p_1} \overline{p_2}})}{d_f}, \frac{d_{p_1 p_2}}{d_f} \right) \tag{16}$$

Next, the Lagrange multipliers are determined. The duality between the Lagrange multipliers and the user-defined constraint averages, allows the Legendre transform to be used to derive the Lagrange multipliers:

$$\mathcal{L}_{trans} = \mathbf{\Lambda} = min_{\lambda_k}\left(ln\, Z(\lambda_1, \lambda_2, \dots \lambda_k) - \sum_{j=1}^{m_{\tau_k}} \lambda_j \langle F_j \rangle\right) \quad (17)$$

The multipliers are derived by varying the values of λ_k while keeping the constraint average, $\langle F_j \rangle$ fixed, until $\mathcal{L}_{trans}$ reaches a minimum.

F. Memory Quantification

Jaynes's seminal paper on the MEP [13] utilizes the principles of information theory, based on Shannon's work on communication theory [14]. Jaynes's MEP is particularly useful for inference, where information is incomplete.

Given the model $\mathcal{M}_{\tau_k}$, the probability distribution, $\mathbf{Q} = \left(q_1, q_2, \dots q_{n_\mathbf{Q}}\right)$, $n_\mathbf{Q} = n_{\tau_k}$over the variables (propositions) of the trigger formula can now be calculated. Given the $m_{\tau_k} \times n_{\tau_k}$ constraint matrix and let $i \in n_{\tau_k}$ and $j \in m_{\tau_k}$, the MEP is formally defined as:

$$\left(q_i | \mathcal{M}_{\tau_k}\right) = \frac{1}{Z(\lambda_1, \lambda_2, \dots \lambda_k)} e^{-\sum_{j=1}^{m_{\tau_k}} \lambda_j F_j (X = x_i)} \quad (18)$$

where, $Z(\lambda_1, \lambda_2, \dots \lambda_k) = \sum_{i=1}^{\tau_k} e^{\sum_{j=1}^{m_{\tau_k}} \lambda_j F_j (X = x_i)}$. Z is the partition function which ensures the probabilities are assigned between 0 and 1. The Lagrange multipliers are represented by λ_j, $j = 1, \dots, k$ and $F_j(X = x_i)$ assigns a real-world, domain-specific constraint, to the state i of variable j.

The memory can now be evaluated, using the probability distribution $\mathbf{Q}$. For example, a memory item with two characteristics (rules) rules, τ_1 and τ_2, will result in a probability distribution of $\mathbf{Q} = (q_1, q_2, q_3, q_4)$, where, $q_1 = pr(\tau_1, \tau_2), q_2 = pr(\tau_1, \bar{\tau}_2), q_3 = pr(\bar{\tau}_1, \tau_2), q_4 = pr(\bar{\tau}_1, \bar{\tau}_2)$.

In this study, the quantification Π_{τ_k} of the memory item $\tau_k \in \mathbf{SM}$ is defined as:

$$\Pi_{\tau_k} = \mathcal{v} \times \mathcal{p} \times q_1 \quad (19)$$

where, $\mathcal{v} = 1$ indicates a valid state-transition and $\mathcal{v} = 0$ indicate an invalid state-transition. A reward $\mathcal{p}$ is applied as:

$$\mathcal{p} = \begin{cases} 0 & ; if\left(\tau_{k_\alpha} \neq \mathcal{S}_{\alpha_{current}}\right) \\ 1 & ; if\left(\tau_{k_\alpha} = \mathcal{S}_{\alpha_{current}}\right) \end{cases}$$

where, τ_{k_α} is the start state of the state-transition currently being evaluated and $\mathcal{S}_{\alpha_{current}}$ is the current state, previously selected by the executive.

Note that the quantification is user-defined and problem specific. It is left to the reader to define problem-specific quantifications using the probability distribution, $\mathbf{Q}$.

G. Executive Action-Selection and Execution

The SPSO uses the memory quantification (19) to evaluate the fitness of the memory item (i.e. state-transition in this study). Ultimately, the optimal set of memory items represent the episodic memory which is used by the executive to select the optimal action to take next.

$$\mathbf{EM} = \left\{\tau_1^*, \tau_2^*, \dots, \tau_{n_\mathbf{SM}}^*\right\} \quad (20)$$

where, $\tau_k^* \in \mathbf{EM}$, $\mathrm{k} = (1, \dots, |\mathbf{EM}|)$ is an optimal memory item which represents a state-transition in the **SM**.

From the episodic memory **EM**, the executive can now select the optimal memory item (state-transition) and pass the associated action $a_\mathrm{n} \in \mathcal{A}$, defined in (4), for execution.

Note that it is possible that $|\mathbf{EM}| > 1$. This is because the memory quantification is performed statistically and may be equal for some memory items. Again, it is left to the reader to define (19) as appropriate for the problem at hand. In this study, only valid state-transitions evaluate to non-zero.

IV. Experiment

A. Experiment Design

The methodology is experimentally evaluated by simulation, where a UAV autonomously executes a "medical delivery" mission. The mission is defined as a list of cues (tasks) and is described as follows:

Fig. 3 UAV Mission design

From the Home (I) location, arm the motors, ascend to a specified operational height and fly to the Collection point (II). Descend and collect the cargo, then ascend to the specified operational height and fly to the Delivery point (III). Descend at the delivery point and deliver the cargo. Ascend to a new operational height and fly to the Charging point (IV) for

recharging. Descend on the charging point and disarm the motors. The mission is illustrated graphically in Fig. 3. Table I lists the UAV states.

TABLE I
UAV STATES

S1 – Motors Off	S2 – Motors On	S3 - Ascending
S4 - Hovering	S5 - Flying	S6 - Descending
S7 - Rotating	S8 – Acquiring Cargo	S9 – Releasing Cargo

TABLE II
UAV STATE-TRANSITIONS

	s1	s2	s3	s4	s5	s6	s7	s8	s9
s1	t_1	***t_2***	t_3	t_4	t	t_6	t_7	t_8	t_9
s2	***t_{10}***	t_{11}	***t_{12}***	t	t_{14}	t_{15}	t	t_{17}	t_{18}
s3	t_{19}	t_{20}	***t_{21}***	***t_{22}***	t_{23}	t_{24}	***t_{25}***	t_{26}	t_{27}
s4	t_{28}	t_{29}	***t_{30}***	***t_{31}***	***t_{32}***	***t_{33}***	***t_{34}***	***t_{35}***	***t_{36}***
s5	t_{37}	t_{38}	t_{39}	***t_{40}***	***t_{41}***	t_{42}	***t_{43}***	t_{44}	t_{45}
s6	***t_{46}***	t_{47}	t_{48}	***t_{49}***	t_{50}	***t_{51}***	t_{52}	t_{53}	t_{54}
s7	t_{55}	t_{56}	***t_{57}***	***t_{58}***	***t_{59}***	t_{60}	***t_{61}***	t_{62}	t_{63}
s8	t_{64}	t_{65}	t_{66}	t_{67}	t_{68}	t_{69}	t_{70}	t_{71}	t_{72}
s9	t_{73}	t_{74}	t_{75}	t_{76}	t_{77}	t_{78}	t_{79}	t_{80}	t_{81}

Table II shows the SM, representing all the state-transitions of the UAV and is defined by the domain expert. Complex (logic-based) state-transitions are shown in shaded/italic/bold.

B. Experiment Setup

The mission was simulated using the AirSim/Unity simulator, with flight-control routines developed in C++. The robo-cognitive architecture was implemented using .NET/C# code and integrated with the simulator via the Redis in-memory data cache.

C. Results

Figs. 4-6 show the behavior of the UAV in the simulation. Fig. 4 shows the successful collection of the medical package. As the UAV approaches its waypoint, it must adjust its velocity in order to avoid overshooting. Fig. 5 shows the autonomous velocity adjustment, as the UAV approaches the delivery point.

As the UAV approaches its target, the fitness reduces from 0.35 to 0.28 and the velocity of the UAV (indicated in the window left) is automatically adjusted accordingly from $8.00\ m/s$ to $2.24\ m/s$. Fig. 6 shows the successful delivery of the medical package as per the "delivery" cue in the mission.

A video of the full mission can be viewed at [15].

V. DISCUSSION

A. Autonomous Velocity Control

Fig. 7 shows the various velocity adjustments, related to the fitness (calculated using (19)), for the whole mission. The graph shows velocity/fitness reductions at the points of "collection", "deliver" and "charging".

B. Resulting State-Flow

Fig. 8 shows the state flow generated for the mission, where each state transition corresponds to an optimal action selected and executed.

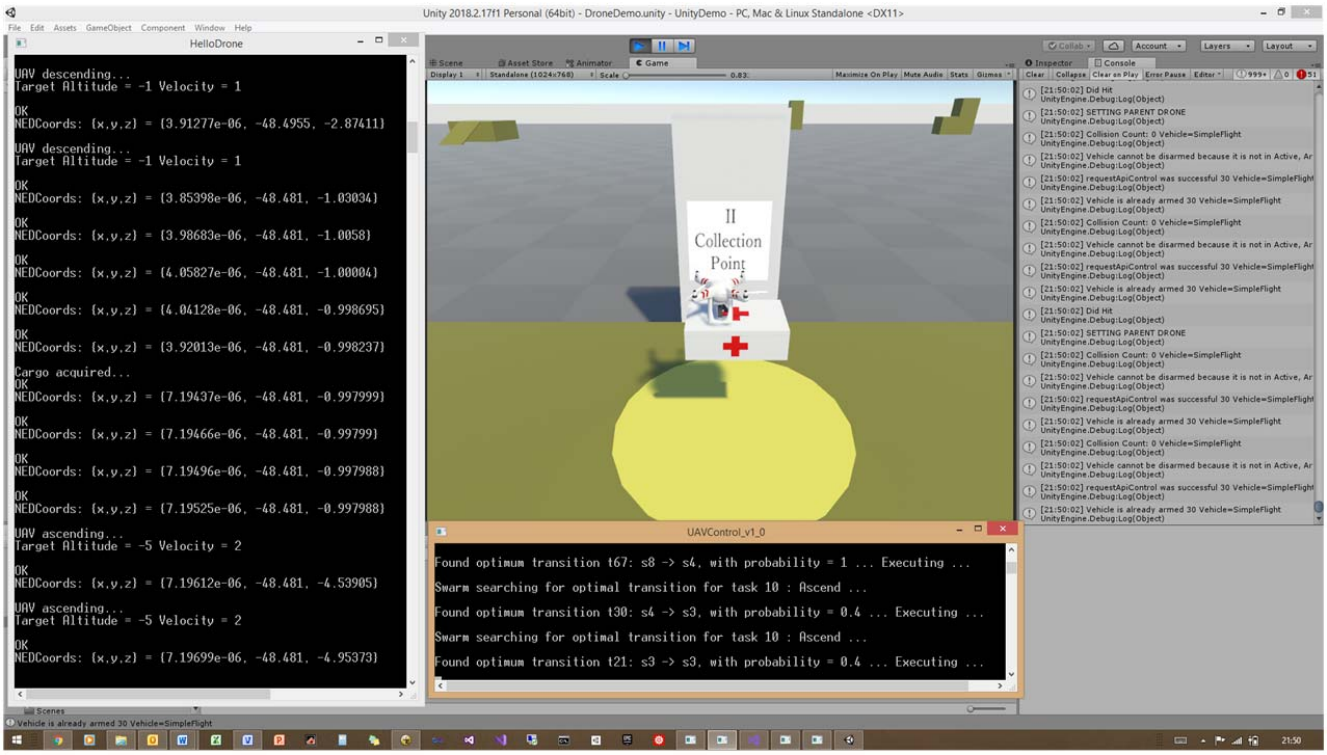

Fig. 4 Successful collection of the medical package

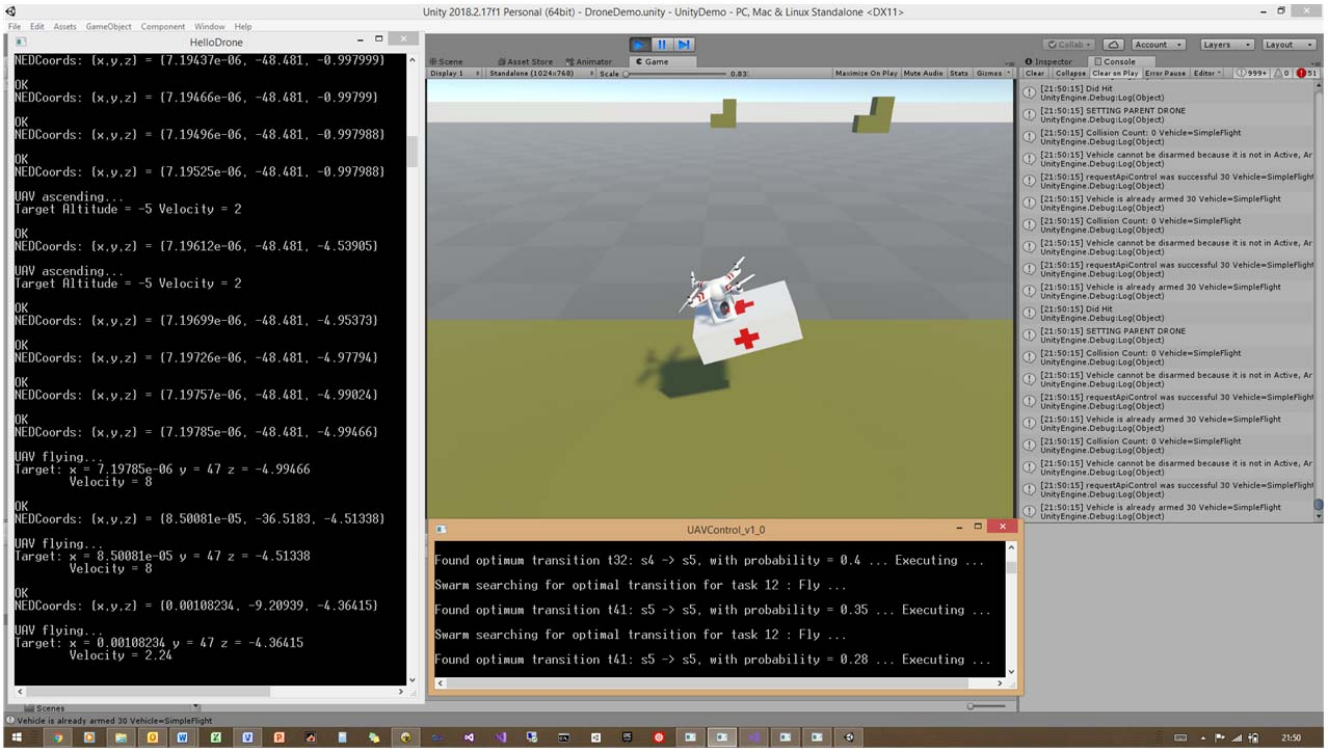

Fig. 5 Autonomous velocity adjustment en-route

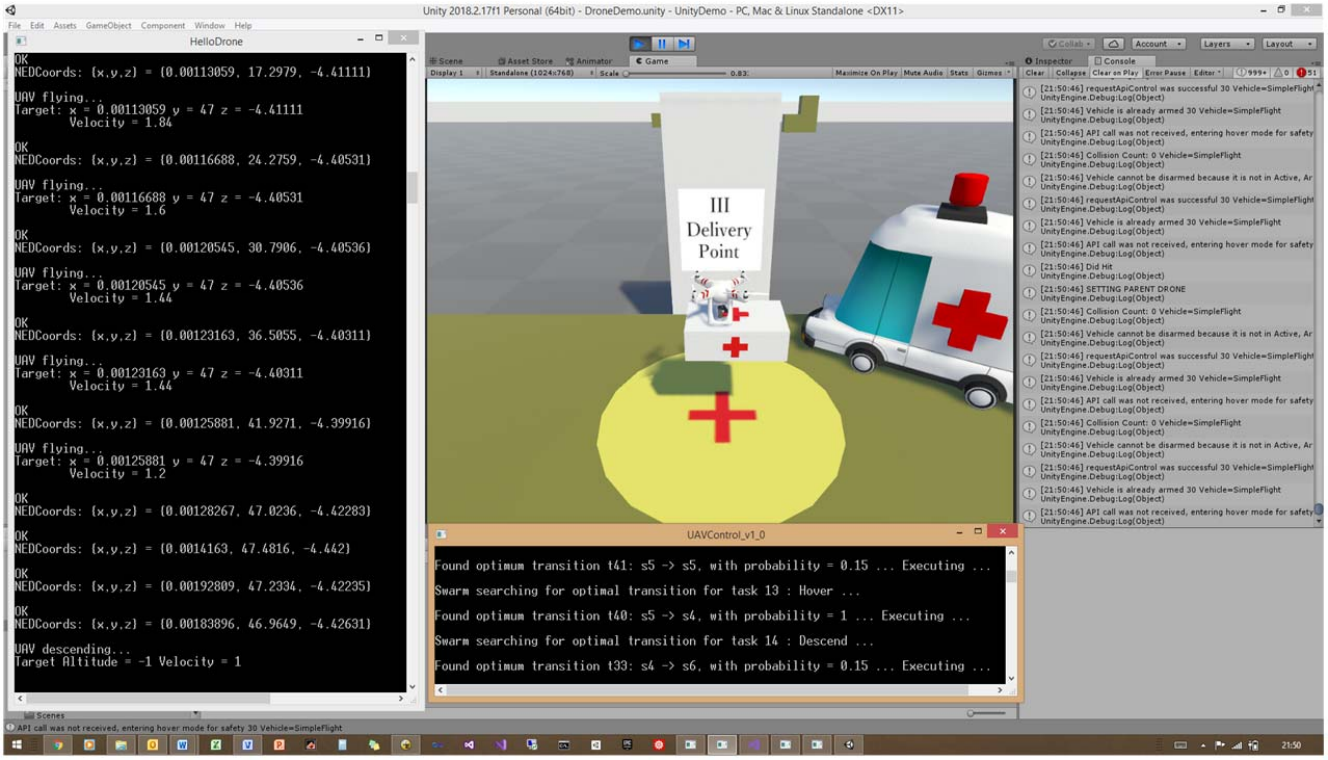

Fig. 6 Successful delivery of medical package

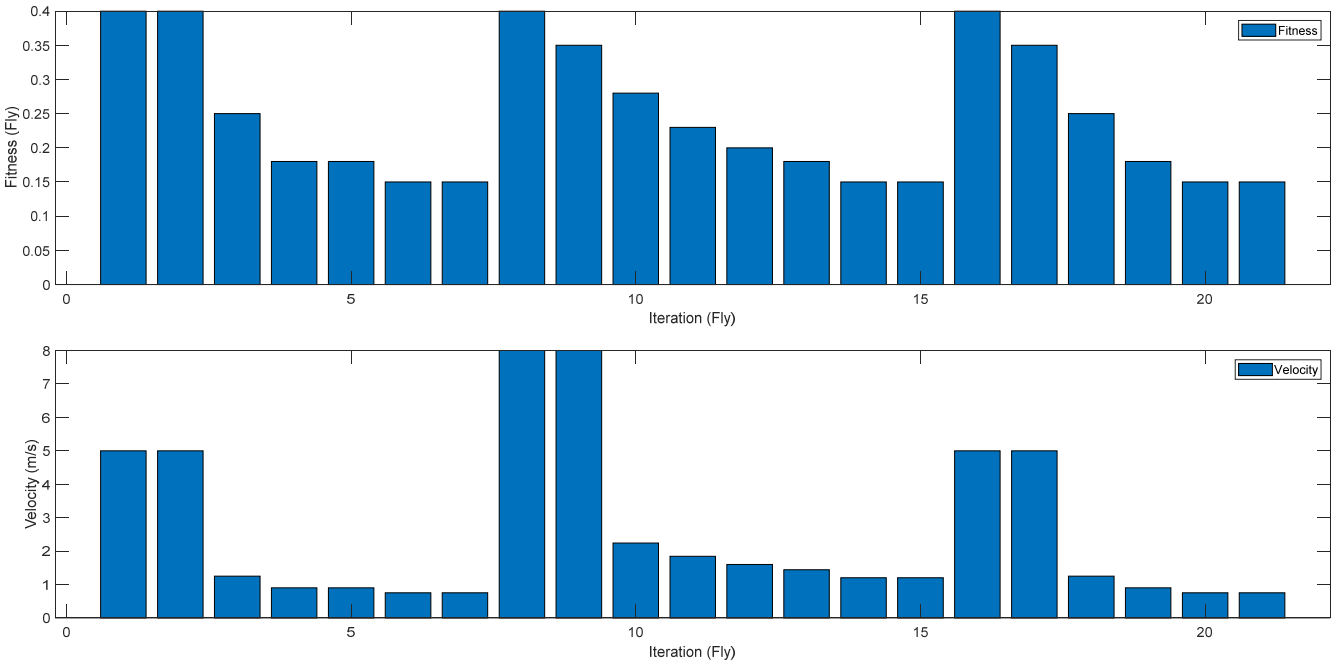

Fig. 7 Fitness/Velocity adjustments

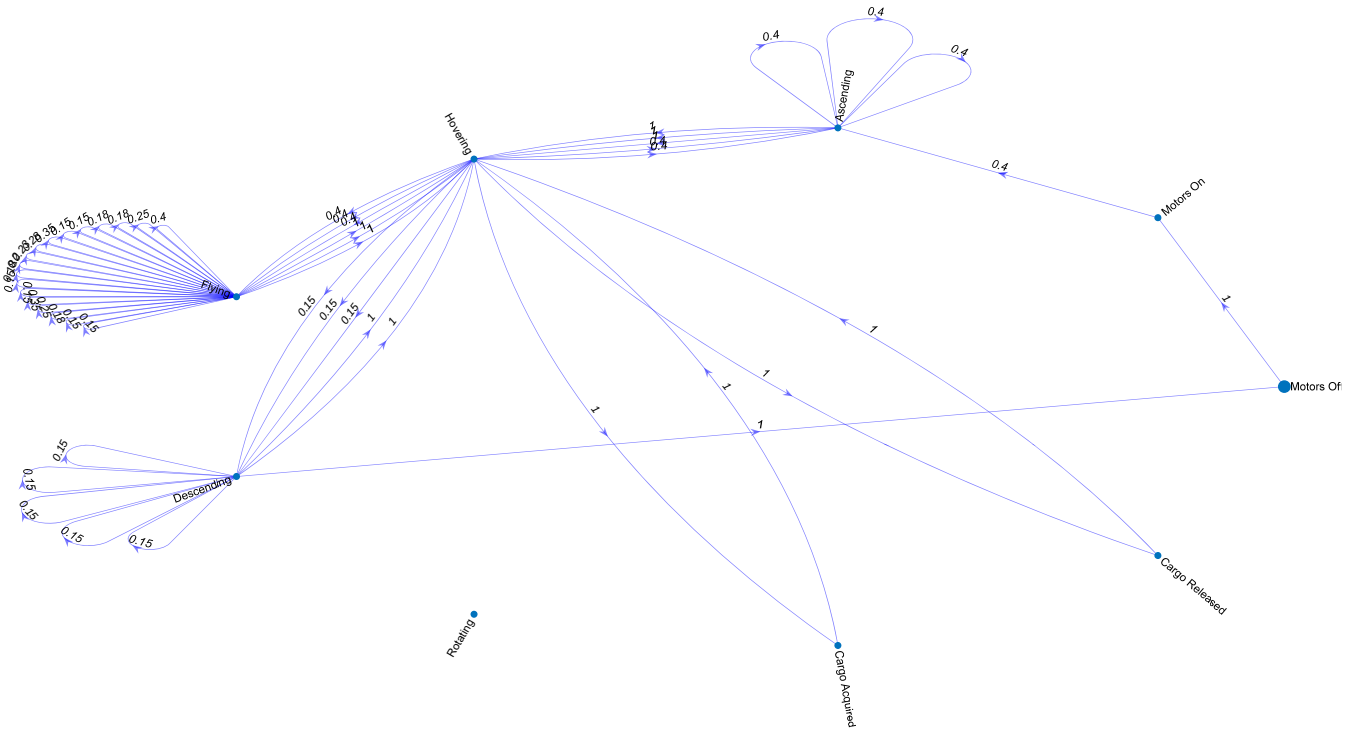

Fig. 8 Resulting action selections

Each cue (task) is repeatedly executed (see "Fly" state in Fig. 8), until the task objective was reached. Due to some lagging between the AirSim simulator and Unity games engine, it was observed that, at high velocity, the UAV would overshoot its target destination in the Unity games engine, but the target position in AirSim would be correct, causing the UAV to miss its objective. However, with the autonomous and dynamic velocity control, the UAV would autonomously correct its positioning, by repeating the task, while constantly reducing its velocity according to the fitness of the task. At low velocity, the positioning of the UAV was more accurate and it could achieve its objectives. With the autonomous velocity control, the UAV was able to successfully collect and deliver the medical package in the simulated mission.

VI. Conclusion

In real-world scenarios, semi-autonomous systems, such as exploratory robots, operate in environments which may constantly change. Therefore, it must be simple and computationally inexpensive to alter a robot's SM and/or cues in real-time. This is especially important for remotely deployed robotic systems, such as extra-terrestrial exploration robots, where communication time and bandwidth are at a premium.

Future study could include the application of metamemory [9], to further optimize memory recall, through real-time clustering of the episodic memory items. This could extend the approach in this study to include the generation of episodic memory for multiple, concurrent tasks.

References

[1] S. Lewandowsky and S. Farrell, *Computational Modeling in Cognition: Principles and Practice*. Sage Publications Inc., 2011, p. 357.
[2] J. R. Anderson, *The Architecture of Cognition*. Harvard University Press, 1983, p. 345.
[3] J. E. Laird, *The SOAR Cognitive Architecture*. The MIT Press, 2012.
[4] C. Eliasmith, *How to Build a Brain* (Oxford Series on Cognitive Models and Architecture). United States: Oxford University Press, 2013, p. 456.
[5] D. J. Blower, *Information Processing - The Maximum Entropy Principle*. CreateSpace Independent Publishing Platform, 2013.
[6] B. J. G. Baars, Nicole M., *Fundamentals of Cognitive Neuroscience - A Beginner's Guide*, Second ed. Academic Press - Elsevier, 2018.
[7] J. R. Anderson, D. Bothell, M. D. Byrne, S. Douglass, C. Lebiere, and Y. Qin, "An integrated theory of the mind," *Psychological Review,* vol. 111, 4, pp. 1036-1060, 2004.
[8] E. Tulving, "How many memory systems are there," *American Psychologist,* vol. 40, pp. 385-398, 1985.
[9] G. A. Radvansky, *Human Memory*, Third ed. Routledge, 2017.
[10] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Micro Machine and Human Science, 1995. MHS '95., Proceedings of the Sixth International Symposium on*, 4-6 Oct 1995 1995, pp. 39-43, doi: 10.1109/mhs.1995.494215.
[11] C. Wei-Neng, Z. Jun, H. S. H. Chung, Z. Wen-Liang, W. Wei-gang, and S. Yu-Hui, "A Novel Set-Based Particle Swarm Optimization Method for Discrete Optimization Problems," *Evolutionary Computation, IEEE Transactions on,* vol. 14, no. 2, pp. 278-300, 2010, doi: 10.1109/tevc.2009.2030331.
[12] J. Langeveld and A. Engelbrecht, "Set-based particle swarm optimization applied to the multidimensional knapsack problem," (in English), *Swarm Intelligence,* vol. 6, no. 4, pp. 297-342, 2012/12/01 2012, doi: 10.1007/s11721-012-0073-4.
[13] E. T. Jaynes, "Information Theory and Statistical Mechanics," *Physical Review,* vol. 106, no. 4, pp. 620-630, 1957. (Online). Available: http://link.aps.org/doi/10.1103/PhysRev.106.620.
[14] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal, The,* vol. 27, no. 4, pp. 623-656, 1948, doi: 10.1002/j.1538-7305.1948.tb00917.x.
[15] D. De Jager, "UAV Benchmark mission 2," ed, 2019, p. Video of UAV Benchmark mission 2.

Deon de Jager graduated from Liverpool University with a Masters degree in Information Technology in 2007. He currently reading for a Ph.D in Cognitive Robotics and has a research interest in real-time cognitive processes. Current focus in research is on particle swarm intelligence for the real-time optimization of cognitive processes in cognitive robotics architectures. He is an experienced I.T. specialist, with a specialty in I.T. solutions architecture, application integration architecture, cloud solutions architecture and artificial intelligence.

Yahya Zweiri is currently Associate Professor with Kingston University London, Kingston, U.K. He received the Ph.D. degree from King's College London, London, U.K., in 2003 and was involved in defense and security research projects at Defense Science and Technology Laboratory (Dstl), King's College London, Field and Space Robotics Laboratory, Massachusetts Institute of Technology (MIT), USA and King Abdullah II Design and Development Bureau (KADDB), Jordan. His research focus is interaction dynamics between unmanned systems and unknown environments by means of deep learning, machine intelligence, constrained optimization and advanced control.

Dimitrios Makris is a Professor at the School of Computer Science and Mathematics, Kingston University. His research interests are in the area of Computer Vision, Machine Learning and Video and Motion Analysis. He is known for his research on learning scene semantic models, on multiple camera surveillance systems and on human motion analysis. He is currently the chair of the IET Vision and Imaging (V&I) Technical Professional Network and have been awarded numerous research grants as principal investigator.