# Hive Collective Intelligence for Cloud Robotics*

## A Hybrid Distributed Robotic Controller Design for Learning and Adaptation

A. Gkiokas[1)], E.G. Tsardoulias [2)], P.A. Mitkas[3)]

[1)] Ortelio Ltd, Coventry, CV1 2TT, UK
[2)] Centre of Research & Technology - Hellas
6th km Xarilaou - Thermi, 57001, Thessaloniki, Greece
[3)] Department of Electrical and Computer Engineering,
Aristotle University of Thessaloniki
54124 Thessaloniki, Greece

a.gkiokas@ortelio.co.uk, etsardou@iti.gr,
mitkas@auth.gr

**Abstract.** The recent advent of Cloud Computing, inevitably gave rise to Cloud Robotics. Whilst the field is arguably still in its infancy, great promise is shown regarding the problem of limited computational power in Robotics. This is the most evident advantage of Cloud Robotics, but, other much more significant yet subtle advantages can now be identified. Moving away from traditional Robotics, and approaching Cloud Robotics through the prism of distributed systems or Swarm Intelligence offers quite an interesting composure; physical robots deployed across different areas, may delegate tasks to higher intelligence agents residing in the cloud. This design has certain distinct attributes, similar with the organisation of a Hive or bee colony. Such a parallelism is crucial for the foundations set hereinafter, as they express through the hive design, a new scheme of distributed robotic architectures. Delegation of agent intelligence, from the physical robot swarms to the cloud controllers, creates a unique type of Hive Intelligence, where the controllers residing in the cloud, may act as the brain of a ubiquitous group of robots, whilst the robots themselves act as proxies for the Hive Intelligence. The sensors of the hive system providing the input and output are the robots, yet the information processing may take place collectively, individually or on a central hub, thus offering the advantages of a hybrid swarm and cloud controller. The realisation that radical robotic architectures can be created and implemented with current Artificial Intelligence models, raises interesting questions, such as if robots belonging to a hive, can perform tasks and procedures better or faster, and if can they learn through their interactions, and hence become more adaptive and intelligent.

---

* Parts of this work have been supported by the FP7 Collaborative Project RAPP (Grant Agreement No 610947), funded by the European Commission. We would also like to thank INRIA for the provision of the HOP framework.

adfa, p. 1, 2011.

**Keywords:** Robotics, Hive Intelligence, RAPP, Cloud robotics, Deep Boltzmann Networks, Neural Networks

## 1 Introduction

In this paper we formulate and examine a deep learning decision-making system, deployed through the Robotic Applications Platform, as a means to create a hive controller that will be able to learn from stimuli from different interactions, whilst outperforming traditional architectures. Robots, and especially humanoid robots, lack computational performance, an inherent hindering property of current central processors, that prohibits fast and complex algorithms such as computer vision, audio processing, knowledge management or learning. A variety of Artificial Intelligence (AI) models, such as Artificial Neural Networks attempted to address this problem, giving rise to a variety of other fields, such as Neuro-Robotics. Yet although those models provided advantages, due to the fact that modern day robots are still constrained by their processing capabilities, implementation of those models was limited and not as advantageous as expected.

And thus, whilst AI models and fields such as Deep Learning advanced, Robotics did not reap the benefits of those advancements. Out of this necessity for better performance, spawned the field of cloud robotics. Its premise is quite simple: delegate intensive and demanding algorithms to a powerful cloud via network connectivity, leaving low latency crucial tasks to be handled on-board the robot. That premise is soundly founded upon the biological counterpart: the human neocortex, which separates higher level cognitive functionality from lower level functions [1][2], such as somatosensory, proprioceptive or motor skills, which more often than not, are handled subconsciously, enabling higher level cognitive functions, such as learning or decision making to be performed consciously.

The usage of Cloud Robotics (CR) offers quite a lot of advantages, and only two major disadvantages: network latency and difficulty in dealing with dissimilar embodiment across a variety of robots. Hence, the need for differentiation of what algorithmic controllers may reside in the cloud, and which should remain on board the robot, are dictated by those disadvantages. Crucial, low latency operations that deal with the robot's balance or physical control should remain on-board, whilst more abstract, higher level cognitive tasks, not important for the robot's physical control and the safety of the people around it, may be delegated to the cloud.

Through examination of Moore's law with regards to the speculated performance of computational capabilities on robots, we can safely assume that we should not anticipate quantum leaps in the processing capability on robots any time soon. Whereas recent multi-core trends, as well as massively parallel algorithms have seen an increase from general purpose graphic processing compute ($GP^2U$), the energy consumption limitations and energy storage of robots, add more value to higher bandwidth and lower latency networking technology, rather than to mobile computing processors. The only exception to this general rule could be the advent of neural-based chips (duped as Neurochips) which are expected within the next 5 years. Due to all the aforementioned reasons, it is justified to invest time and research in distributed controllers that rely on networking, rather than to invest on singleton models and architectures that require increased computational performance on-board the robot.

# 2 Related Research

## 2.1 Cloud Robotics

Previous research in Cloud Robotics is somewhat limited. The most notable project is RoboEarth [3] which focused on an internet for robots and a knowledge sharing ontology for robots. Only after 2011 does research appear related to Cloud Robotics [4][5]. Albeit such a new field may appear revolutionary, it is based upon existing models and technologies, proven and thoroughly evaluated, many of which are imported from Computer Science and Artificial Intelligence. A wide array of existing AI models, previously used in robotics, such as Neuro-Robotics, Finite State Automata (FSM), Evolutionary Computation, Heuristic controllers and such, are all still usable and implementable, through a much more elastic, large and distributed platform. Because of that, previous research still relevant and related to the proposed Hive Intelligence model should be mentioned: Swarm Intelligence and Distributed Systems. Distributed systems as a Computer Science field are used for the purpose of the hive, and are hereinafter represented as the operation of Swarm Intelligence on robots and the cloud. Ant and Bee colonies, as sub-fields of Swarm intelligence have been researched, as artificial and natural systems that give rise to higher intelligence as a collective system of lower-lever intelligence.

On the contrary, the hybridisation between cloud, centralised and decentralised intelligent agents we propose, revolves around a ubiquitous disembodied intelligence based on the cloud, enabling the swarm. Our work, as part of the Robotic Applications Platform (RAPP) focuses on that hybridisation, described as Hive Intelligence, which aims to bridge complex and massively parallel algorithms for the purpose of learning, in ensemble with swarm intelligence. The accumulated collective interactions of the physical robots, aim to stimulate and teach the Queen, the agent residing on the cloud, to further enhance the collective intelligence, a concept never attempted before. With the exception of FSM and Heuristic controllers, Hive intelligence shares common components of Neuro-Robotics and Swarm Intelligence.

## 2.2 Neuro-Robotics

Neuro-Robotics as a sub-field combines Neuroscience, Robotics and AI, and has mostly focused on locomotion, motor control, learning and action selection. Research in the past two decades, examined biologically plausible mechanisms derived from the nervous system[6], behaviour-based robotics dependent upon stimuli [7], whereas more recent research investigated brain-centric architectures [8][9][10] emphasising a brain-like processing system. In a similar fashion, our design portrays a partial processing system, which behaves as the central brain of the hive, whilst at the same time, permitting localised processing centres in physical robots. The architecture of

Hive Intelligence does employ state-of-the-art Deep Networks, in order to enable the hive agent to learn efficiently and accurately, albeit not limited to it.

### 2.3 Swarm Intelligence

Swarm Intelligence, distinctly different from Neuro-Robotics, investigates the collective behaviour of decentralised self-organised artificial agents. The pioneers of the field, Gerardo Beni and Jing Wang, introduced the concept [11] in 1989. There exists a plethora of algorithms belonging to Swarm Intelligence, such as Particle Swarm Optimisation [12], Ant Colony Optimisation [13] and Artificial Bee Colony [14], amongst others.

In stark contrast to the research in swarm intelligence, we're not examining optimisation techniques through simulated swarms, but rather we propose to utilise a variety or robots, as part of a swarm, that *may or may not be* self-organising. This rather different approach, has been examined in the recent past [15][16], yet focused on the behaviour of the swarm. The title *Hive Intelligence* does not indicate any similarity with the Bee Colony algorithms, but with the organisation of a Bee Hive, in an effort to emphasise the key role of a central superior entity within the swarm.

Whilst Hive intelligence does not inhibit or prohibit collective behaviour or self-organisation, thereby collective behaviour is not the main focus. Providing a new type of non-biologically plausible artificial hive intelligence, focuses at enabling centralised surrogacy for intensive learning and memory, in order to create a more adaptive swarm.

## 3 The Hive

*Hive Intelligence* as a system is based upon the Robotics Application Platform (RAPP), which utilises an array of cloud-based services and robotic applications. Whereas traditional robotic controllers use heuristics or FSM for action-decision in order to execute a corresponding action or procedure, the hive design allows for greater flexibility, adaptation, learning and adjustment, mostly due to the fact that input received from robots is not tied to a specific action or decision, but may be interpretable as different actions or decisions, depending on previous training and the context.

Therefore, the hive is made up of the core-agent residing on the cloud, and the swarm agents, residing in robots. In that effect, the core-agent in the cloud is the queen, which is responsible for most action-decisions, through the execution of existing robotic applications, smaller controllers that perform a distinct or characteristic operation. However, to assume that the queen acts simply as a black box is erroneous, because the agent can utilise previous knowledge, ontological, visual or audible, as well as concepts derived from categorisation or clustering via multiple robot-users interactions, in order to evaluate and estimate what the output to a specific user input should be.

### 3.1 Input Preprocessing

In the most abstract sense, the queen acts as an i/o neural module, which is dependent on other algorithms to process its input, and other algorithms which will execute its output (equation 1).

$$q(f_n(x)) = g_k(y) \tag{1}$$

The input $x$ may be pre-processed by algorithms (function $f_n(x)$), and then propagated into the queen agent (function $q$). The output of the queen agent $q$, is the activation of a rapp or rapps $g_k$ with the parameters $y$. The input pre-processing is multidimensional and belongs to different domains, and we assume that the type of input provided by robot stimuli is known, and hence we can infer which functor should pre-process it.

If for example the input provided is audio (voice), then the speech-recognition functor $f_1$ should produce *partial* input to $q$. Similarly, if the input given is video, the computer-vision functor $f_2$ should produce a partial *input* to $q$. Therefore, the input to the queen agent is the encoded information produced as output from the sensor functors $f_n$. Whereas the state space of the input $x$ may be infinite, the output produced by the pre-processing functors $f_n$, is finite, since the pre-processing functors may be unable to process $x$, correctly classify it or group it.

Yet, in the event that neural-based functors, acting as approximators are employed, such as Restricted Boltzmann Machines [17] (RBM), and thereby producing probabilities of what an object is or what a sound means, represented as encoded input to $q$, we can hypothesize, that near-infinite or very large processing capabilities should arise, based upon a *brain-like* deep-network.

Encoding of the input provided to the visible layer of the RBM is a *sparsely encoded* representation of heterogeneous symbolic representations, in combination with non-symbolic metric data. We chose to do so, due to the relative advantageousness of sparse encoding when compared to localised or distributed encoding [18][19]. In addition to the aforementioned reason, distributed encoding of symbolic representations (such as words, object names or location titles) could possibly hinder the ability of the RBM to correctly classify input patterns.

### 3.2 Neural Architecture

The queen agent $q$, is a neural-based *Deep Belief Network* (DBN) implemented as a *Deep Boltzmann Machine* [20] with great *representational power*, whilst relying on *probabilistic and statistical* neural properties as derived from the Boltzmann architecture (fig. 1).

Furthermore, we chose to enhance the DBN that is the queen agent, by utilising similar characteristics of *Convolutional Neural Networks* [21] (CNN), the most notable being the interconnectivity of smaller sub-networks in order to extract subsets or features of the given input. In that sense, the agent queen is a hybrid system of two different *Deep Learning* designs, in order to be able to approximate, learn and identify important features and estimate which should be the best action-decision. However, we do not allow interconnectivity between Deep RBM as CNN do, but rather isolate stacks of Deep RBM, which are tasked with the propagation of a specific type of input. Only at the final Deep RBM, acting as the bridge between the other stacks, we allow information to be universally propagated.

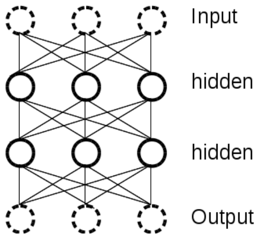


**Fig. 1.** Deep Boltzmann Network

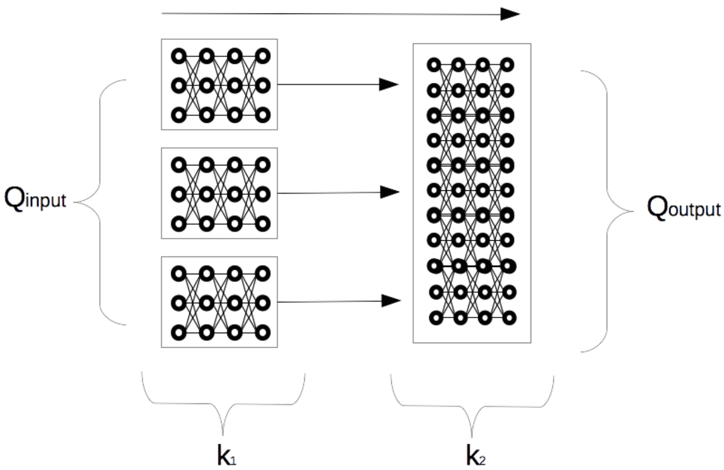


**Fig. 2.** Demonstration of Multi-Stack Deep Boltzmann Network

The overall design, uses the stacked or deep RBM model, whilst separating deep layers $k_1$. The input $Q_{input}$, as discussed in (3.3), may be symbolic or metric information, and as such, different *Deep RBM* handle each category, as shown in figure (2), in layer $k_1$. During feature extraction and subspace sampling in $k_1$, the categories across different stacks of different kinds of input, are *not* interconnected. The last layer $k_2$ interconnects the categorical Deep RBM, into one universal Deep RBM, that approximates for the given input across sets of Deep RBM, known patterns and their associated actions-decisions.

Another feature, studied in both biological and artificial systems, is the *small world characteristics* [22] with a smaller amount of neurons, connected by a much higher amount of synapses. As with Restricted Boltzmann Machines, no connections exist within the neurons of a layer, and the square boxes in figure (2) represent a Deep RBM. The stochastic and representative abilities of the RBM, in combination with their fast and easy training due to Contrastive Divergence and Gibbs sampling, proven to work well [17][23], provide sound reasons for employment of this architectural model.

The Neural Model uses Hinton's simple RBM training technique [24], as shown in equation (2).

$$E(u,h) = -\sum_{i\in visible} a_i u_i - \sum_{j\in hidden} b_i h_i - \sum_{i,j} u_i h_j w_{ij} \quad (2)$$

The Hopfield configuration $(v,h)$, where $v_i, h_i$ are the states of visible unit $i$ and hidden unit $j$, respectively. $a_i, b_j$ are their biases and $w_{ij}$ is the weight between them. The network assigns a probability by using the energy function, shown in equation (3).

$$p(u,h) = \frac{1}{Z} e^{-E(u,h)} \quad (3)$$

Where $Z$ is calculated by summing all visible and hidden pairs, shown in (4).

$$Z = \sum_{u,h} e^{-E(u,h)} \quad (4)$$

The learning rule, when simplified, is shown in (5).

$$\Delta w_{ij} = a(\langle u_i h_j \rangle_{data} - \langle u_i h_j \rangle_{model}) \quad (5)$$

Constant $a$ is the learning rate. The network is assumed to be initialised in a random state, and using Gibbs sampling, whereas updating uses Contrastive Divergence [23][24]. Models other than the Binary RBM as described by Hinton, include the possibility of using a Sparse RBM [25], or a Gaussian-Bernoulli RBM [26]. The learning rate may be adjustable (as mentioned in [24]) and the hidden to visible neuron ratio may also be adjusted, through trial and error, in order to fine-tune performance.

### 3.3 Encoding and Input Types

Per the Restricted Boltzmann Machine literature, all input will be min-max normalised between 0 and 1. Whereas a 0 could be assumed to be a false or absent encoding, and a 1 could be assumed to be a true or present encoding, real non-binary values may also be used, in order to encode arithmetic values. That distinction is important, and is the main driving reason behind the distinct categorisation of Deep RBM, as different categories, may have different domains and thus, different encodings. As already discussed in section (3.1), encoding is sparse, and in some cases it may be auto-encoded.

A synopsis of the input to the Queen mechanism is given in order to best portray the ambiguity of information processed by the system. The possible input types to the learning system, denoted as $Q_{types}$ are:

- $Q_{objects}$: Objects in view range detected. This information can be encoded as the object's ontology class, thus indexed by sparse encoding.

- $Q_{spatial}$: Description of the location of robot and users, such as *Kitchen, Living Room,...* etc. The possible locations will exist as classes in the ontology, also indexed via an absent or observable encoding.
- $Q_{temporal}$: Temporal information such as time of day, day of week, month etc. Integral type, min-max normalised.
- $Q_{sentiment}$: Sentiment analysis of the words (e.g. sentiment analysis pre-processing of the verbal user input). This input can be encoded with three classes: *Positive, Neutral, Negative*. Symbolic binary encoding is used in this scenario.
- $Q_{tokenst}$: A tokenised sentence, from speech-to-text pre-processing. This subset is sparsely encoded, where the number of inputs will be equal to the cardinality of the lexicon.
- $Q_{rapp}$: Recently or previously executed application sparsely encoded by a unique identifier.
- $Q_{cognitive}$: Cognitive condition based on pre-processed measurements, or hard-coded information. This encoding is both sparse and real-valued, as different types of cognitive values may represent psychological phenomena, such as anxiety, stress, depression, etc.

### 3.4 Output Decision System

The output of the Queen agent is actions or decisions. Yet the output of the Deep RBM model is an approximation of similarity to known patterns, e.g., known states. If we describe all the input attributes to the queen agent as a Markov state $s_i$, then an action $a_i$, is the choice to execute a RAPP on the robot or cloud. That action is not the output of the Deep RBM, but knowledge, discovered through random or heuristic searches and direct robot-user interactions. Thus, the association of an action $a_i$ to a specific state $s_i$, also describes the relation of the $Q_{input}$ to $Q_{output}$.

Whilst the Deep RBM serves as an approximation for similar input, actions are not discovered or necessarily tied to a specific action or output. Typically, such approximations perform equally well to heuristic controllers or FSM, but their advantages are performance (especially during *reconstruction* in RBM), but most importantly, feature extraction and partial reconstruction, hence, plasticity in decision-making and adaptation to new or unforeseen states and events.

Whereas an FSM or Heuristic controller need to be explicitly programmed or have defined output or actions for a specific input, the proposed design can deal, in a probabilistic manner, with unknown input-states, partial input-states or noisy input-states, and thus approximate with minimal algorithmic searches, the best *known course of action*.

Another interesting aspect of the suggested architecture it that the queen agent can produce many different outputs by approximating input to output, in a probabilistic manner. Therefore, the hive contains a RAPP suggestion mechanism, regardless of previous experience of an individual robot or the robot swarm. Furthermore, the hive is able to be trained on heterogeneous problems, and through feature extraction and classification, detect alarming situations concerning the user. In such a scenario, aside from the action, as described by a RAPP suggestion or execution, the hive can alert medical personnel or caregivers, or even perform a rudimentary form of medical prediction via previous states experienced by other robots of the swarm, even if that particular robot or user has never experienced this state.

The actual $Q_{output}$ is a vector containing the metric similarity or the approximation of similarity of the input, to an already experienced pattern used during training. As such, in the event that a new input during *reconstruction*, matches with high probability a dangerous or worrying condition, an alert may be issued by the hive. The normal operation is to execute the best scoring RAPP as either suggested by a heuristic controller or an FSM, or as discovered via other means, such as Reinforcement learning [27] and $\epsilon$-greedy exploratory policies, which are outside the scope of this paper.

# 4 Hive Training and Learning

## 4.1 Information and Knowledge Management

A basic node of the RAPP system, where the hive Queen algorithms will be executed, is *RIC*, the RAPP Improvement Centre. This information scheme resides in the cloud and consists of the following subsystems.

Firstly, there is an ontology component, created by the fusion of the KnowRob robotic ontology (RoboEarth project) and the OpenAAL ontology, which contains concepts relative to assisted or independent living. This component will be responsible for storing different concepts (ontology classes), their semantic relations, as well as aggregated instances of these classes.

Furthermore, several general purpose services that provide access to heavy duty algorithms such as *Speech-to-Text* audio processing, *Computer Vision* algorithms and others. These services can be called from any RApp and any information captured or generated by them will be stored in the RAPP cloud, and thus be available to the Hive intelligence. The information warehousing will follow a top-down encapsulated scheme, where data will be stored per RApp and per User.

Other data stored and manipulated either by RAPP services, Rapps, or the hive itself, is raw data, or inferred data, such as statistical information and calculated arising probabilities.

## 4.2 Robot – User Interaction

The actual training, employing the RIC data can be unsupervised, semi-supervised or even supervised through interaction. Unsupervised training may be deployed as feature extraction and clustering procedures, or even as HMMs (Hidden Markov Models) [28] for predictions concerning anticipated user actions and interactions in the User-Robot-Environment domain. A supervised approach includes direct input from the user, which indicates if the output was the desired one or in general terms acceptable. This is possible via robot-user vocal communication, where the robot obtains affirmation that the an action $a_i$ is indeed the wanted output for an input state

$s_i$, as discussed in (3.4). This is a form of direct and supervised training, as the information collected during such an interaction, becomes in essence the data used to train the hive intelligence.

A paradigm of this approach may be, information based on user-generated responses and non-intrusive observations from compliance of the end user. In the first scenario, the *robot initiated* communication, obtains a response, which when processed by the hive agent, produces a set of viable Rapps that could be executed. In the latter scenario, an action (described by the execution of a Rapp) may be assumed to be correct, unless the user objects to the robot about its course of action.

Seamlessly switching between both scenarios, creating a fusion of *supervised* and *semi-supervised* training, with resemblance to physical learning and problem-solving, such as the way the human mind learns by collecting unlabeled data [29]. Learning from interactive experience, may be a radical new approach to training robots, which will enable knowledge and experience re-use in the robot swarm, through the hive's collective memory.

### 4.3 Training Procedures

Through the *robot-user* interaction as described in (4.2), a variety of symbolic and metric data is collected, whilst the queen agent has not been trained. During this pre-training phase, the Hive is virtually a passive system, which only observes the interactions and stores data in a shared memory, employed via the RAPP Platform. At this phase, only FSM or heuristic controllers (cloud and robot based) are deployed.

When deemed that enough data has been collected, from the robot-swam, via a variety of interactions between different users and robots, aggregated and encoded data, that has already been pre-processed (see 3.1), is then trained into the Deep RBM in one single batch, after being sparsely encoded (see 3.3). The indexing and encoding procedure is not part of the training process, but rather a part of the queen agent and the RIC. Doing a single training session, rather than mini-batches, is based upon the decision to contain two identical Deep RBM in the agent's memory, each mirroring one another. Whilst a one-time training could take longer than mini-batch training, a mirroring system, will allow one RBM to function, while its copy can be updated by being trained with new data.

In order to reach the decision point, where one of the Deep RBM has to be re-trained, a statistic variance of new information, must be calculated which demonstrates that enough new data has been recorded that is *very different* from currently existing data in the agent's memory. Whilst current implantation of the Deep RBM is based upon a CPU version, future versions based on $GP^2U$ compute, should provide a significant reduction in training time, as experienced in recent research [30].

## 5 Work In Progress

The described platform is currently work in progress. Parts of the RAPP platform are being actively developed, as is part of the Hive system. The RAPP platform is expected be complete within 2015, the Hive system, not being a standalone system, but rather relying on the rest of the RAPP platform, will also be complete within 2015. Other machine learning models, that can be used as inference or learning mechanisms are also being examined, but the core of the Deep RBM is already being evaluated.

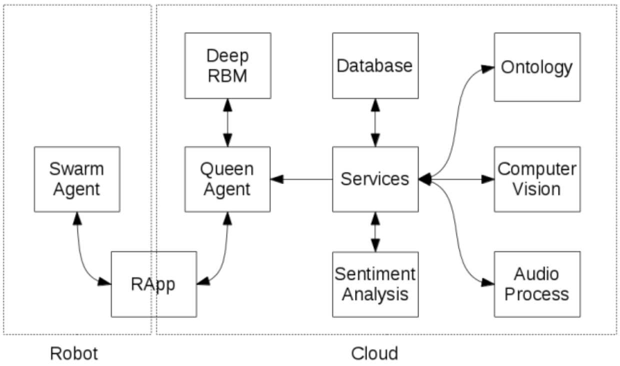


**Fig. 3.** Blueprint for the Hive

The work so far was oriented in the full specification of the overall RAPP System architecture (figure 3). As indicated, the RAPP System is divided in the Robot and the Cloud part. The Cloud part, denoted as *RAPP Platform* is comprised of the RApp Store which is indexes all available RAPPS and distributes them to the robots, the RIC which encapsulates the DB, the Ontology, the Services and the Hive Intelligence, and the dynamically allocated Cloud agents which are distributed asynchronous RApps executing concurrently with their robot counter-part. The Robot counter-part consists of a Core agent that controls the communication with the cloud and provides robot-specific resources to the applications, and the dynamic agents which are the robot-part of a RApp, and the communication layer that is responsible for all module interactions in the robot realm.

The consortium's next step is the actual implementation of this system, followed by its actual deployment in elders suffering from mild cognitive impairment or going through hip fracture rehabilitation. The robots to be used are Aldebaran's NAO, a humanoid social robot, and ANG-Med, a robotic walking aid, manufactured by the INRIA science and technology institution.

## 6 Conclusions

In order to advance the research in the field of cloud-robotics beyond the current state of the art, we've described a hybrid system, which fuses cloud robotics, with Deep-Learning. This combination tackles computational limitations onboard robotics, knowledge and experience sharing amongst robot swarms, and highly stimulates a

Deep-Learning-based robotic controller, which is used to process high-level decisions delegated by individual robots.

We have argued and described how such a system is much more elastic and adaptive to robot-user interactions, and how it benefits from interactive learning, either by direct vocal communication with the users, or by indirect observations. Furhermore, in order to take a holistic approach with regards to the learning of the hive intelligent agent, information used to train the hive agent, is obtained across different domains, such as Computer Vision, Knowledge Representation, Speech Recognition, Sentiment Analysis and so on. Due to the heterogenous nature of the information used, we've described a Deep Learning model, based upon Deep RBM, but with certain biological characteristics. The main hypothesis of this neural design, is the ability to correctly classify and approximate heterogenous information provided to the hive intelligence, and extract features which are important, and upon which the agent will base its decisions.

Finally, the described system finds a fruitful application field in the RAPP project, which delivers a Robotic Application store combined with a collective knowledge representation and a centralised memory storage. It is unquestionable that as the trends indicate, the future of robotics lies in distributed cloud execution and centralised inference mechanisms that gather and produce knowledge in a machine friendly form, allowing robots to form their own digital communities. As such, the way we currently perceive robot software and their controllers, may be changing towards directions which borrow from biological systems, but enable new unique artificial intelligence blueprints, as hybrids of collective and centralised cybernetic controllers.

## References


1. Halsband U., Lange R. K. *Motor learning in man: a review of functional and clinical studies*, Journal of Physiology, Paris, Volume 4, Issue 6, pp. 414-424, 2006.
2. Haycock, D. E. *Being and Perceiving*, Manupod Press, July 31, 2011, ISBN-10: 0956962106.
3. Waibel M. et al, *RoboEarth - A World Wide Web for Robots*, Robotics and Automation Magazine, IEEE, Volume 18, issue 2, pp. 69-82, June 2011.
4. Kamei K., Nishio S., Hagita N., and Sato M. *Cloud Networked Robotics*, IEEE Network, Volume 26, issue 3, pp. 28-34, June 2012.
5. Quintas J. M., Menezes P. J., and Dias J. M. *Cloud Robotics: Toward Contect Aware Robotic Networks*, in Proceedings of IASTED, The 16th IASTED International Conference on Robotics (Robo 2011), Pittsburgh, USA, November 7-9, 2011
6. Chiel H. J. and Beer R. D. *The brain has a body: adaptive behavior emerges from interactions of nervous system, body and environment*, Trends in Neurosciences, Volume 12, issue 20, pp. 553-557, 1997.
7. Mataric M. J. *Behavior-based robotics as a tool for synthesis of artificial behavior and analysis of natural behavior*, Trends in Cognitive Sciences, Volume 3, issue 2, pp. 82-87, 1998.
8. Warwick K. *Implications and consequences of robots with biological brains*, Ethics and Information Technology, Volume 12, issue 3, pp. 223-234, 2010

9. Cox B. R. and Krichmar J. L. *Neuromodulation as a Robot Controller A Brain-Inspired Strategy for Controlling Autonomous Robots*, Ieee Robotics & Automation Magazine, Volume 16, issue 3, pp. 72-80, 2009.
10. Rucci M., Bullock D. and Santini F. *Integrating robotics and neuroscience: brains for robots, bodies for brains* Advanced Robotics, Volume 21 issue. 10, p.p. 1115-1129, 2007.
11. Beni G. and Wang J. *Swarm Intelligence in Cellular Robotic Systems*, Advanced Workshop on Robots and Biological Systems, Volume 102, pp. 703-712, 1993.
12. Clerc M. *Particle Swarm Optimisation*, France Telecom, France, 2006, SBN: 9781905209040
13. Dorigo M. and Stutzle T. *Ant Colony Optimization*, MIT Press, 2004, ISBN 0-262-04219-3
14. Pham D.T. and Castellani M. *The Bees Algorithm - Modelling Foraging Behaviour to Solve Continuous Optimisation Problems*, Journal of MEchanical Engineering Science, Part C, Volume 223, issue 12, pp. 2919-2938, 2009.
15. McLurkin J. and Yamins D. *Dynamic Task Assignment in Robot Swarms*, Proceedings of Robotics: Science and Systems, pp. 129-136, 2005.
16. Winfield A.F.T. and Nembrini J. *Safety in numbers: fault-tolerance in robot swarms,* International Journal of Modelling, Identification and Control, Volume. 1, issue 1, pp. 30-37, 2006
17. Hinton G. *A Practical Guide to Training Restricted Boltzmann Machines, Neural Networks: Tricks of the Trade*, Second Edition, Lecture Notes in Computer Science, Springer, 2014, pp. 599-619, ISBN: 978-3-642-35289-8.
18. Rolls, T. E. and Treves A. *The relative advantages of sparse versus distributed encoding for associative neuronal networks in the brain*, Journal Network, Volume 1, issue 4, pp. 407-421, 1990.
19. Wixted, T. J. et al, *Sparse and distributed coding of episodic memory in neurons of the human hippocampus,* Proceedings of the National Academy of Sciences, USA, Volume 111, issue 26, pp. 9621-9626, 2014
20. Le Roux N. and Yoshua B. *Representational Power of Restricted Boltzmann Machines and Deep Belief Networks*, Neural Computation, Volume. 20, issue 6, pp. 1631-1649, June 2008.
21. Ciresan D., Meier U., Masci J., Gambardella L. M., Schmidhuber J. *Flexible, High Performance Convolutional Neural Networks for Image Classification*, Proceedings of the Twenty-Second international joint conference on Artificial Intelligence, Volume. 2, pp. 1237-1242, 2011
22. Sporns O. and Zwi JD. *The small world of the cerebral cortex,* Neuroinformatics, Volume 2, Issue 2 , pp 145-162, 2004.
23. Hinton G. *Learning multiple layers of representation*, Trends in Cognitive Sciences, Volume 11, pp. 428-434, 2007
24. Hinton G. *A Practical Guide to Training Restricted Boltzmann Machines, Neural Networks: Tricks of the Trade*, Second Edition, Lecture Notes in Computer Science, Springer, 2014, pp. 599-619, ISBN: 978-3-642-35289-8.
25. Ekanadham C. *Sparse deep belief net model for visual area V2*, Advances in Neural Information Processing Systems, Volume 20, Curran Associates, 2008.
26. Cho K., Ilin A. and Raiko T. *Improved Learning of Gaussian-Bernoulli Restricted Boltzmann Machines*, Artificial Neural Networks and Machine Learning - ICANN 2011, Volume 6791, pp. 10-17, 2011.
27. Sutton R. S. and Barto A. G. *Reinforcement Learning: An Introduction*, MIT press, 1998, ISBN-10: 9780262193986

28. Baum, L. E. and Petrie T. *Statistical inference for probabilistic functions of finite state Markov chains*, The Annals of Mathematical Statistics, Volume 37, issue 6, pp. 1554-1563, December 1966.
29. Younger B. A. and Fearing D. D. *Parsing items into separate categories: Developmental change in infant categorization,* Child Development, Volume 70, issue 2, pp. 291-303, 1999.
30. Raina R., Madhavan A., and Ng A. Y. *Large-scale Deep Unsupervised Learning Using Graphics Processors*, Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09, Montreal, Quebec, Canada, pp. 873 - 880, 2009