Development of a Cloud-based Computational Framework for an Empathetic Robot

Syed Moshfeq Salaken
Deakin University
syed.salaken@deakin.edu.au

Saeid Nahavandi
Deakin University
saeid.nahavandi@deakin.edu.au

Conor McGinn
Trinity College Dublin
mcginnc@tcd.ie

Mohammed Hossny
Deakin University
mohammed.hossny@deakin.edu.au

Kevin Kelly
Trinity College Dublin
kekelly@tcd.ie

Ahmed Abobakr
Deakin University
a.abobakr@deakin.edu.au

Darius Nahavandi
Deakin University
darius.nahavandi@deakin.edu.au

Julie Iskander
Deakin University
j.iskanderistafanos@deakin.edu.au

ABSTRACT

This article presents the development and preliminary evaluation of an empathy controlled robot. Such a robot is one step forward towards industry 5.0, as it provides a theoretical framework to enable the performance of the robot to be customized to suit the needs of both the task as well as the operator. An inventive step is taken through the separation of computational resources based on whether the algorithms are addressing functional or experiential needs. The paper therefore addresses the requirement for new approaches that can be employed in the design of mobile robots to reduce cost, power consumption, and computational burden of the system. We propose that tasks requiring real-time and safety critical control are processed using dedicated on-board computers, whereas functionality dedicated to system optimization, machine learning and customization are handled through use a cloud-based platform. In this paper, key components of the architecture are defined, and the development and preliminary evaluation of an exemplar robot capable of changing its behavior in accordance with the perceived emotional state of an operator's voice is presented.

CCS Concepts

• **Computing methodologies~Evolutionary robotics**

Keywords

emotion classification, robot, deep learning, cloud control, intent perception

ICCAE 2019, February 23–25, 2019, Perth, WN, Australia

© 2019 Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM 978-1-4503-6287-0/19/02…$15.00

https://doi.org/10.1145/3313991.3314018

1. INTRODUCTION

Due to the advancement in mobile cellular technology, wi-fi and other forms for Internet connectivity, we are now living in a connected world. A push towards Internet of Things (IoT) is also making Internet connectivity ubiquitous for otherwise passive, non-connected devices such as light globes, garage door controller, air conditioner etc [1]–[3]. Over the past few years, we have seen a number of robotic applications utilizing connectivity of some form to deliver more value towards the end-user [4]–[9]. Therefore, it is logical to expect that the robotics world will soon embrace the era of connectivity and move towards utilizing the positives of cloud technologies. We argue that these will happen soon, as a 'cloud-first' principle delivers the following benefits over a traditional design:

•Cloud robotics can improve maintainability and feature addition capabilities through reduction in the downtime to perform software upgrades. Using a cloud infrastructure, changes to software can be made remotely and then brought online almost instantaneously. This overcomes significant challenges with existing robots as performing these functions requires the robot to be taken offline

while wireless software updates are being performed. The problem is more severe for robots that do not have the capability for wireless updates, as physical disassembly is required

•Autonomous mobile robots often require expensive computers, since their applications demand high levels of processing, a small form factor, low power and environ- mental protection. Using a cloud robotics approach en- ables robots to offset many of the computational functions to remote servers that have significantly lower operating costs, thus enabling cheaper computer hardware to be used on the robot.

•Cloud infrastructure makes it possible to vary (i.e. in- crease or decrease) the computational resources dele- gated to individual robots. This facilitates better 'future-proofing' of robots since physical hardware does not need to be replaced/upgraded on robots to enable improved performance.

•Using the cloud to process and store data can have tangible data security benefits, since data is generally not stored locally on the robot and can be easily backed up across multiple servers. Also, using the cloud minimizes risks caused by on-board hardware failures, since processing in the cloud can allow for higher levels of redundancy.

•Cloud robotics facilitates an efficient means of data aggregation, which may have significant benefits in distributed robot tasks (i.e. optimal scheduling of mobile robots). Furthermore, efficient data collection can improve the performance of machine learning models, since they can benefit from bigger dataset than is normally possible using robots in isolation.

•While computers used on robots tend not to have the highest energy consumption, using them continuously (especially when GPUs are involved) can significantly effect the robots operational life. For example, the Care- O-Bot 4 [10] robot can be configured with 4-6 Intel Nuc small form factor PCs that each consume more than 30W during use [11].

•Deploying cloud applications embraces the modular de- sign philosophy; it breaks down traditional monolithic code-base and encourages micro-service based design methods. In turn, this makes the code-base more manageable and may make bug identification easier. Using standardized API calls (i.e. REST API), it can also allow easier integration of third party on the robot.

In addition to providing some compelling benefits, cloud services also have several notable limitations. First, the time it takes to perform off-board computation can introduce latency that may problematic for use in near real-time applications and performance may be dependent on external factors (i.e. quality of 4G signal, location in a building). Also, since many aspects of the robots' performance is dependent on cloud processes, and failure in the network can result in downtime and dramatic change in performance capability in one or a fleet of robots. This may reduce operational efficiency, but may also have adverse effects on the robot operators who may observe an unexplained mismatch in observed performance and expected performance levels.

The main contribution of this paper is to propose a frame- work for leveraging the benefits of cloud-based architecture with the stability and reliability of a local on-board computers. Our research works as a proof-of-concept, intended to demon- strate how a low-cost robot can be configured in a manner that ensures high functional reliability, while also allowing for advanced functionality normally reserved for high-spec robots that have access to significant on-board processing resources. The rest of the paper is organized as follows: section-II describes the cloud architecture and information flow experiment, Section-III describes the methodology used in the testing including the deep learning method used in emotion classification, Section-V shows the characterizes the proposed system in terms of network latency, Section-V talks about the improvements scope and finally, Section-VI concludes the paper

2. SYSTEM ARCHITECTURE

2.1 Theoretical framework

The emergent behaviour of an autonomous robot is dependent on the interaction between its reactive and deliberative control systems [12]. Reactive control systems are characterised by the direct coupling of perception and actuation, whereas deliberative control systems explicitly incorporate some degree of planning, which is normally based on an internal model/representation that the robot has of its environment [12]. Research over the past decade has found that the highest performing robots utilize hybrid architectures that comprise both deliberative and reactive elements [13]–[15]. For mechanically simple robots, reactive control involves relatively low data bandwidth which can be handled using microprocessors, DSPs and other types of conventional embedded hardware. For more complex robots, such as those required to perform 3D motion planning (bipeds, drones, etc.), it has been common to use FPGA [16] or other more bespoke types of computational hardware [17].

Over the past decade, innovations in robotic perception, driven by the development of probabilistic methods [18], cloud robotics [19] and more recently deep learning [20] has led to significantly improved levels of autonomy on robots. While deploying these methods on robots does not usually require 'hard' real time performance, it does demand large processing bandwidth owing to the amount of data processed by sensors (i.e. camera feeds, point cloud fields) as well as the often high computational requirements of the underlying algorithms (i.e. CNNs, template matching, SLAM). Owing to high system complexity of implementing state-of-the-art algorithms on real robots, it has become the standard for engineers to develop and deploy using the Robot Operating System (ROS) [21]. ROS is a type of middleware, normally run on a Linux operating system, that provides hardware abstraction, low- level device control, implementation of commonly used functionality, message-passing between processes, and package management. While using ROS has many benefits (code reuse, hardware abstraction, debugging functionality, etc.), using it comes at a cost, namely a reduction in overall computational performance.

It is proposed that some of the computational limitations of ROS may be overcome using a computational architecture that can connect to the cloud. This can be done by creating communication interfaces between the robot and the cloud, by using ROS (i.e. through creation of a ROS slave) or utilization of a REST API which permits a ROS program to exchange information with a 3rd party cloud service. To protect against problems that may arise in the event of a network failure, it is proposed that on-board computers are used to perform all reactive control elements, as is the ROS master which possesses the requisite autonomy level to take the appropriate actions in the event of a network failure. An illustration of the key components of the proposed architecture is given in figure 1.

2.2 Implementation on a physical robot platform

An exemplar of the proposed framework was developed using an off-the-shelf Turtlebot3 robot (Burger model) [22]. The Turtlebot3 is a small, differentially driven robot developed for education and research purposes. The robot is controlled natively in ROS using an OpenCR low-level controller along with a Raspberry Pi3. The robot possesses a range of on- board sensors including a LIDAR module, camera and USB microphone with required digital circuitry.

A ROS program was written to enable the robot to use the Google Cloud Speech engine [23] to convert speech-to-text. Google Cloud Speech is considered a state-of-the-art speech- to-text engine, which was trained using advanced techniques in deep learning. The program makes use of the openly available REST API provided by Google.

A small dedicated cloud server was also setup. This was implemented using a Amazon Elastic Cloud Computing node with 12 GB storage and 512 MB of RAM hosted in Sydney region. This server was configured to store CNN models and serve prediction on the robot. Since the server is only serving one model over HTTP/CURL with no other tasks to be done (not even an UI for the API), the workload is minimal.

However, in case the requests rate increases, the RAM can be upgraded easily through a few mouse clicks or the cloud server be placed behind a load balancer with more of these smaller machines configured in the same way.

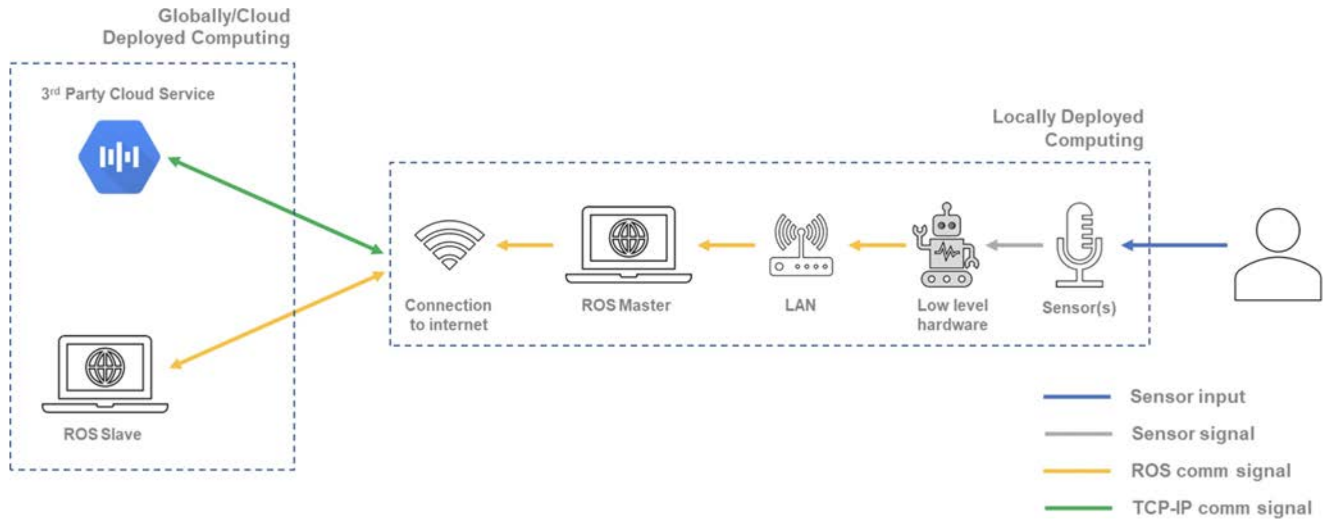

Figure 1. The Framework.

3. TESTING METHODOLOGY

A simple experiment was conceived which permitted for the adjustment of robot behaviour based on the perceived emotion of a human operator. The robot was programmed to drive forwards and backwards. These capabilities were deployed using on-board local computers and succeed in enabling the robot to perform its base functionality. More complex behaviour was made possible through addition of a microphone input, which allowed a human controller to speak to the robot. The human controller speaks to the robot in plain, simple and short sentences. The robot then passed the captured voice to a ROS master node which then checks if the voice is intelligible. If not, the robot is instructed to listen again. Otherwise, the ROS master node sends the voice data to Google cloud speech API which then returns a text corresponding to the voice. Google cloud speech API returns multiple alternatives for the transcription, however, in this experiment, only the most probable alternative is used. Once the transcription is received, passes the data to the dedicated cloud server through a REST API, where a program to detect emotion based on the transcribed text is run. The process flow of the experiment is illustrated in figure 2.

When the transcribed text is received, the ROS master node sent the text form of the sentence (via REST API) to the dedicated cloud server where emotion classification was performed using a model trained from a CNN. Once the emotion server responds with an identified emotion, the robot checks if the emotion has positive (i.e. joy, love or surprise) or negative (i.e. anger, fear, sadness) valence. The direction of travel of the robot is then adjusted based on the emotional valence (positive – towards operator, negative – away from operator).

3.1 Dataset for training emotion classification

The dataset used in this work is a emotion classification dataset published on Kaggle by Eray Yildiz [24]. It contains 416, 809 records and 2 columns, text and emotions. The text field contains a sentence and emotions contain a label for corresponding sentence. The dataset is imbalanced in the sense that joy and sadness covers nearly 62.9% of all records (see Fig-3 for details).

4. RESULTS

4.1 Deep Learning Model

The emotion classifier used in this experiment is a three- layer convolutional neural network. The first layer is an embedding layer that handles the embedding of word in the vocabulary, second layer is a convolutional layer with rectified linear unit activation, third layer contains a max-pooling layer which is then followed by one dense layer. The output layer classifies the network output in 6 emotion classes. The network is trained to minimize cross-entropy (as defined in equation-1) using 'Adam' optimizer [25].

$$E = -\sum_{x} p(x) \log q(x) \tag{1}$$

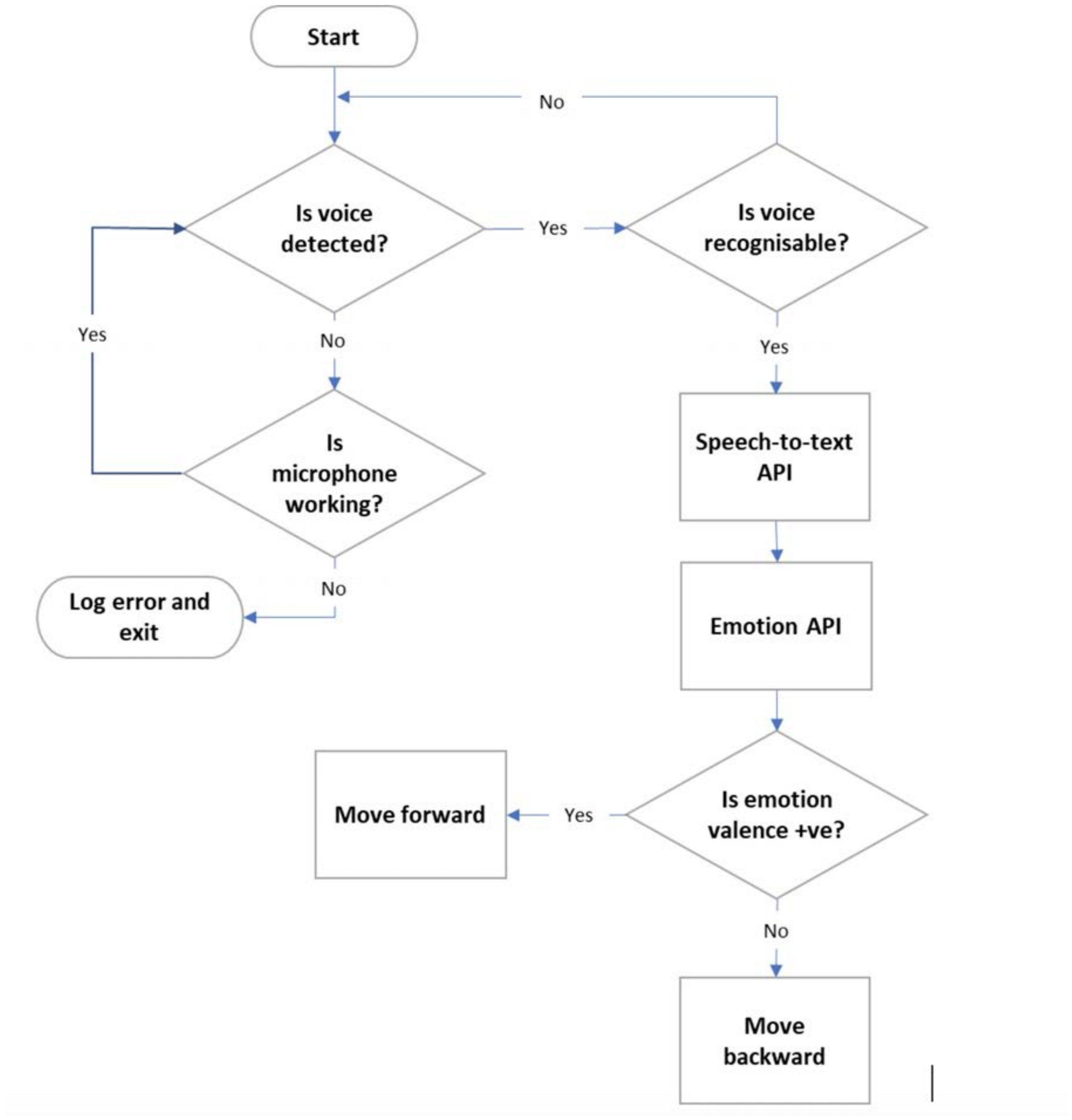

Figure 2. Process flow chart for emotion driven robot.

Best parameters for the model are chosen using a randomized grid search with the following parameters search space:

- Number of filters: 32, 64, 128
- Kernel size: 3, 5, 7

The best model achieved 92.5% accuracy on a held-out test set with the following parameters:

- Number of filters: 64
- Kernel size: 5
- Embedding dimension: 50

Other training and network parameters are shown in table-I.

TABLE 1. PARAMETERS FOR CNN AND TRAINING

Parameter	Value
Mini-batch size	10
Maximum length of sequence	100
Padding Mode	post
Dense embedding dimension	50
Vocabulary size	65791
Training epochs	20
Number of parameter settings to be sampled	5
Learning rate	0.001
β_1 / Momentum	0.9

We did not optimize for embedding dimensions and other relevant parameters in the experiment. This has been reserved for the future works.

4.2 Latency

Tests were performed to test the latency of the respective cloud services. Network latency was identified in section I as one of the biggest possible limitations of cloud applications.

TABLE 2.LATENCY FROM TESTS USING CLOUD APIS BASED ON 20 TRIALS

	Mean Response Time (seconds)	Std. Dev
Transcription API	1.91	0.57
Emotion Detection API	0.11	0.01

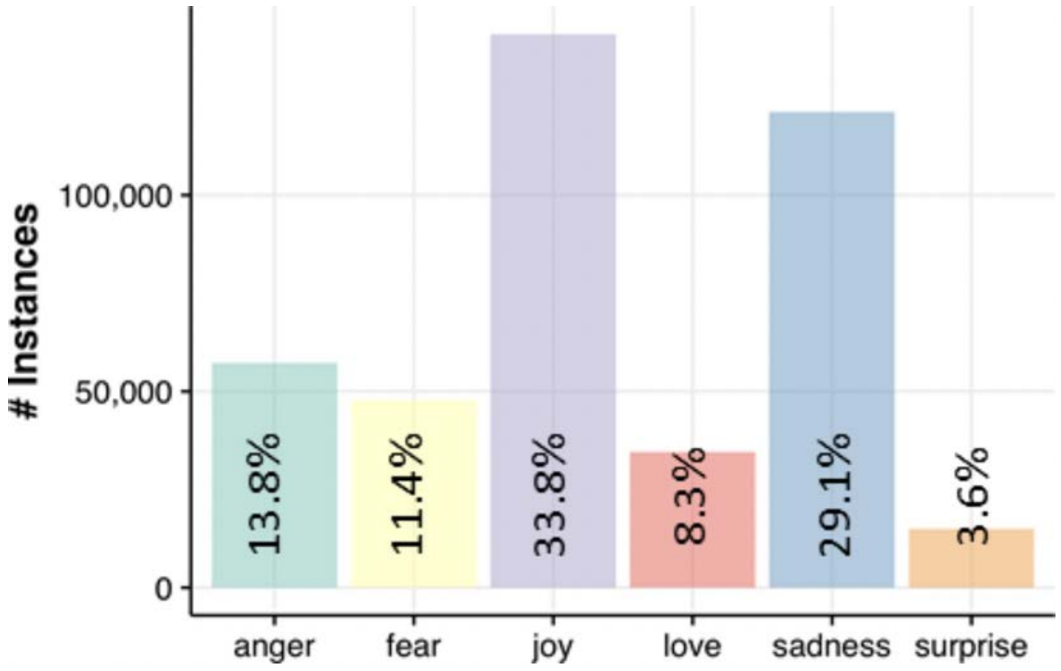

Figure 3. Frequency of emotional labels in for dataset.

Table II shows the mean response time for main components of that decision making process. It was found that the latency was significantly higher for the transcription tests than for the emotion detection tests. This difference may be attributable to the fact that the transcription algorithm would likely require greater levels of computational processing, given the signal-to- noise ratio is large for speech. Also, it is possible that since the transcription API used a free version of Google Speech API [16], it may have been the case that some of the delay was attributable to throttle requests coming through this free version. These issues were largely avoided when using the dedicated server, since the model was likely less complex and only served requests from one robot.

5. FUTURE WORKS

The paper reports a relatively simple use-case that was conceived to serve as a proof-of-concept of the proposed framework. Future work will initially seek to improve the performance of existing elements of the current system. This may be done by broadening the parameter search space in the optimization step and by exploring the performance of other parameters optimization methods such as genetic algorithms [26]. Other learning techniques, such as transfer learning [27], [28], fuzzy learning [29] may also be used to improve the robot's behavior as well as collecting new data points in regards to brain activity [30] and posture of the operator [31]. A second direction of future research will be deploy methods that can more accurately predict the emotional state of the operator, namely through triangulation process involving a number of different sentiment analysis techniques based on facial expression, tone of voice, brain activity, physiological state, etc.

Building on the initial success of this proof of concept, it is our intention to demonstrate the framework's utility on a real robot use-case. Possible examples may include: a cobot performing a collaborative task in a factory environment, a social robot engaging in dialogue-based social interaction with a user, or mobile manipulator being teleoperated by a remote operator.

Finally, it would also be interesting to conduct a study to compare the practical performance of cloud-based services that may have practical interest to robotics developers. Over the course of this research, it became apparent that while many such services exist from companies like IBM [32], Google [33], Amazon [34], Nvidia [35], Microsoft [36], and others, there have been few independently conducted tests to investigate the performance of the respective systems.

6. CONCLUSION

This paper proposed a framework for developing robot con- trol systems capable of adapting to both real-time functional performance requirements, as well as to near real-time hedonic requirements. The paper reported preliminary results from a robot prototype that allowed a human operator to augment robot performance based on the sentiment of speech content. This is a step forward towards intent controlled robots which, in near future, may work as companion for human in regular tasks. We have shown how the proposed framework utilizes cloud technologies and remove dependencies on expensive computing platforms for mobile robots, introduces ease of maintenance, establishes central control and allows for easy feature upgrades in the field of robotics. We believe this work also advances the field of cooperative robotics by using affective computing to control different forms of robots. Furthermore, emotion detection methods, speech recognition techniques and system characterization has been provided as part of this application paper.

7. ACKNOWLEDGMENT

This research is supported by iTrust Lab at Institute of Intelligent Systems Research and Innovation (IISRI) at Deakin University, Australia.

8. REFERENCES

[1] S. Meyer, A. Ruppen, and C. Magerkurth, "Internet of things-aware pro- cess modeling: integrating iot devices as business process resources," in International conference on advanced information systems engineering, pp. 84–98, Springer, 2013.

[2] B. Martinez, M. Monton, I. Vilajosana, and J. D. Prades, "The power of models: Modeling power consumption for iot devices," IEEE Sensors Journal, vol. 15, no. 10, pp. 5777–5789, 2015.

[3] S. D. T. Kelly, N. K. Suryadevara, and S. C. Mukhopadhyay, "Towards the implementation of iot for environmental condition monitoring in homes," IEEE Sensors Journal, vol. 13, no. 10, pp. 3846–3853, 2013.

[4] S. E. Salcudean, W. H. Zhu, P. Abolmaesumi, S. Bachmann, and

[5] P. D. Lawrence, "A robot system for medical ultrasound," in Robotics Research, pp. 195–202, Springer, 2000.

[6] A. Guerraz, B. Hennion, P. Thorel, F. Pellissier, A. Vienne, and I. Bel- ghit, "A haptic command station for remote ultrasound examinations," in haptics, p. 88, IEEE, 2002.

[7] Y. Wang, C. S. Jordan, M. Pinter, and M. C. Chan, "Remote presence display through remotely controlled robot," July 20 2010. US Patent 7,761,185.

[8] R. C. Luo, K. L. Su, S. H. Shen, and K. H. Tsai, "Networked intelligent robots through the internet: issues and opportunities," Proceedings of the IEEE, vol. 91, no. 3, pp. 371–382, 2003.

[9] S. Nahavandi, Z. Najdovski, B. Horan, and A. Bhatti, "Method and apparatus for haptic control," Nov. 3 2015. US Patent 9,174,344.

[10] S. Nahavandi, Z. Najdovski, B. Horan, and A. Bhatti, "Universal motion simulator," Apr. 7 2016. US Patent App. 14/872,177.

[11] N. Board, "Intel product compatibility tool." http://compatibleproducts.intel.com/ ProductDetails?activeModule=Intel

[12] A. Cunningham, "Review: Much-improved iris gpu makes the skylake nuc a major upgrade." https://arstechnica.com/gadgets/2016/03/ review- much-improved-iris-gpu-makes-the-skylake-nuc-a-major-upgrade/, March 2016.

[13] R. C. Arkin, R. C. Arkin, et al., Behavior-based robotics. MIT press, 1998.

[14] C. McGinn, M. F. Cullinan, G. Walsh, C. Donavan, and K. Kelly, "Towards an embodied system-level architecture for mobile robots," in Advanced Robotics (ICAR), 2015 International Conference on, pp. 536– 542, IEEE, 2015.

[15] R. C. Arkin, "Governing lethal behavior: embedding ethics in a hy- brid deliberative/reactive robot architecture," in Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction, pp. 121–128, ACM, 2008.

[16] D. Nakhaeinia, S. H. Tang, S. M. Noor, and O. Motlagh, "A review of control architectures for autonomous navigation of mobile robots," International Journal of Physical Sciences, vol. 6, no. 2, pp. 169–174, 2011.

[17] S. Murray, W. Floyd-Jones, Y. Qi, D. J. Sorin, and G. Konidaris, "Robot motion planning on a chip.," in Robotics: Science and Systems, 2016.

[18] D. Pena, A. Forembski, X. Xu, and D. Moloney, "Benchmarking of cnns for low-cost, low-power robotics applications," in RSS 2017 Workshop: New Frontier for Deep Learning in Robotics, 2017.

[19] S. Thrun, W. Burgard, and D. Fox, Probabilistic robotics. MIT press, 2005.

[20]

[21] B. Kehoe, S. Patil, P. Abbeel, and K. Goldberg, "A survey of research on cloud robotics and automation.," IEEE Trans. Automation Science and Engineering, vol. 12, no. 2, pp. 398–409, 2015.

[22] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," The International Journal of Robotics Research, vol. 34, no. 4- 5, pp. 705–724, 2015.

[23] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs,

[24] R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in ICRA workshop on open source software, vol. 3, p. 5, Kobe, Japan, 2009.

[25] Robotis, "Turtlebot3." http://emanual.robotis.com/ docs/en/platform/turtlebot3/overview/, Retrieved December 2018.

[26] Google, "Cloud speech-to-text." https://cloud.google.com/speech-to- text//, Retrieved December 2018.

[27] E. Yildiz, "Emotion classification." https://www.kaggle.com/eray1yildiz/emotion-classification, October 2018.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization,"

[29] CoRR, vol. abs/1412.6980, 2014.

[30] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," Machine learning, vol. 3, no. 2, pp. 95–99, 1988.

[31] S. M. Salaken, A. Khosravi, T. Nguyen, and S. Nahavandi, "Seeded transfer learning for regression problems with deep learning," Expert Systems with Applications, vol. 115, pp. 565–577, 2019.

[32] S. M. Salaken, A. Khosravi, T. Nguyen, and S. Nahavandi, "Extreme learning machine based transfer learning

algorithms: A survey," Neuro- computing, vol. 267, pp. 516–524, 2017.

[33] S. M. Salaken, A. Khosravi, T. Nguyen, and S. Nahavandi, "Output uncertainty score for decision making processes using interval type- 2 fuzzy systems," Engineering Applications of Artificial Intelligence, vol. 65, pp. 159–167, 2017.

[34] T. Nguyen, A. Khosravi, D. Creighton, and S. Nahavandi, "Eeg sig- nal classification for bci applications by wavelets and interval type-2 fuzzy logic systems," Expert Systems with Applications, vol. 42, no. 9, pp. 4370–4380, 2015.

[35] H. Haggag, M. Hossny, S. Nahavandi, and D. Creighton, "Real time er- gonomic assessment for assembly operations using kinect," in Computer Modelling and Simulation (UKSim), 2013 UKSim 15th International Conference on, pp. 495–500, IEEE, 2013.

[36] I. W. Documentation, "Ibm watson." https://www.ibm.com/watson/, Re- trieved December 2018.

[37] S. Crowe, "Google cloud robotics platform coming to developers in 2019." https://www.therobotreport.com/google-cloud-robotics-platform/, October 2018.

[38] A. R. Documentation, "Aws robomaker." https://aws.amazon.com/robomaker/, Retrieved December 2018.

[39] N. G. C. Documentation, "Gpu cloud." https://www.nvidia.com/en- us/gpu-cloud/, Retrieved December 2018.

[40] M. A. Documentation, "Cognitive services." https://azure.microsoft.com/en-us/services/cognitive-services/, Retrieved December 2018.