Embedding Consequence Awareness in Unmanned Aerial Systems with Generative Adversarial Networks

Guoxiang Zhang[1], Jose Alcala[1], Jeffrey Ng[1], Mighty Chen[1],
Xiangyu Wu[2], Mark Mueller[2] and YangQuan Chen[1]

***Abstract*— Small unmanned aerial systems (sUAS) are becoming more prevalent, driven by consumer interest and their potential for revolutionizing aspects of commercial applications, such as delivery of urgent goods. The expected ubiquity of such systems raises concerns about their safety, and the ability of such autonomous systems to operate safely in densely populated areas (where their value will be greatest). In this paper, we outline a new framework aiming to add an additional layer of safety to aerial systems operated by a human pilot or autopilot by monitoring the UAVs environment for visual cues, and monitoring the human pilot for signs of distraction. The system will endow a UAS with the ability to reason about its safety, and the consequences of safety failures during its operation. The UAS will furthermore continuously reason about possible safety maneuvers in response to likely failures – in the event of an emergency, the vehicle can then execute its last safe maneuver, thus reducing the systems impending danger. Embedding consequence awareness in sUAS is an obvious appeal to safer and more insurable missions. For pilot skill level awareness, a method utilizing generative adversarial networks, which improves pilot skill level classification accuracy in our experiments, is proposed to compensate limited training data availability.**

I. Introduction

As the Federal Aviation Administration (FAA, responsible for safety in the US airspace) keeps relaxing the sUAS (small-Unmanned Aerial Systems) flying restrictions, there is an increasing concern and need for a safer drone flight framework that can harness the human pilot experience, a drones own cognition of safe environment and safe operation, as well as robust flight control strategies that are consequences-aware.

Autonomous systems are typically engineered to minimize the likelihood of a fault occurring that would lead to the failure of the system. For aerial systems, such as drones, this need is particularly acute: if such a system experiences a failure that causes the system to lose the ability to continue normal flight, it poses a severe risk for anything below it. This is because flying systems necessarily have a certain amount of potential energy in operation. If such a system is able to reason about the consequences of possible failures, given its current environment and state, it could result in being a safer system.

A crucial aspect to the safety of human-operated machines is the ability of the operator to understand the safety consequences of their actions. A safety- and consequence-aware system is able to reason about potential actions, and could overrule a human operator's commands which are perceived to be dangerous. The system may, for example, decide to force an early landing in an unpopulated field, rather than fly over a crowded park. The system should also constantly be creating and storing contingency plans, which would allow minimizing the risk associated with its current actions should an unexpected failure occur.

Such contingency planning represents an additional layer to the typical fail-safe systems in a system. For example, a well-engineered system has an extremely low probability of having a critical subsystem fail – nonetheless, it is impossible to rule out such failures completely. The ability of a system to continuously reason about potential failures, and responses thereto that might minimize damage could act to reduce the potential harmfulness of any individual failure.

A typical strategy for dealing with an emergency in a drone is to execute a rapid landing, and come to the ground as quickly as possible. However, it is clear that such a strategy may be arbitrarily harmful, *e.g.* if a drone's emergency landing terminates in a children's playground. Instead, a system that is able to weigh the consequences of different potential outcomes, and decide on the least-harmful emergency behavior could be substantially safer.

Of especial interest is a system that operates with a human in the loop. The typical assumption is that the human is ultimately a good decision-maker, and can create "best" strategies in the event of an emergency. However, it is clear that this would often not be the case, *e.g.* if the human is paying insufficient attention, or if the human is, in fact, the cause of the failure (*e.g.* the human loses sight of a vehicle being remotely operated). In such situations, the system should be able to reason about the competency of the human in the loop, and the potential consequences of executing the human's commands versus overriding them with an autonomous emergency maneuver.

II. Proposed Consequence Awareness Framework

We propose to develop a simple but extensible system allowing a drone to detect whether it is in a safety critical environment. This will be done by combining visual cues, the drones state estimate, and a pre-existing map of hazards. We will also work on algorithms that allow the drone to create, in real-time, a set of alternative safety actions, each action based on a different contingency or fault occurring. For example, the system will constantly evaluate motions

This project is supported by CITRIS Seed Grant (2016-2018).

[1] Mechatronics, Embedded Systems and Automation Lab, School of Engineering, University of California, Merced, CA 95343.

[2]J. Dept. of Mechanical Engineering, University of California, Berkeley, Berkeley, CA 94720-1740

from where it predicts it will be in the near future to positions it has identified to be safe to land at. If an emergency arises, *e.g*. the human operator is no longer capable of flying the vehicle, it executes one of its contingency plans.

Depending on the real needs, the requirements for *Consequence Awareness* may be quite different. In this paper, we consider the following scenarios. As shown in Fig. 1, there were several fields of research which motivated the creation of *Consequence Awareness* in UAV.

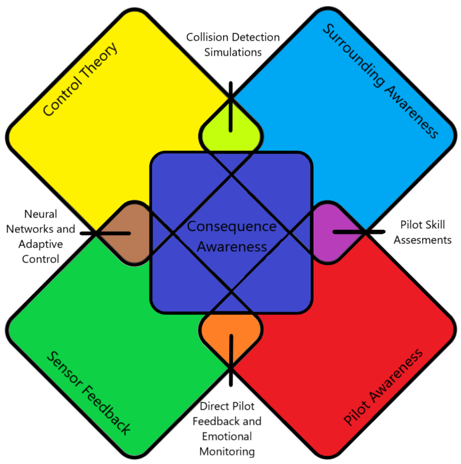

Fig. 1: Diagram of intersecting fields of research which comprise Consequence Awareness.

A. *Surrounding Awareness*

It is important that the drones are aware of their surrounding environment. Only when drones know their surrounding, consequences of specific actions can be predicted. This information includes the drones position in the world, combined with pre-existing maps and information from onboard sensors. Specifically, onboard cameras will be used to detect and characterize the area around the drone, allowing to detect *e.g*. whether the drone is flying over an area where the consequences of failure are likely be severe.

By combing object detection in 3D [1], [2] with a simultaneous localization and mapping system [3], [4], so that we can go one step further than single image based surrounding aware. This object-based 3D map will not be targeted to have a perfect 3D representation of the environment. Instead, it detects, calculates and remembers the location, velocity, rough size, category label of objects and consequence severity if they are crashed into. This abstracted map enables the possibility of real-time onboard processing and long-term storage. With velocity information of dynamic objects, it is made possible to reason on potential collisions and avoid them in advance.

B. *Pilot Skill Level Awareness*

Among many possible consequence awareness scenarios, our first focus is on using the pilots command signal to reason if the drone is under safe control of an experienced pilot or not. We conduct an offline analysis to determine signal signatures under various skill levels of a human pilot. And we propose to use Generative Adversarial Nets (GANs) for this analysis.

The idea of GANs is proposed in [5] by Goodfellow *et al*. It is a framework for estimating generative models via an adversarial process, during which a generative model $\mathcal{G}$ that captures the data distribution, and a discriminative model $\mathcal{D}$ that estimates the probability that a sample came from the training data rather than $\mathcal{G}$ are trained simultaneously. Coupled with deep convolutional neural nets (CNNs), deep convolutional generative adversarial networks (DCGANs) [6] are proposed to train CNNs in the unsupervised way, yet the learned features demonstrated their applicability as general image representations for other tasks.

Inspired by the achievements of GANs in computer vision and time series processing. GANs are attractive because of its weak requirements on training data, compared to other supervised classification algorithms. In pilot skill level analysis, the cost for collecting data from inexperienced pilots is high, since it is very likely that inexperienced pilots lead to crashes and damages. We propose to use GANs to then generate auxiliary data to enlarge the flight dataset.

C. *Action Consequence Awareness*

Motivated by the fact that even skilled pilots may make mistakes in a complicated environment, such as landing a drone close to a road when strong wind appears. It will be very helpful if there is a system available to check each control input and give warning on or even override dangerous actions. We envision that reinforcement learning will be useful for action level consequence awareness. Reinforcement learning is shown to be capable of achieving super-human performance on many tasks [7], [8], [9], such as that the Alpha Zero can learn to play the game of the Go by adversarial self-play without any human data input [8]. Reinforcement learning agents can be trained for specific piloting tasks, such as landing and flying at low altitude. And then use the model learned to evaluate the safeness of control actions from pilots. With a simulator that can detection collision and crashing, a learning agent can try different control inputs and observe the corresponding consequences. It is also cost-efficient to setup different dynamic environments so that the model can generalize well at different cases.

D. *Pilot Emotional Status Awareness*

We make this awareness an online detection and perform real-time alert (in the form of an audible tone, to remind the pilot to re-focus). For pilot wrist band signal, we perform a similar investigation. This will use some similar expertise developed in previous work [10], [11].

Affective Computing can help improve current UAS safety systems providing them intelligent and dynamic characteristics. Affective Computing(AC) is the process of detecting human emotional state based on a variety of data types [12]. These data types can include visual data like facial recognition, physical data like body position or body language, and collected data like surveys or flight logs [12], [13].

III. Related Work

Can or should a drone understand the emotional status of its human pilots? On May 31, 2013, Yahoo News reporter Liz Goodwin quoted the last author of this paper in [14]: Drones can also take the heart rate and other physiological data from their on-the-ground operators to gauge their stress levels. The system could be trained to take over from the human operator if it decides his or her stress levels are too high or that the operator is making irrational decisions. This idea motivated the study in [10] and rooted the idea of this project and the part of the thesis [15] where human drone operator's stress was shown to be able to get quantified by using operator's heart rate variability (HRV) analytics. Here we further review a few relevant studies as prior art.

A. Human Emotional UAS Encoding

The research conducted in [16], the UAV systems were encoded with predetermined emotional states which affected the flight path of the UAV. Participants viewed flights for each emotional state [16]. Each participant rated the emotional state of the UAV and the operator [16]. The goal in [16] was to create a visual observer an emotional state or intention of the drones behavior. This research falls short by not allowing the emotional stated to be dynamically defined by the pilot's input. From an outward appearance, our research will resemble the work by [16], due to the UAV limiting the pilot's actions therefore appearing to an emotional state.

B. Learning Drone Action and Action Consequence

In the research performed by [17] uses neural networks to predict the intent of UAS pilots [17]. As in the research conducted by [16], [17] is based on the observer's point of view *i.e.* security cameras feeds, human observer. In [17], the observer is focused on the pilot rather than the UAS. Video data was processed into two separate regions, one to identify the pilot and the other to identify the object used to control the UAS [17]. Then Neural Networks determined the pilots actions or intent by determine the movement of the device they used to control the UAS. This work shows how effective the neural network can learn the pilot's intentions from their observed behavior. In contrast, our Generative Adversarial Networks will have the task to help identify the pilot's experience level.

C. Fault Detection and Diagnosis and Neural Networks

In reviewing previous attempt to instill fault detection and diagnosiss techniques in [18] cover the major aspects of the field including UAV. In the field of UAV there is focus on the faults related to actuator, sensor and propeller damage [18], [19]. Attempts have been promising in controlling the UAV with Neural Networks techniques as seen in [20], [21], [22], [23], [24], [25], [26], [27], [28], [29]. Within this body of work, there have been advances on controlling the UAV class of nonlinear systems. Examples of Linearization by use of Neural Networks as seen in [24], [26], [29]. . The use of Neural Networks have been combine with other Nonlinear stabilization theory of Lyapunov like in in [28], [30], [23]. In inversion techniques on the nonlinear system are seen in [20], [27], these were used to help control the dynamics of the vehicle. The work of [25], [28] use of back propagating networks in controlling position or creating a dynamics observer, respectively. The research in [21], [22] use Neural Networks to control the UAV altitude. In [20] the vehicles control was accomplished with a Recurrent Neural Network. In contrast the work in [22], was accomplished using a mission-based approach and a Reinforcement Learning model.

IV. Method for Pilot Skill Level Awareness

In addition to the overall consequence awareness framework, a method for pilot skill level awareness is proposed. In the proposed method, pilot skill level awareness is achieved with supervised recurrent neural networks (RNNs) classifier, while its training data are partially generated by GANs.

We use RNNs with long short term memory (LSTM) [31] units as the classifier so that it can handle variable length data. RNNs also provide the model with the ability to model temporal relationship between different input dimensions. An LSTM unit is defined as in eq. (1):

$$\begin{aligned} \begin{bmatrix} i_t \\ f_t \\ o_t \\ g_t \end{bmatrix} &= \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} W \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix}, \\ c_t &= f_t \odot c_{t-1} + i_t \odot g_t, \\ h_t &= o_t \odot \tanh(c_t). \end{aligned} \tag{1}$$

RNNs used in this work are stacked LSTM units with a sigmoid function on the ouput. We denote it as a function approximator:

$$p = f_C(x|W_C), \tag{2}$$

where W_C is a parameter collection from W, as in eq. (1), of all LSTM units. x is the flight signal data. p is the probability that data x is from a skilled pilot. This model is trained with cross-entropy loss, as in eq. (3), using stochastic gradient descent.

$$loss = \sum_{i=1}^{N} -(y_i \log(p_i) + (1 - y_i) \log(1 - p_i)). \tag{3}$$

After the training process, we apply the RNNs model $f_C(x|W_C)$ to classify flight signal. It can calculate the probability of a signal series coming from a good pilot or an inexperienced one.

Due to limited flight data available for training the RNNs, we propose to use GANs to generate auxiliary data to assist training classifier. We use a network structure similar to the WaveGAN [32], which is derived from DCGAN by converting 2D convolution kernels to 1D while preserving the number of parameters per kernel. Specifically, as in Fig. 2, we use five layers for the generator $\mathcal{G}$. Three of them are transposed convolution with kernel size 16 and stride 4 or 2 for up-sampling. While the first layer is a dense projection and the last layer is a 1D convolution of kernel size 25 for output channel number reduction. The discriminator $\mathcal{D}$

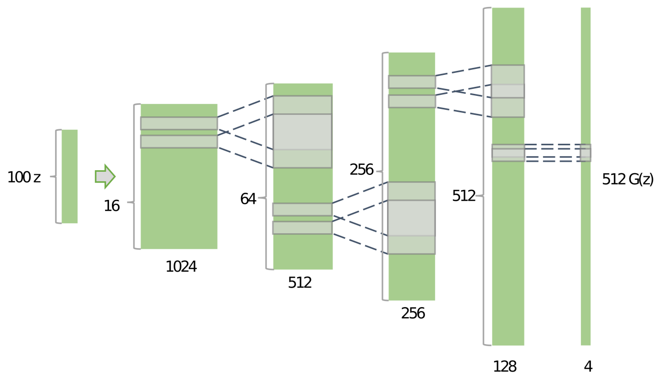

Fig. 2: The generator used for creating auxiliary data. A 100 dimensional Gaussian distribution Z is projected to a 128 channel feature map with sequential length 16. Three transposed convolutions then converts this feature map into 512 length feature map. The final convolutional layer then transfer it into a 4 channel 512 time step RC signal.

consists of four 1D convolutional layers with kernel size 16 and stride 4 and 2 (first layer). We train the GANs model by alternating between training discriminator by minimizing $-\log(\mathcal{D}(x))$ and training generator by minimizing $-\log(\mathcal{D}(\mathcal{G}(z)))$.

(a)

(b) (c) (d)

Fig. 3: The experimental flight space available at UC Berkeley is shown in (a), while three different available flight platforms are shown in (b)-(d): (b) shows the Bitcraze Crazyflie 2.0 [33], a 30g open-source quadcopter, and (c) shows the Parrot Bebop 2 [34], a consumer drone. A large, custom-made quadcopter is shown in (d).

V. Training Data Collection

As we mentioned, to acquire pilot skill level awareness, it is necessary to collect flight data with ground-truth available as a labeled dataset by purposely designed real flight tests. The following subsections introduce the details of both data collections methods.

A. *Indoor Drone Tests*

At UC Berkeley, we have access to a controlled, indoor flight space where we may conduct experiments on robotic flight, and human control thereof. The space, and some experimental vehicles, is shown in Fig. 3. The lab setup comprises a room of dimension $7 \times 6 \times 5$m, equipped with eight motion capture cameras for high-rate, high-fidelity state estimation. In this space, the team operates a number of different aerial robots, both of standard and custom designs, and at a variety of scales, ranging from less than 50g to more than 1.2kg.

Off-the-shelf components are leveraged whenever possible. The Robot Operating System (ROS) [35] is used as PC-side middleware, specifically connecting different components together and allowing for the recording (and later analysis) of experimental flight data. A custom low-level firmware is run on all vehicles, based off of the open-source Pixhawk project [36]. Custom vehicles use the Crazyflie hardware [33], which provides key electronic components in a compact, low-mass form factor.

In addition to operating fully autonomously, the experimental space allows for operation with human pilots, at various levels of autonomy. This means that the system may be set up for testing a human pilot's capabilities, using the extensive data logging capabilities for capturing data for either online or later offline analysis.

VI. Experiments

We started by manually collecting and labeling data. We labeled 300 flight logs, while 145 of them are positive and the rest 155 are negative examples. Then data is divided into a training set and a test set. The test set contains 60 logs in total with 30 in each class. We truncated data to the first 30 seconds for the training of classifier and 51.2 seconds for the training of GANs.

Each flight data log has 13 channels, including x, y and z for local position estimate, speed v_x, v_y and v_z, Yaw angle Ψ, pitch φ, roll θ, RC inputs channels RC_Ψ, RC_θ, RC_φ and RC_{th}. We take RC channels: RC_Ψ, RC_θ, RC_φ and RC_{th} for the current experiments.

With the dataset available, we train RNNs with different hyper-parameters, including hidden layer number, hidden layer size (HLS) and flight log channel combination. We use 2, 4, 8, 16 as hidden layer number options and 64, 128, 256, 512 as HLS options. The best average performance we get is 64.0% test accuracy by a model of 2 hidden layers $\times$ HLS 128. We use this model as a baseline.

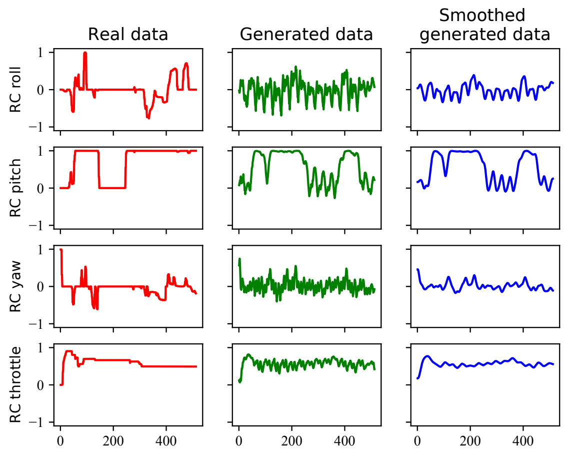

Fig. 4: Visualization of real and generated logs

Second, we train a GANs model on flight data from experienced pilots in training set and then generate 200 new logs to enlarge training set. Then we do the same with another GANs instance on inexperienced pilot data. Fig. 4 includes examples from real fight logs and generated flight data. With all the data available, we train a classifier without model selection, we get 80.8% classification accuracy on average, which shows improvements on 64.0% from the baseline. This shows that it is beneficial to the training of pilot skill level classifier by generating auxiliary flight data with GANs, when only a limited amount of labeled data is available.

VII. Conclusion

We outline a new framework aiming to add an additional layer of safety to aerial systems operated by human pilots. The system will endow a UAS with the ability to reason about its safety, and the consequences of safety failures during its operation. We also proposed a method for pilot skill level awareness. This method utilizes generative adversarial networks to enlarge the limited amount of training data, which shows its effectiveness by improving pilot skill level classification accuracy in our experiments.

ACKNOWLEDGMENT

This research was supported by a 2017 Seed Fund Award from CITRIS and the Banatao Institute at the University of California. We thank Amanda N. Olson and Karsten Widjanarko for their help on collecting and labeling data.

References

[1] A. Mousavian, D. Anguelov, J. Flynn, and J. Koeck, "3D bounding box estimation using deep learning and geometry," in *Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 5632–5640.

[2] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun, "3D object proposals for accurate object class detection," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 424–432.

[3] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[4] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semidirect visual odometry for monocular and multicamera systems," *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 249–265, April 2017.

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., 2014, pp. 2672–2680.

[6] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[7] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with deep reinforcement learning," in *Proceedings of NIPS Deep Learning Workshop*, 2013.

[8] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, *et al.*, "Mastering chess and shogi by self-play with a general reinforcement learning algorithm," *arXiv preprint arXiv:1712.01815*, 2017.

[9] C. Finn and S. Levine, "Deep visual foresight for planning robot motion," in *Proceedings of 2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 2786–2793.

[10] B. Stark, T. Patel, and Y. Chen, "HRV monitoring for human factor research in UAS," in *Proc. of the ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference.*, vol. V004T08A055, no. DETC2013-12746, August 2013.

[11] P. Oh and M. Goodrich, "Section XVII UAV - human interfaces and decision support systems," in *Handbook of Unmanned Aerial Vehicles*, K. Valavanis and G. J. Vachtsevanos, Eds. Springer, 2015.

[12] T. Yamauchi and K. Xiao, "Reading emotion from mouse cursor motions: Affective computing approach," 2017.

[13] M. Spivey, *The continuity of mind*. Oxford [u.a.]: Oxford Univ. Press, 2007, vol. 44.

[14] L. Goodwin, "Drones to enter public skies in 2015: Will it be safe?" https://www.yahoo.com/news/blogs/lookout/drones-enter-public-skies-2015-safe-095826982.html, accessed: 2018-02-26.

[15] M. Ko, "Applications of long range dependence characterization in thermal imaging & heart rate variability," Spring 2015, school of Engineering, University of California, Merced. [Online]. Available: http://escholarship.org/uc/item/4hx087tj

[16] J. R. Cauchard, K. Y. Zhai, M. Spadafora, and J. A. Landay, "Emotion encoding in human-drone interaction," in *Proceedings of 2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2016, pp. 263–270, iD: 1.

[17] S. Cho, D. H. Kim, and Y. W. Park, "Learning drone-control actions in surveillance videos," in *Proceedings of 2017 17th International Conference on Control, Automation and Systems (ICCAS)*, 2017, pp. 700–703, iD: 1.

[18] Y. M. Zhang, A. Chamseddine, C. A. Rabbath, B. W. Gordon, C. Y. Su, S. Rakheja, C. Fulford, J. Apkarian, and P. Gosselin, "Development of advanced FDD and FTC techniques with application to an unmanned quadrotor helicopter testbed," *Journal of the Franklin Institute*, vol. 350, no. 9, pp. 2396–2422, Nov 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0016003213000264

[19] M. Darrah, A. Rubenstein, E. Sorton, and B. DeRoos, "On-board health-state awareness to detect degradation in multirotor systems," in *Proceedings of 2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2018, pp. 1134–1141, iD: 1.

[20] M. T. Frye and R. S. Provence, "Direct inverse control using an artificial neural network for the autonomous hover of a helicopter," in *Proceedings of 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2014, pp. 4121–4122, iD: 1.

[21] B. Pugach, B. Beallo, D. Bement, S. McGough, N. Miller, J. Morgan, L. Rodriguez, K. Winterer, T. Sherman, S. Bhandari, and Z. Aliyazicioglu, "Nonlinear controller for a UAV using echo state network," in *Proceedings of 2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2017, pp. 124–132, iD: 1.

[22] W. Koch, R. Mancuso, R. West, and A. Bestavros, "Reinforcement learning for UAV attitude control," *ACM Trans. Cyber-Phys. Syst.*,

vol. 3, no. 2, pp. 22:1–22:21, Feb. 2019. [Online]. Available: http://doi.acm.org/10.1145/3301273

[23] T. Dierks and S. Jagannathan, "Output feedback control of a quadrotor UAV using neural networks," *IEEE Transactions on Neural Networks*, vol. 21, no. 1, pp. 50–66, 2010, iD: 1.

[24] B. Lee, H. Lee, and M. Tahk, "Analysis of adaptive control using online neural networks for a quadrotor UAV," in *Proceedings of 2013 13th International Conference on Control, Automation and Systems (ICCAS 2013)*, 2013, pp. 1840–1844, iD: 1.

[25] Y. Teng, B. Hu, Z. Liu, J. Huang, and Z. Guan, "Adaptive neural network control for quadrotor unmanned aerial vehicles," in *Proceedings of 2017 11th Asian Control Conference (ASCC)*, 2017, pp. 988–992, iD: 1.

[26] M. Jafari and H. Xu, "Adaptive neural network based intelligent control for unmanned aerial systems with system uncertainties and disturbances," in *Proceedings of 2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2018, pp. 1010–1016, iD: 1.

[27] Q. Lin, Z. Cai, Y. Wang, J. Yang, and L. Chen, "Adaptive flight control design for quadrotor UAV based on dynamic inversion and neural networks," in *Proceedings of 2013 Third International Conference on Instrumentation, Measurement, Computer, Communication and Control*, 2013, pp. 1461–1466, iD: 1.

[28] D. Nodland, H. Zargarzadeh, and S. Jagannathan, "Neural network-based optimal adaptive output feedback control of a helicopter UAV," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 7, pp. 1061–1073, 2013, iD: 1.

[29] H. Nakanishi and K. Inoue, "Development of autonomous flight control systems for unmanned helicopter by use of neural networks," in *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290)*, vol. 3, 2002, p. 2631 vol.3, iD: 1.

[30] D. H. Shim, H. J. Kim, and S. Sastry, "Control system design for rotorcraft-based unmanned aerial vehicles using time-domain system identification," in *Proceedings of the 2000. IEEE International Conference on Control Applications. Conference Proceedings (Cat. No.00CH37162)*, 2000, pp. 808–813, iD: 1.

[31] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, nov 1997.

[32] C. Donahue, J. McAuley, and M. Puckette, "Adversarial audio synthesis," in *Proceedings of the International Conference on Learning Representations*, 2019.

[33] Bitcraze. (2018) Crazyflie 2.0. [Online]. Available: www.bitcraze.io/crazyflie-2

[34] Parrot, "Parrot bebop 2," www.parrot.com/us/Drones/Parrot-bebop-2, accessed: 2017-02-23.

[35] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *Proceedings of ICRA Workshop on Open Source Software*, 2009.

[36] L. Meier, P. Tanskanen, L. Heng, G. H. Lee, F. Fraundorfer, and M. Pollefeys, "Pixhawk: A micro aerial vehicle design for autonomous flight using onboard computer vision," *Autonomous Robots*, vol. 33, no. 1-2, pp. 21–39, 2012.