Proceedings of the 2020 AIAA Scitech Forum, 6-10 January, Orlando, Florida, USA
DOI:10.2514/6.2020-0898

High Fidelity Progressive Reinforcement Learning for Agile Maneuvering UAVs

U. Can Bekar*
Istanbul Technical University, Istanbul, Turkey, 34469

Burak Yuksek†
Istanbul Technical University, Istanbul, Turkey, 34469

Gokhan Inalhan‡
Cranfield University, United Kingdom, MK43 0AL

In this work, we present a high fidelity model based progressive reinforcement learning method for control system design for an agile maneuvering UAV. Our work relies on a simulation-based training and testing environment for doing software-in-the-loop (SIL), hardware-in-the-loop (HIL) and integrated flight testing within photo-realistic virtual reality (VR) environment. Through progressive learning with the high fidelity agent and environment models, the guidance and control policies build agile maneuvering based on fundamental control laws. First, we provide insight on development of high fidelity mathematical models using frequency domain system identification. These models are later used to design reinforcement learning based adaptive flight control laws allowing the vehicle to be controlled over a wide range of operating conditions covering model changes on operating conditions such as payload, voltage and damage to actuators and electronic speed controllers (ESCs). We later design outer flight guidance and control laws. Our current work and progress is summarized in this work.

I. Introduction

Agility of unmanned aerial vehicles (UAVs), rather this be in operations such as urban air mobility (UAM) or cargo delivery, plays a crucial role in ensuring high precision maneuvering capability and control strategies in face of disturbances or anomalies. As such, modeling of uncertainties, sensor noises and process noises such as wind, turbulence, aerodynamic interaction around objects and ground still present a considerable challenge in achieving robust and safety-assured policies.

Beside of the high-fidelity modeling of the system dynamics and environment, flight control systems should provide closed-loop stability and meet flying quality requirements to perform a safe flight especially in the urban airspace. For this reason, variation of the dynamical characteristics of the aerial vehicle due to changes in inertial parameters and possible faults/failures on critical components should be considered in the flight control system design process. In the last decade, optimal adaptive controllers which include features of adaptive and optimal controller algorithms have promising results. Optimal controllers are designed offline by solving the Hamiltonian-Jacobi-Bellman equations. Adaptive controllers are designed to perform adaptation in the presence of parametric uncertainties and changes in dynamical characteristics. Hence, they are not designed to be optimal. To combine the optimal and adaptive control properties, a technique is developed known as reinforcement learning (RL) [1].

In literature, there are several applications of the RL-based controllers on unmanned aerial vehicles. In [2], state-of-the-art applications of bioinspired flight control systems that have the capability of self-learning are reviewed. Reinforcement learning has been applied to autonomous helicopters to learn how to track trajectories, specifically how to hover in place and perform various maneuvers [3–5]. In [6], using an RL decision process, a quadrotor agent is simulated within a grid world of 3 collapsed buildings and an upload station. In [7], authors proposed a survivability analysis and optimal mission planning methodology using RL for an aircraft flying in a human-made and hostile natural environment. In [8] the RL agents learn short-range, point-to-point navigation policies that capture robot dynamics

*Ph.D. Student, Department of Aeronautical Engineering, bekar18@itu.edu.tr, AIAA Student Member
†Ph.D. Candidate, Department of Mechatronics Engineering, yuksekb@itu.edu.tr, AIAA Student Member
‡Professor, School of Aerospace, Transport and Manufacturing / Centre for Autonomous and Cyber-Physical Systems, inalhan@cranfield.ac.uk, AIAA Associate Fellow

and task constraints without knowledge of the large-scale topology. In [9], authors applied RL to solve the problem of finding swing-free trajectories for rotorcraft. In [10], a hybrid control technique comprising model predictive control assisted with RL in the framework of guided policy search is developed. The first use of RL in quadrotor inner-loop control was presented by Waslander et al. [11]. The authors developed a model-based RL algorithm to search for an optimal control policy. In [12], Koch et al. presented an analysis of intelligent inner-loop flight control performance developed with RL compared to traditional proportional-integral-derivative (PID) control. Authors also developed a high-fidelity, 3 degree-of-freedom, open-source simulation environment and RL-based control algorithms are evaluated in this environment. In [13], an inverted pendulum balancing problem on a quadrotor platform is investigated and a solution is provided based on the RL method. The performance of the proposed control system is evaluated in simulation environment. It is shown that the control policy is computationally efficient and appropriate for real-time applications. In [14], a novel deterministic on-policy learning algorithm is developed and utilized to control a quadrotor platform. It is shown that, a high-performance policy can be learned by using zero-bias and zero-variance samples which are collected from a deterministic dynamical model. The proposed algorithm is demonstrated in both simulation environment and real quadrotor platform. Relatively accurate step response tracking performance is observed. Also, stabilization of the quadrotor platform is demonstrated under harsh initial conditions by throwing it upside-down attitude.

In this work, we present a high-fidelity model-based progressive reinforcement learning method for control system design for an agile maneuvering UAV. Our work relies on a simulation-based training and testing environment for doing software-in-the-loop (SIL), hardware-in-the-loop (HIL) and integrated flight testing within photo-realistic virtual reality (VR) environment as shown in Fig. (1). Using high-fidelity flight system identification test based mathematical models (as shown in Fig. (3) [15], and actual indoor flight system (utilizing Vicon precision localization), we provide an environment in which the UAVs learn the nonlinearities of the real-world by our novel hybrid approach. Real-world noise and their characteristics are learned and/or embedded by the simulation environment model. Through progressive learning with the high fidelity agent and environment models, the guidance and control policies build agile maneuvering as a means of proactive control, rather than the classical methods for reactive control, such as PID and linear quadratic optimal regulators (LQR). Our final agile UAV control system design is aimed at adapting to model changes such as payload, voltage, damage on actuators, electronic speed controllers (ESCs) and environment operating temperatures.

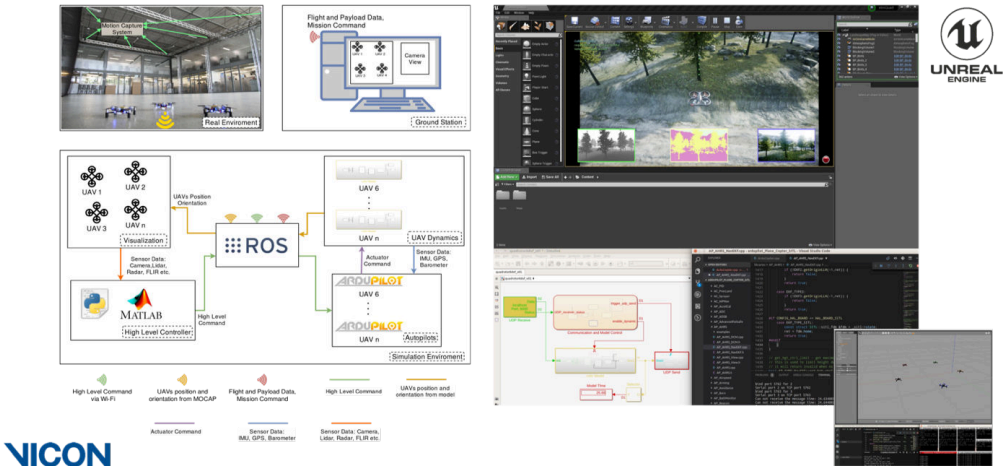

Fig. 1 Simulation-based Training and Testing: SIL, HIL and Integrated Flight Test Environment with VR.

Towards this goal, we have designed a progressive learning strategy in which attitude rate, attitude, velocity and position control is learned step-by-step, closing the loops, progressively. This approach allows us to build control capacity in a natural fashion similar to human learning behavior. This is critical as it provides expansion of the state-space coverage in a manageable fashion. As such, the inner-loop controllers have the ability to inhibit controlled instability, leading to agility. This is non-existent in classical stability based control system design philosophy. Reinforcement learning framework is capable of running in headless and batch modes, filling the experience replay buffer in parallel, without the need for rendering, hence speeding up the learning for the neural networks. The agility of the UAV is reinforced progressively and the stitching of learned maneuvers is monitored by the agility metrics that our team had been developing for the last decade [16–19]. Outer loop logic's main aim (trajectory planning and trajectory control) is to develop intelligent navigation strategies above and beyond imitation learning [20], leading to super-human performances

on newly seen environments. Besides, the above approach will allow us to further enhance control system strategy based on additional sensor inputs (such as camera, LIDAR) above and beyond IMU/GPS/INS measurements and provide the potential to embed in-flight (real-time) learning design for the inner and outer loop control system.

This paper is organized as follows; in Section II, general structure of the high-fidelity quasi-nonlinear mathematical model is given. In Section III, classical nested-loop controller design approach is summarized. In Section IV, closed-loop reference model (CRM) adaptive control system is introduced and an RL agent id utilized to improve its transient performance. In Section V, RL-based attitude and position control systems are developed and tested in simulation environment. In Section VI, concluding remarks and future works are given.

II. Mathematical Modeling

In developing and testing process of the control and guidance systems, it is crucial to have a high-fidelity simulation model which represents environmental effects and vehicle dynamics including airframe, actuators and sensors. There are two fundamental methods that can be utilized to obtain the model of a dynamical system. The first method is called physics-based modeling approach in which basic physical relationships are used to describe the system mathematically. However, this method requires extensive component-level analyses to characterize the system dynamics and it may not be practical in various applications.

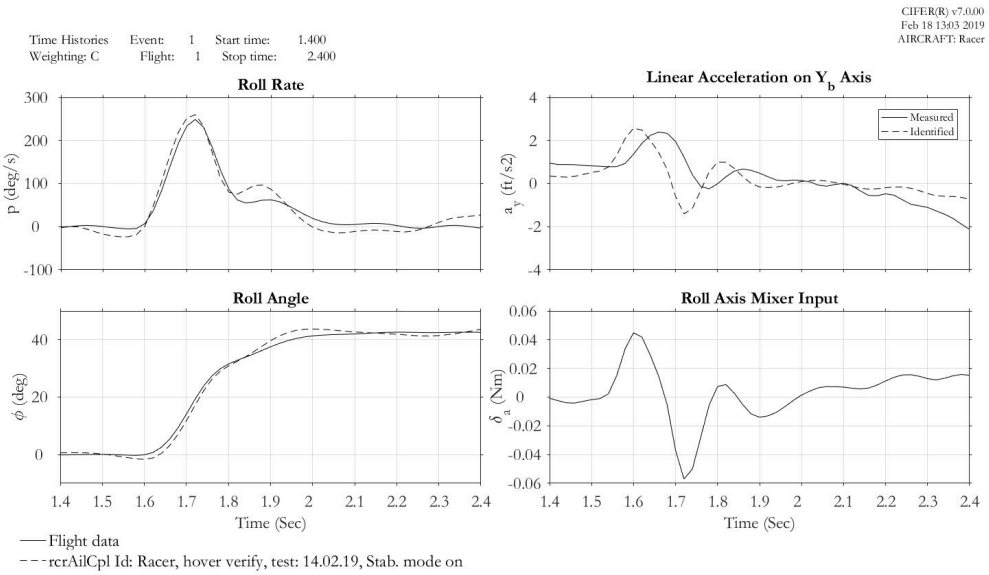

Fig. 2 Roll axis verification test results in hover flight.

The second method is called system identification in which pre-designed test signal is applied into the system and its responses are logged to identify the relationship between the input and output. It is basically an optimization process in which it is desired to minimize a cost function which is a function of error between system and identified mathematical model responses. In this study, the system identification is performed in the frequency-domain.

As a result of the system identification process, linear mathematical models are obtained for a specific flight condition. When the flight speed, altitude or total mass change, linear mathematical model may not provide accurate dynamical characteristics of the aerial vehicle. To obtain a high-fidelity model that covers the full-flight envelope, identified linear models are stitched together by using trim state data. This method is called *model stitching*. In this application a quasi-nonlinear model is obtained in which aerodynamic effects are calculated based on linear models, gravity effects and equations of motion are given in nonlinear form.

To generate the full-flight envelope simulation environment which is valid for hover and fast forward flight conditions, frequency domain system identification process is applied in both hover and forward flight with 20 m/s total airspeed. Linear mathematical models are obtained and they are verified in the time-domain verification analysis. As an example, roll axis verification results are given in Fig. (2).

Identified linear models are used in the stitched model which covers hover and forward flight phases. General structure of the stitched model is given in Fig. (3). Detailed information about the system identification, verification and

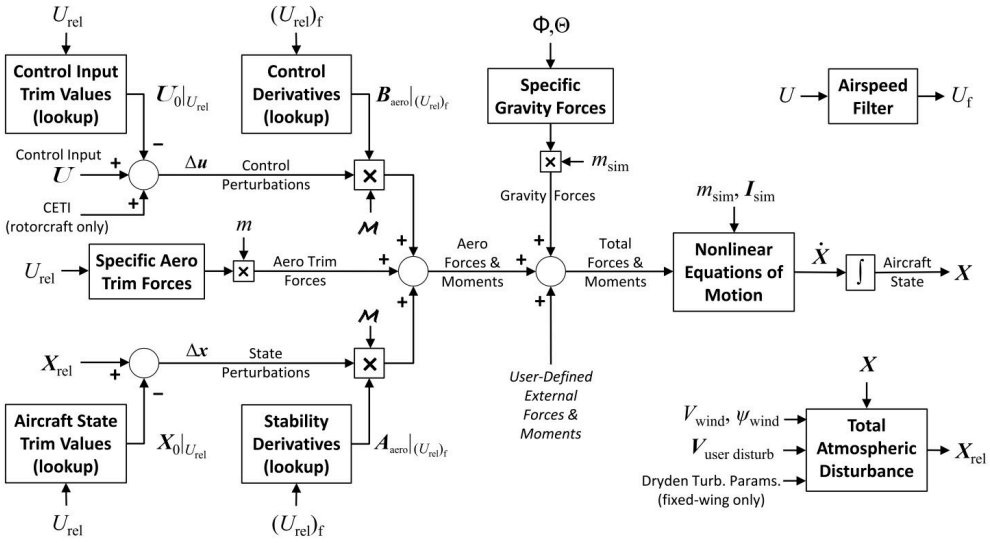

Fig. 3 General view of the stitch model structure [21]

stitched model development of the Racer quadrotor platform is given in [15].

III. Classical Controller Design

After obtaining linear and quasi-nonlinear mathematical models, controller design can be performed according to selected design specifications such as stability margins and disturbance rejection requirements. At the beginning of the controller design process, it is important to decide controller structure. In this study, a position control system is required for tracking the commanded position in North-East-Down (NED) coordinate frame. The proposed control system is based on nested-loop structure as shown in Fig. (4).

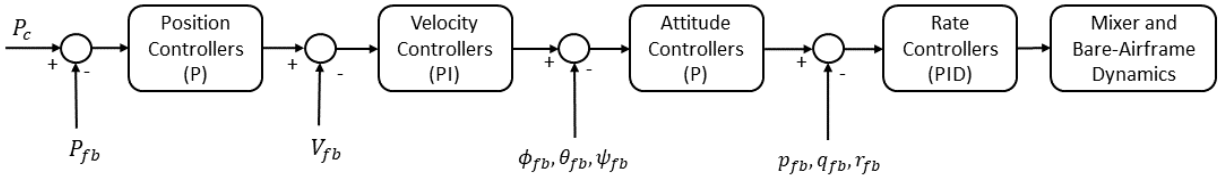

Fig. 4 Block diagram of the position control system.

Magnitudes of the controller parameters are directly related with the closed-loop system dynamics and hence they should be selected carefully. For this purpose, desired handling quality specifications are integrated into the multi-objective parameter optimization procedure in Control Designer's Unified Interface (CONDUIT) software which is used in several control system design projects for full-scale and sub-scale aerial vehicles [22]. Each of the controller parameters are optimized in CONDUIT software to meet the selected design requirements. Then, 3σ robustness analysis is performed on each loop to evaluate system dynamics in the presence of parametric uncertainties. As an example, roll attitude controller robustness analysis results are given in Fig. (5). As seen in this figure, closed-loop system handling qualities remain in the Level-1 region even in the worst case which represents the robustness of the proposed system. Detailed explanations of the control system design and test process for the Racer quadrotor platform, readers may refer to [15].

IV. Improvement of CRM-adaptive Control System by using Reinforcement Learning

In flight control system design applications for the aerial vehicles in urban airspace, it is quite critical to evaluate the closed-loop system stability and performance for different flight and weight conditions. Significant changes may

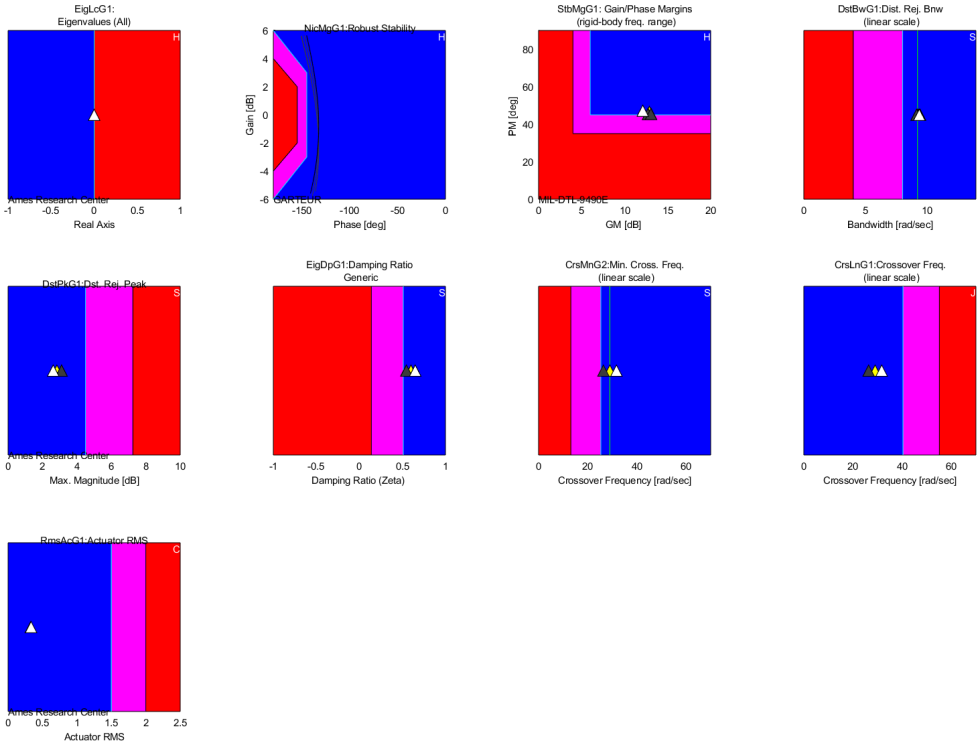

Fig. 5 3σ **robustness analysis of optimized roll attitude controller in CONDUIT for hover/low speed conditions.**

be observed in dynamical characteristics of the aerial vehicle as a result of variations in airspeed, altitude and mass properties. In addition, instability may also occur because of these variations which may lead to catastrophic accidents [23].

Adaptive control theory has promising results especially in flight control system design applications for aerial vehicles which have wide flight envelopes. It is able to compensate variations in the dynamical characteristics of the aerial vehicle and provide stability in different flight conditions. One of the fundamental applications of the adaptive control is Model Reference Adaptive Control (MRAC). Stability and adaptation of the closed-loop system is provided by the Lyapunov Theory. However, at the beginning of the adaptation process, high frequency oscillations are observed in adaptation parameters, control signal and system response. This may lead undesirable results in the flight control applications where operation safety is crucial.

To improve the transient response of the MRAC, closed-loop reference model (CRM) adaptive system is developed in which reference model has a feedback gain [24]. The feedback gain is optimized to minimize the oscillations in the control signal and peak system response observed in the transient phase. Also, in our previous study [25], optimized fixed-gain CRM-adaptive system is further improved by utilizing reinforcement learning (RL) method. In this algorithm, an RL agent is trained by using Deep Deterministic Policy Gradient (DDPG) algorithm to learn scaling policy of the optimized feedback gain. By using this scaling factor, a time-varying feedback gain is obtained and transient performance of the CRM-adaptive system is further improved. Mathematical description of the closed-loop reference model in the proposed RL-CRM algorithm is given in Eq.(1).

$$\dot{x}_m(t) = a_m\, x_m(t) + b_m\, r_{cmd} + L\, k(t)\, e(t) \tag{1}$$

where x_m is reference model state, a_m, b_m are parameters of the reference model, r_{cmd} is command signal, L is optimal feedback gain of the classical fixed-gain CRM-adaptive system, e is error between the reference model response and actual system response and k is scaling factor generated by the RL agent. General structure of the proposed RL-CRM system is given in Fig.(6).

In the proposed RL-CRM architecture, true error e^o is defined as the difference between the actual system response and open-loop reference model response. It is used to create the observation vector which is given in Eq.(2).

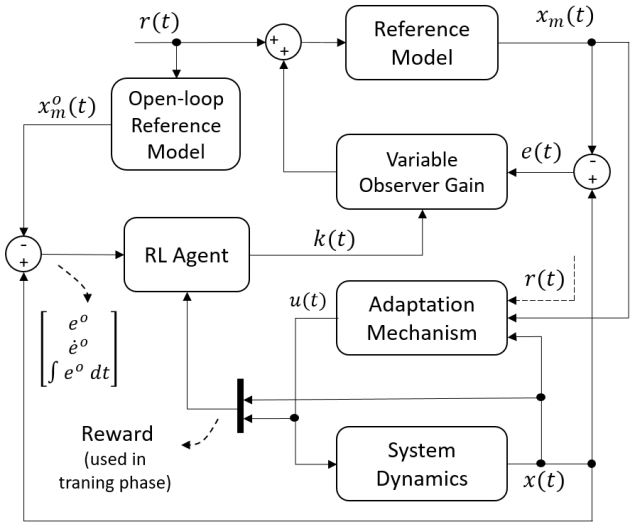

Fig. 6 General view of the RL-CRM adaptive control system structure [25]

Table 1 Transient response comparison of CRM and RL-CRM algorithms.

Performance Metrics	CRM (L_{opt})	RL-CRM	Improvement (%)
$\|\|\dot{\hat{k}}_{q_c}\|\|$	2.6601	2.2130	16.8076
$\|\|\dot{\hat{k}}_{q}\|\|$	1.9200	1.5622	18.6354
$\|\|\dot{\hat{\theta}}\|\|$	1.2213	1.0367	15.1150
$\|\|e\|\|$	0.1663	0.1383	16.8370
$\|\|\dot{u}\|\|$	2.1141	1.7640	16.5602

$$O(t) = \left[e^o(t),\, \dot{e}^o(t),\, \int e^o(t)dt \right]^T \tag{2}$$

In the DDPG algorithm, a reward function $R(t)$ is used to evaluate the system performance. It can be obtained by using the control, feedback and generated auxiliary data. In this study, it is proposed to minimize the system peak response, true error and command tracking error. The proposed RL-CRM algorithm is demonstrated on the linearized and simplified pitch dynamics model of the identified racer quadrotor platform.

Transient response performance analysis is performed in terms of several signal norms such as derivation of adaptation parameters ($\dot{\hat{k}}_q$, $\dot{\hat{k}}_{q_c}$, $\dot{\hat{\theta}}$), reference model tracking error (e) and control signal ($\dot{u}$). After the training process, transient response analysis is performed and results are given in Table (1). As given in this table, RL-CRM adaptive control algorithm provides improvement in the transient response of the system in terms of selected performance metrics. Also, time history of the MRAC, CRM and RL-CRM systems are compared in Fig.(7). Agent output is given in Fig.(8).

V. RL-Based Control System Design

For SIL environment we have used Matlab Simulink, utilizing Matlab Reinforcement Learning Toolbox for the training of our agents. We picked Deep Deterministic Policy Gradient (DDPG) [26] agent since it was the only agent currently made available inside the toolbox that made training for continuous action spaces. DDPG algorithm is an off-policy algorithm with actor-critic architecture. Neural networks used in trajectory tracking are shown in Figures (9) and (10).

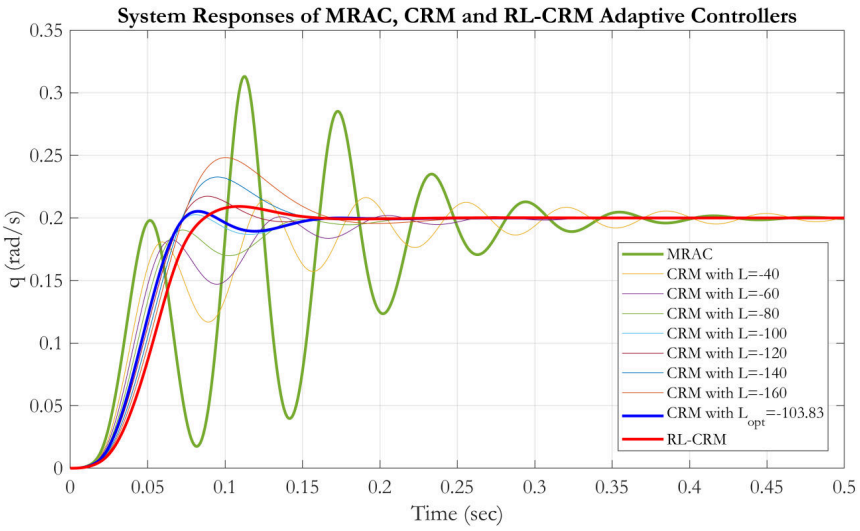

Fig. 7 Time history of the MRAC, CRM and RL-CRM system responses.

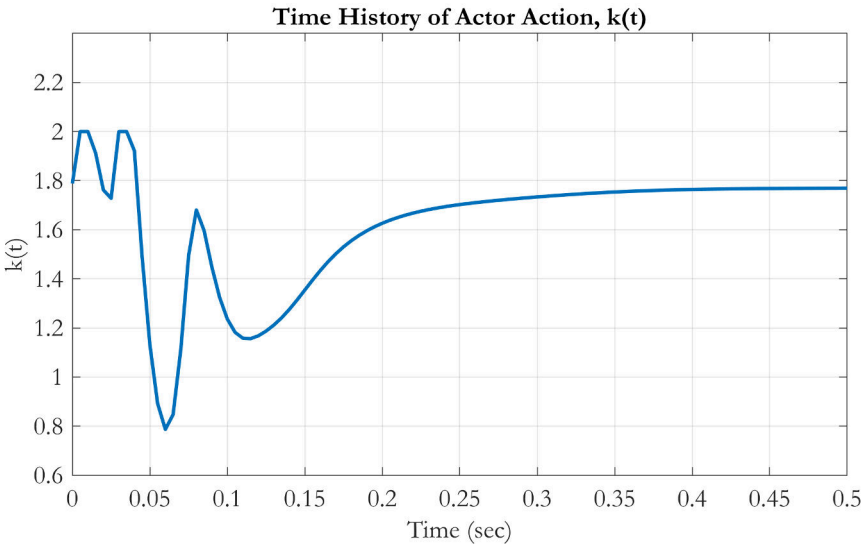

Fig. 8 Time history of the agent output.

Table 2 DDPG agent's training hyperparametes

Target smooth factor	1e-3
Discount factor	0.95
Mini batch size	64
Experience buffer length	1e6
Optimizer	Adam
Learn rate	1e-4
Gradient Threshold	1

A. Neural network architecture

Since our agent has an actor-critic architecture, we need two neural networks. Width of the first fully connected layers are 300 and in the remaining layers we have picked 400. Actor and critic networks are given in Figure (9) and Figure (10), respectively. The training process is expected to converge on optimal values after 50.000 steps of simulation for the outer loop. In Figure (9), green layer corresponds to observation inputs with a dimension of 6: North and east positions, target position coordinates in NED and linear accelerations in x-y plane. Orange layer represents the action output, north and east position reference signals. Blue layers represent fully connected networks. In Figure (10), green layer corresponds to observation inputs with a dimension of 6: North and east positions, target position coordinates in NED and linear accelerations in x-y plane. Orange layer represents the action output, north and east position reference signals. Blue layers represent fully connected networks, with the exception of addition layer. Q output is the approximated value scalar of the current policy.

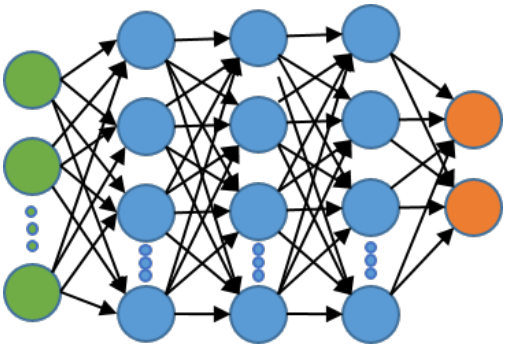

Fig. 9 Actor network's layer graph.

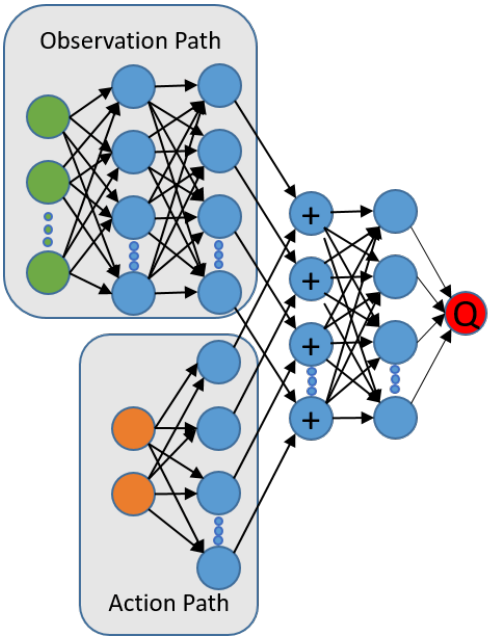

Fig. 10 Critic network's layer graph.

B. RL-based Trajectory Tracking Controller

The RL-based trajectory tracking controller has 6 observation and 2 action dimensions. Actions are north and east position reference signals and observations are linear accelerations in x-y plane, north-east position of the drone and north-east position of the target position. Rewards are two-fold, one is the negative distance to target position and in

addition, there is an agility reward function which at the simulation stop signal, evaluates the pitch axis agility of the drone using our agility metric $q_{max}/\Delta\theta_{max}$. During training we limit simulation time to 3 seconds and let the agent fly from hover initial condition to a point 1 meter away on the unit circle. A simulation trace from our trained agent can be found on Figure 11, showing a 1 meter away target position can be achieved after around half a second. Target coordinates are 0.75 m North and -0.67 m East. Blue lines are the target coordinates and green lines are the position reference outputs of the controller agent, red lines are actual states of the UAV. Simulation stops when the UAV is within 0.1 m range of the target, and in this simulation UAV reaches the vicinity around 0.75 seconds.

In a similar fashion, PID-based trajectory tracking response can be analyzed on Figure 12, showing a 1 meter away target position can be achieved after approximately 1.5 seconds. Target coordinates are 1 m North and 0 m East. Blue line is the position reference of the controller, red line is the actual state of the UAV. UAV cannot reach the 0.1 m vicinity of the target coordinates after 1 seconds.

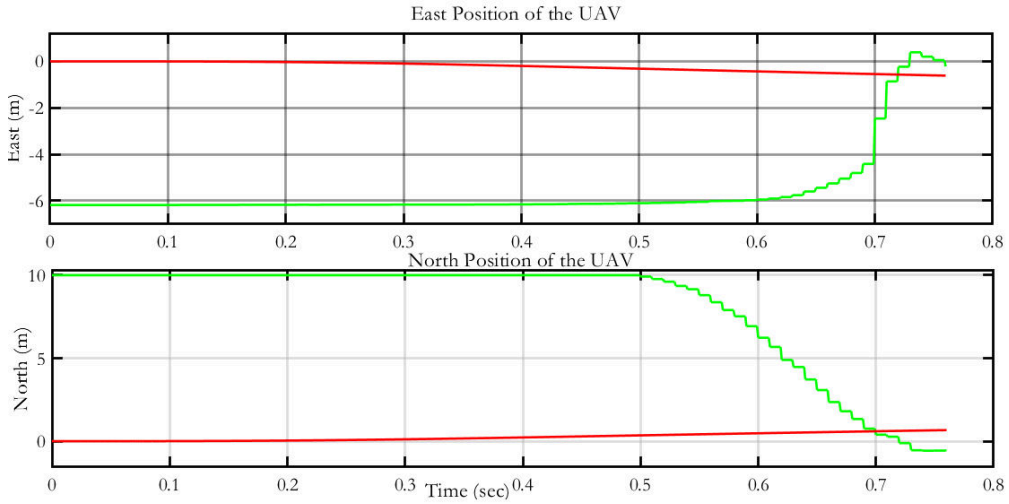

Fig. 11 A simulation trace of our RL-based trajectory tracking controller.

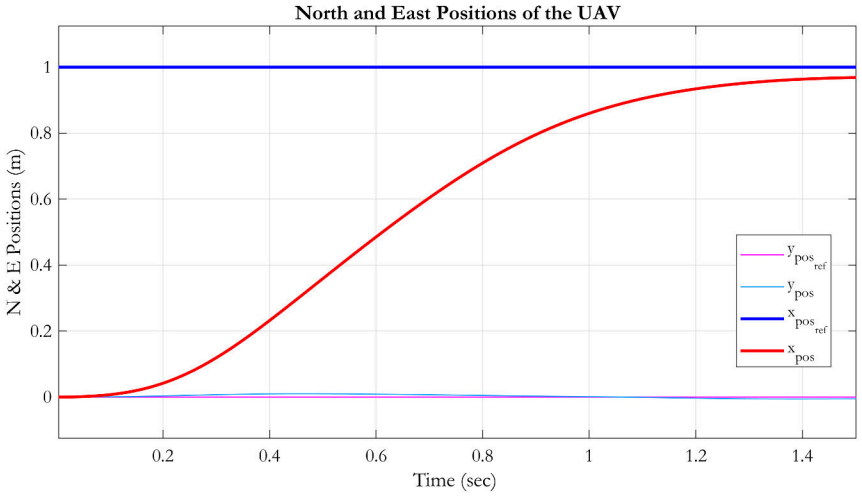

Fig. 12 A simulation trace of the traditional trajectory tracking controller.

C. RL-Based Attitude Controller

RL-based flight control method is also applied on the inner-loop attitude control system. In this application, RL agent is trained to provide hover flight. In training phase, identified linear longitudinal model of the Racer quadrotor is

used and the DDPG algorithm is utilized to update the actor-critic structure. Then, trained agent is integrated into the quasi-nonlinear 6-DoF simulation environment of the quadrotor platform and system is evaluated in the light turbulence conditions. Results for this application is given in Figure (13). Here, pitch attitude (θ) controller is based on RL. Roll and yaw rate controllers are kept as PID-based to evaluate the RL-based controller performance clearly.

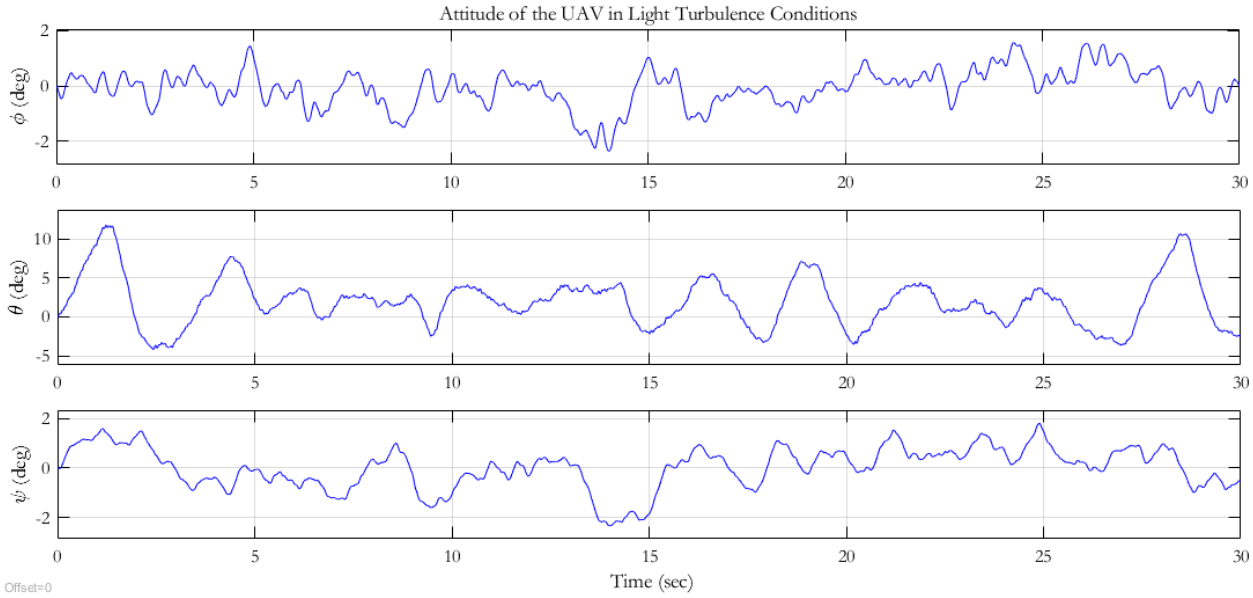

Fig. 13 A simulation trace of our attitude controller.

VI. Conclusion

In this work, we present a high-fidelity model-based progressive reinforcement learning method for control system design for an agile maneuvering UAV. We show that RL-based trajectory tracking and RL-CRM controllers outperforms PID-based trajectory tracking and classical adaptive controllers respectively. Agile maneuvers are performed using deep reinforcement learning approach. The methodology introduced provides the first stepping stone towards creating a unified progressive learning strategy for agile maneuvering UAVs. Current work focuses on enhancing the flight envelope and development of metrics that ensure antifragility as the uncertainty in models and environment increase.

References

[1] Lewis, F. L., Vrabie, D., and Vamvoudakis, K. G., "Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers," *IEEE Control Systems Magazine*, Vol. 32, No. 6, 2012, pp. 76–105.

[2] Santoso, F., Garratt, M. A., and Anavatti, S. G., "State-of-the-Art Intelligent Flight Control Systems in Unmanned Aerial Vehicles," *IEEE Transactions on Automation Science and Engineering*, Vol. 15, No. 2, 2018, pp. 613–627. doi:10.1109/TASE.2017.2651109.

[3] Bagnell, J. A., and Schneider, J. G., "Autonomous helicopter control using reinforcement learning policy search methods," *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, Vol. 2, 2001, pp. 1615–1620 vol.2. doi:10.1109/ROBOT.2001.932842.

[4] Abbeel, P., Coates, A., Quigley, M., and Ng, A. Y., "An Application of Reinforcement Learning to Aerobatic Helicopter Flight," *Advances in Neural Information Processing Systems 19*, edited by B. Schölkopf, J. C. Platt, and T. Hoffman, MIT Press, 2007, pp. 1–8. URL http://papers.nips.cc/paper/3151-an-application-of-reinforcement-learning-to-aerobatic-helicopter-flight.pdf.

[5] Kim, H. J., Jordan, M. I., Sastry, S., and Ng, A. Y., "Autonomous Helicopter Flight via Reinforcement Learning," *Advances in Neural Information Processing Systems 16*, edited by S. Thrun, L. K. Saul, and B. Schölkopf, MIT Press, 2004, pp. 799–806. URL http://papers.nips.cc/paper/2455-autonomous-helicopter-flight-via-reinforcement-learning.pdf.

[6] Junell, J., Van Kampen, E.-J., De Visser, C., and Chu, Q., "Reinforcement Learning Applied to a Quadrotor Guidance Law in Autonomous Flight," 2015. doi:10.2514/6.2015-1990.

[7] Baspinar, B., and Koyuncu, E., "Survivability Based Optimal Air Combat Mission Planning with Reinforcement Learning," 2018, pp. 664–669. doi:10.1109/CCTA.2018.8511604.

[8] Faust, A., Ramirez, O., Fiser, M., Oslund, K., Francis, A., Davidson, J., and Tapia, L., "PRM-RL: Long-range Robotic Navigation Tasks by Combining Reinforcement Learning and Sampling-based Planning," *CoRR*, Vol. abs/1710.03937, 2017. URL http://arxiv.org/abs/1710.03937.

[9] Faust, A., Palunko, I., Cruz, P., Fierro, R., and Tapia, L., "Learning swing-free trajectories for UAVs with a suspended load," *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 4902–4909. doi:10.1109/ICRA.2013.6631277.

[10] Zhang, T., Kahn, G., Levine, S., and Abbeel, P., "Learning Deep Control Policies for Autonomous Aerial Vehicles with MPC-Guided Policy Search," *CoRR*, Vol. abs/1509.06791, 2015. URL http://arxiv.org/abs/1509.06791.

[11] Waslander, S. L., Hoffmann, G. M., Jung Soon Jang, and Tomlin, C. J., "Multi-agent quadrotor testbed control design: integral sliding mode vs. reinforcement learning," *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 3712–3717. doi:10.1109/IROS.2005.1545025.

[12] Koch, W., Mancuso, R., West, R., and Bestavros, A., "Reinforcement learning for UAV attitude control," *ACM Transactions on Cyber-Physical Systems*, Vol. 3, No. 2, 2019, p. 22.

[13] Figueroa, R., Faust, A., Cruz, P., Tapia, L., and Fierro, R., "Reinforcement learning for balancing a flying inverted pendulum," *Proceeding of the 11th World Congress on Intelligent Control and Automation*, IEEE, 2014, pp. 1787–1793.

[14] Hwangbo, J., Sa, I., Siegwart, R., and Hutter, M., "Control of a quadrotor with reinforcement learning," *IEEE Robotics and Automation Letters*, Vol. 2, No. 4, 2017, pp. 2096–2103.

[15] Yuksek, B., Saldiran, E., Cetin, A., Yeniceri, R., and Inalhan, G., "System Identification and Model-Based Flight Control System Design for an Agile Maneuvering Quadrotor Platform," *Accepted for 2020 AIAA SciTech Forum (in Guidance, Navigation, and Control Session)*, 2020.

[16] Ure, N. K., and Inalhan, G., "Autonomous Control of Unmanned Combat Air Vehicles: Design of a Multimodal Control and Flight Planning Framework for Agile Maneuvering," *IEEE Control Systems Magazine*, Vol. 32, No. 5, 2012, pp. 74–95. doi:10.1109/MCS.2012.2205532.

[17] Moghadam, M., Ure, N. K., and Inalhan, G., "Autonomous Execution of Aircraft Supermaneuvers with Switching Nonlinear Backstepping Control," *2018 AIAA Guidance, Navigation, and Control Conference*, 2018, p. 1594.

[18] Akcal, U., Hostas, B., Ure, N. K., and Inalhan, G., "Recoverability envelope analysis of nonlinear control laws for agile maneuvering aircraft," *2018 AIAA Guidance, Navigation, and Control Conference*, 2018, p. 1865.

[19] Yildiz, A., Akcal, U., Hostas, B., Ure, N. K., and Inalhan, G., "Finite state automata based approach to autonomous stall and upset recovery for agile aircraft," *2018 AIAA Guidance, Navigation, and Control Conference*, 2018, p. 1867.

[20] Uzun, S., Akbiyik, B., Yuksek, B., Demirezen, U., and Inalhan, G., "A Simulation-Based Machine Learning Approach for Flight Control System Design of Agile Maneuvering Multicopters," *AIAA Scitech 2019 Forum*, 2019, p. 1978.

[21] Tischler, M. B., and Tobias, E. L., "A Model Stitching Architecture for Continuous Full Flight-Envelope Simulation of Fixed-Wing Aircraft and Rotorcraft from Discrete Point Linear Models," Tech. rep., Aviation and Missile Research, Development and Engineering Center, 2016.

[22] Tischler, M. B., *Practical methods for aircraft and rotorcraft flight control design: an optimization-based approach*, American Institute of Aeronautics and Astronautics, Incorporated, 2017.

[23] Dydek, Z. T., Annaswamy, A. M., and Lavretsky, E., "Adaptive control and the NASA X-15-3 flight revisited," *IEEE Control Systems Magazine*, Vol. 30, No. 3, 2010, pp. 32–48.

[24] Gibson, T. E., Annaswamy, A. M., and Lavretsky, E., "Adaptive systems with closed-loop reference-models, part I: Transient performance," *2013 American Control Conference*, IEEE, 2013, pp. 3376–3383.

[25] Yuksek, B., Demirezen, U., and Inalhan, G., "A New Reinforcement Learning Based Approach for Closed-loop Reference Model Adaptive Flight Control System Design," *Submitted for American Control Conference (ACC) 2020*, 2020.

[26] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D., "Continuous control with deep reinforcement learning." *ICLR*, edited by Y. Bengio and Y. LeCun, 2016. URL `http://dblp.uni-trier.de/db/conf/iclr/iclr2016.html#LillicrapHPHETS15`.