RL MSc theses

A compiled description of the previous RL MSc theses that i could gather, sorted chronologically.

1. Joost Dorscheidt (N/A, Nov 2018): **flexible DDPG** (curriculum(ed) learning), novelty: the algo used
2. Jonathan Hoogvliet (20 months, Nov 2019): **HRL** (Hierarchical Reinforcement learning) , novelty: the algo used

> Hierarchical RL aims to improve the sample efficiency of flat RL, a coin termed by Dietterich to describe the nominal form of RL. A common problem with most RL methods is the curse of dimensionality resulting in poor sample efficiency, which is commonly tackled by using function approximators (e.g. ANN) for many of the estimated functions in an RL agent with successful examples being `SAC` or `TD3`. HRL improves on sample efficiency by acknowledging the fact that time passes during an MDP, and that activities are usually composed of sub-activities/actions -much like in human activities-. The main formulations of HRL comes in the form of: `options, HAM, and MAXQ`. Learning in HRL is different than flat-RLs wherein *multiple* policies are trained instead of just *one*, and these policies can be placed in hierarchies where a higher level policy invokes lower level ones, thus learning decomposes into training high level policies to activate appropriate lower policies, and training lower level policies into mimicking the appropriate primitive actions. This structure is also said to be useful for creating guidance and navigation agents, as opposed to simply controllers.
>
> A swingup pendulum problem is used to illustrate the application of HRL, as an efficient controller can be devloped by exploiting the split between a swing-up task and a balancing task. For this problem, a scheduled swing-up and PD controller, a FRL controller (Q-learning algo), an HRL controller, and a hybrid (has built in PD controller) HRL controller are implemented. Where it was found that the hybrid HRL and FRL controllers performed joint best, but due to the fact that this preliminary problem has a small action/state space, it was presumed that the performanceof FRL would not translate to the aircraft enviornment and hence hybrid-HRL is said to out-perform FRL.
>
> Then the in-depth analysis involved using 3 RL architectures for altitude control of F-16: `FRL, HRL sans TA, HRL + TA`, all using a 3 layer h-gamma-q controller structure, with TA enabled through options. FRL had the poorest sample efficiency and tracking performance, while HRL + TA had the best sampling efficiency and HRL sans TA had the best tracking error. Although, none of the controllers presented either **a)** outperformed controllers made with other algos (e.g. Killian with SAC or Willem with TD3), and probably could be outperformed by a PID.

3. Dave Kroezen (12 months, Apr 2019): **DHP**, novelty: the algo used
4. Stefan Heyer (12 months, Mar 2019): **IDHP** i think?

> Stefans' scope was focused around actor-critic algos because they can be applied online, to continuous control problems, and have better sample *complexity*. The specific class of algos considered are called actor-critic designs (ACDs) which fall under the umbrella of a model-based set of algorithsm called ADP (except for the element ADHDP). ADP's generally require an initial offline training phase, but *supposedly* can be avoided if an incremental approach is applied to ADP methods, such as IDHP. The main difference betweeh DHP and its incremental version IDHP

> is the model that each algorithm uses, where in DHP the model used should be at initialization and throughout training/testing a complete description of the model dynamics (aka enviornment dynamics), whilst IDHP's model can be an incrementally learnt one. Both DHP and IDHP can be trained online as evident by the fact that training is performed over the span of one episode (one flight), which goes detracts from his claim that "(ACDs) have the commone limitation that an initial offline training phase is required". The main results indicate that IDHP is able to deal with controlling flight through regions of the flight envelope previously unencountered, as well as adapt to changes in aircraft dynamics (e.g. failure in ailerons or disturbance injection at plant input) owing to the use of an RLS estimator to identify aircraft dynamics online.

5. Ramesh Konatale (21 months, Jan 2020): **iADP**, novelty: the algo used, used on Flying-V

> Ramesh concluded from his literature review that iADP would be the most suitable online learning controller, owing to their ability to produce theoretically optimal actions, and the ability to handle nonlinear systems by using RLS to incrementally identify models of the controlled system. Approximately, iADP is an algorithm which uses optimal control theory to implement *GPI*. First a preliminary analysis is done to compare the regulation performance of state vs output feedback control of a missile, in the face of a single disturbance kick through implementing them on LADP; as well as comparing LADP to its' linear counterpart the iADP. The first analysis showed that GPI convergence is faster using SF over OF, and that the (spectral) property of the exciting signal is important in the case of OF, and less so in the case of SF. The second preliminary test showed that iADP has better disturbance rejection performance than LADP. The main analysis of iADP is carried out by investigating the controllers robustness, e.g. susceptability to sensor and actuator dynamics (bias, noise, delay, saturation). It was shown that the effects of sensor discretization, sensor bias, & actuator delays had insignificant impact of controller performance or incremental sys id; while sensor noise, & sensor delays did have a noticeable impact on controller performance and incorrect system identification, but can be mitigated through filtering.

6. Killian Dally (10 months, Feb 2021): **SAC**, novelty: the algo used

> Killian applied the SAC algorithm to flying the Citation through different challenging aerial maneouvres, including climbing, decending turns, and high bank angle turns. He first developed a single feedback loop RL controller (tracking altitude, roll, and sideslip) which had very poor robustness, remedied by adopting a cascaded control architecture instead where altitude and attitude are controlled with seperate controllers. He proposed in his research to have agents actions be actuation rates/increments instead of exact amplitude of actuation. He remarks that actuation during sinusoidal reference following results in *very* noisey (oscillatory) actuation, though looking at the graph results it seems that the noise in actuation *and pitch* exists for altitude tracking task as well. He suggests that training should be diversified to include smoother references to overcome this issue, theorizing that this effect is due to lack of exposure to smooth references.A noteworthy result obtained was a reward function sensitvity analysis, where he observed that sampling effciency was at least 1.6 times higher when the reward was based on a clipped MAE function istead of a clipped MSE, throughout the report he clipped this error clipped to the interval [-1, 0] as suggested by Mnih et al to avoid large gradient updates.

7. Zhou Xin Ge (N/A, Aug 2021): **PPOC** (Proximal Policy optimization with Option-Critic).

> Zhou's thesis used an algorithm called PPOC rooted in the combination of an Option-Critic architecture and Proximal Policy Optimizatino (PPO) architecture. This algorithm is compared in terms of their *adaptivity* and *sample efficiency* with PPO in a multi-mass-spring-damper (MSD) trained offline, where adaptivity is tested by changing the environment properties drastically midway through training (maing the system unstable by flipping the sign of the spring or daamping constants), this is so called "offline adaptivity". The results of MSD training indicates that sample efficiency of PPOC is inferior to PPO, but it is much more adaptive to system changes during training. Then PPOC in two variants is trained on a longitudinal LTI model of the citation taken from lecture notes, variant 1 has 1 option and variant 2 has 2 options. Training in this environment was done offline and online. Offline, the two variants converged to similar reward levels, but PPOC's reward variance was higher owing to the fact that it can select from multiple options. To apply PPOC online, the algorithm is simply trained for one episode of flying. Online is where the most important comparisons are gleamed, where it is found that PPOC 2 has better fault tolerance (loss of control effectiveness, suddenly resonant pitching dynamics), and could be more readily transferred to a new aircraft with different dynamics.

8. Willem Völker (10 months, Jun 2022): **TD3** , novelty: the algo used (didnt seem like prev studs used TD3), used on flying-V

> Willems' thesis begins with a careful examination of SAC and TD3's history, which has their origins in value and policy based learning as they are both actor-critic algorithms. And he emphasizes the lack of consensus on which is better; TD3 or SAC. But because of the fact that SAC is **a)** trained to give a distribution of actions and **b)** observed to have oscillatory actuation, he opted for TD3. As the preliminary run, he explores how 4 hyperparameters of TD3 affects its' learnnig, namely: `learning rate, exploration noise, discount rate, and batch size`. The focus of the paper, i.e. the system he applied TD3 to was the Flying-V, specifically the VLM based model, on which he demonstrates that TD3 can satisfy a 20m altitude tracking requirement. However, the altitude time trace is oscillatory for the ramp and saw wave references. And moreover is as robust to aerodynamic uncertainty as INDI. However in his results he observes that the actuation is still oscillatory inspite of his inferred claim that TD3 generates smoother actions than SAC. But whether it is more or less oscillatory than actuation from a SAC agent is not known, as no rigorous comparison is performed. He notably has tried to counter this oscillation not by using CAPS, but by adding penalty to the reward function for high elevon rates, which failed. He then adopted Dally's method which defined an agent's action as a normalized actuator deflection rate, instead of deflection angle itself, which solved the oscillation issue.

9. Casper Teirlinck (11 months, Sep 2022): **SAC + IDHP**, novelty: hybrid (2+ algos combined) online offline

> Casper was the first from C&S to implement a hybrid offline-online trained RL agent, which he trained on the citation environment, which showed improved tracking and fault tolerances than a SAC only controller. The hybrid agent is comprised of an offline trained SAC controller, and an online trained IDHP controller. It is created by initializing the Actor network of the IDHP with the network learned from the offline SAC algorithm, and then splicing additional identity hidden layers in between the existing hidden layers. The critic-network during online control, on the other hand, is purely learned by IDHP with zero information used from the critic learnt in offline training, due to the nature of IDHP's critic network. The control problem tackled in the 2nd research phase was a full 6DOF **attitude** control, which is controlled by a pair of lateral-

> longitudinal controllers, a standard split in aircraft dynamics. The hybrid controller still requires excitation along all axis to properly initialize the IDHP controller. It is only after this initial online training that the hybrid controller is to be "flown".

10. Peter Seres (10 months, Oct 2022): **DSAC**, novelty: the algo used

```
> Peter's thesis revolved around comparing distributional algorithms with their
traditional counterpart, with the focus being DSAC vs SAC, noting that a [previous
source] (from which an implementation of DSAC is available) demonstrated that DSAC
outperformed SAC and TD4 (distributional variant of TD3). The comparison of DSAC
vs SAC was performed by comparing their training in two environments, the OpenAI
pendulum-v1 gym and a linearized citation taken from Flight Dynamics course
AE3202. From his results, Peter concluded that DSAC had better sample efficiency
and reference tracking performance than SAC, using the analogy that
"distributional RL compared to traditional RL as taking a colored photo vs
greyscale photo". This is further emphasized by citing different sources that
claim distributional RL increases sample efficiency (source [1], [2], [3]). In the
return over samples plots of his scientific paper, it seems that SAC & DSAC both
drop in reward before rising, something not commented upon. Furthermore, his
implementation of SAC converged faster than Killians' implementation (convergence
in 3e4 vs 1e6 samples)
```

11. Vlad Gavra (11 months, Jan 2023): **TD3**, novelty: ERL with safety & TD3

    > Vlads' contribution consists of developing an RL framework which he coined SERL, safety-informed evolutionary reinforcement learning. He starts by picking PDERL from a list of ERL's and comparing its performance with that of DDPG and CMA-ME, from which he demonstrates PDERL's superiority in fault tolerance. SERL is then created by replacing the DDPG actor-critic algo in PDERL with TD3 (to improve sample efficiency), and by introducing a safe mutation operator to the ERL. This safe mutation is a direct copy of the pre-existing proximal mutation operator in PDERL, except it defines a new set of genetic memory D_c which contain transitions that cause the aircraft to enter an alpha >= 11 deg and roll angle >= 60 deg. To make the best use of the pool of trained policies in an ERL, Vlad proposes an online policy switching mechanism. Here, based on real time simulation of each policy and their resulting performances, the best performing policies are switched out in a hard (complete replacement) or soft (weighted average) manner with the acting policy.

12. Thomas vd Laar (13 months, Jun 2023): **SAC**, novelty: used in landing

    > Thomas used SAC to handle the task of automatic landing. He first compared the performance of 5 DRL algorithms along with one PID controller in the Lunar Lander OpenAI gym environment, and concluded that SAC was the best in terms of robustness (variance of reward in nominal and perturbed condition). The five algorithms compared were taken directly from Stable-Baselines (linked to v3 but Thomas might have used v1): `DDPG, TD3, SAC, A2C, & PPO`. Then SAC was applied to control a simulated Citation to perform landing. Wherein a 3D ILS model was implemented, along with a cascaded swithcing controller structure; very similar to the control structure made in the AFCS practical. A choice was made to train seperate pitch and roll SAC controllers, and then combine the two into one controller, though reasons for this choice could

> not be found, it is assumed that this results in a lower dimension RL problem, in addition to the standard practice in flight control of decoupling lateral and longitudinal motions. The SAC implemented adopted ideas of simulated annealing proposed by K Dally for learning rate. An interesting result observed was that the SAC controller used much higher actuation than the PID controller (~10 vs ~3 deg deflections).

13. Lucas Viera (11 months, Jul 2023): **DSAC + IDHP**, novelty: hybrid (2+ algos combined) online offline

> Lucas' thesis extended the research done by Casper, by replacing SAC with DSAC in the hybrid approach proposed by Casper, while keeping the online algorithm as IDHP. This swap was motivated by Peter Seres' finding that DSAC had improved learning performance compared to SAC. First a comparison of the offline algorithms: SAC vs TD3 vs DSAC was carried out over a series of experiments, which varied in the tracking tasks performed, the observation vector available to the RL agent, and the reward function. He found that the returns of TD3 are competitive with SAC and DSAC across most experiments, exceeding and subceeding them in some. But he found that variance of TD3 returns is much higher than that of SAC and DSAC. Then the hybrid controller is created using the setup created by Casper, which showed better performance than the original hybrid and the SAC or DSAC only controllers.
>
> He performed a "reliability analysis" of his algorithms to show that the distribution of results are statistically significant, or reliable. Where he used the idea of "stratified bootstrap confidence intervals". And ultimately the hybrid methods are shown to have superior robustness (sensor noise + biases), fault tolerance (reduced actuator effectiveness) and performance (reference tracking nMAE) in compared to purely offline algorithms. Weirdly, the results showed that SAC out performs DSAC in terms of tracking nMAE.