Delft University of Technology
Department of Control & Simulation

Research Plan v1.0

W. Chan

Date:	20/02/2024
Subject:	RL for Flying-V
Supervisors:	Dr. E.J. van Kampen
Project term:	01/2024 - 08/2024
E-mail:	w.y.chan@student.tudelft.nl

1 Introduction

Civil aviation is a sector that has a good track record in terms of transport safety [1], this excellent record is nonetheless tarnished by individual accidents that involve high fatality rates. The most fatal category of such accident is known as Loss of Control – Inflight (LOC-I), which groups together accidents that occur due to an aircraft deviating from flight path or aircraft operation outside normal flight envelope; 94% of accidents in this category resulted in fatalities [2]. Fault-tolerant or robust controllers are flight control methods that remain performant and stable despite changes in the dynamic properties of an aircraft or operation in extreme conditions. One of the most exciting areas of research in fault-tolerant controllers is the application of reinforcement learning (RL) as an agent to control aircraft. This approach can allow for directly optimizing a controller for scenarios that lead to LOC-I accidents [3], or even online adaptation of controller parameters in the face of dynamic behavior changes [4], all of which provides the flight controller with greater safety margins.

In addition to flight safety and control design efforts, there is also the issue of sustainability. Based on 2018 data the aviation industry is responsible for approximately 2.4% of global anthropogenic greenhouse gas emissions [5]. Despite aviation traffic volumes falling drastically as a result of the COVID-19 pandemic, the recovery of air travel has been rapid and the number of flights is expected to recover soon, ultimately resulting in the aviation sector contributing 0.1 °C of warming by 2050 [6]. To alleviate the severity of climate change, the aviation industry must therefore become carbon neutral, which will require innovation in all aspects related to aviation. On this front, the novel aircraft design concept of Flying-V promises fuel efficiency improvements over current tube-and-wing designs, and thus increasing the technological readiness level of the Flying-V is of special interest. This maturity will require dedicated efforts toward the design of flight control systems, which traditionally involves dedicated expert knowledge of control theory and time-intensive design campaigns [7].

Research Questions

Thus, the Flying-V presents an interesting opportunity as a test-bed for reinforcement learning-based flight control systems, which can autonomously "learn to fly". This leads to the objective of the present research:

Research Objective

To advance the state-of-the-art reinforcement learning based flight controllers and further the technological readiness level of the Flying-V, by developing an intelligent and fault-tolerant flight control system for the Flying-V.

This objective is broken down into specific subquestions to fully define the scope of the present research:

Research Questions

1. What reinforcement learning algorithm can yield a flight controller that is the most tolerant to faults?
   (a) What RL algorithms are considered to be state-of-the-art?
   (b) How is fault tolerance defined and tested in past research?
   (c) Which algorithms have been shown to provide the best fault tolerance?
   (d) What reference tracking performance have these algorithms shown in past research?
2. What are the flight control-related challenges when it comes to designing a controller for the Flying-V?
   (a) What are the flight dynamic properties of the Flying-V and what implications do they have on an AFCS system?
   (b) What are potential fault scenarios that warrant attention from the AFCS system in the Flying-V?
   (c) What criteria and metrics should be used to characterize a controller's fault tolerance?
3. How can the identified algorithm be applied to control the Flying-V?
   (a) How should the flight control system be structured?
   (b) What are the variables defining the MDP in the case of controlling the Flying-V?
4. How does the implemented flight controller perform during nominal flight and in the presence of faults?
   (a) Noting the dynamical properties and possible fault scenarios of the Flying-V, what flight scenarios should be designed to test the performance of the controller?
   (b) What is the nominal performance and degree of fault tolerance of the implemented controller?
   (c) How well does the nominal flight performance of the proposed flight controller compare to other state-of-the-art controllers?
   (d) How are the fault tolerance characteristics of the proposed flight controller compared to other state-of-the-art controllers?

2 Planning and Work Packages

This section presents the high-level plan of the thesis work in subsection 2.1, which is elaborated in subsection 2.2, and presented as Gantt chart at the end of this document

2.1 High-Level Plan

The online-offline hybrid approach to applying RL to flight controllers seems the most promising solution to tackle the challenge of fault tolerance. The "offline" algorithms come in the form of model-free actor-critic methods, which while being relatively sample inefficient to the "online" algorithms, are firstly able to generalize to more control tasks as they learn policies and value function estimates by repeated traversals across the state-action space, and secondly can provide an initialization condition for the "online" algorithms which have been pre-trained and tested rather than from tabular rasa, otherwise requiring online system identification and inconvenient persistent excitations.

The research plan will set a goal to implement and evaluate such an online-offline hybrid controller for attitude control of the Flying-V. The first of the goals would be to implement an augmented IDHP algorithm, this implementation is prioritized as this incremental model-based algorithm is highly adaptive and can thus provide online control, which is expected to have better fault tolerance. The choice of using IDHP is also motivated by literature research showing many examples of using ADP algorithms for fault tolerance over DRL algorithms [8, 9, 10, 11, 12]. Achieving this first goal would yield the academic contribution of this MSc research, by contributing towards the development of the IDHP algorithm. The second sub-goal would be to implement the "offline" learning algorithms namely DSAC; achieving this sub-goal will yield the societal contribution of this research, as an offline-online hybrid algorithm has much-improved flight performance and fault tolerance characteristics than a purely online trained controller [13].

1. Early March to early May, research phase 1:
   Implement, verify, and evaluate the studied algorithms (IDHP, extended IDHP, potentially DSAC-IDHP) on a simple LTI. Document the results this evaluation generates, and used methodologies in the preliminary report.

   Mid-term review submission planned for 5th May, meeting on 20th May.

2. Late May to early July, research phase 2:
   Implement and verify comparison controllers. Compile the Simulink model of Flying-V to a Python-usable form. Evaluate all controllers on this nonlinear model.

   Greenlight review and draft submission are planned for 15th July, meeting on 29th July.

2.2 Work Packages

Table 1: Table of work packages to be completed throughout the technical phase of the present research.

WP.x	WP title	Start dd/mm- End dd/mm	Working Days: x
	Research Phase 1		
WP.2.1	Design nominal & faulty evaluation cases	02/03 - 06/03	Working Days: 3
Contin-gent WP(s): -	Decide on parameters for the flight simulation that will demonstrate a given controller's performance. Example parameters to be considered: duration of flight, onset time of fault, duration of fault, types of fault& disturbances, severity of fault & disturbances... **Relevant Research Questions**: Uses answer of question **1b, 2a,b**. Partly answers question **4a**.		
WP.2.2	Obtain linearized model	02/03 - 07/03	Working Days: 4
Contin-gent WP(s): -	Obtain an LTI model of the Flying-V to use as a training environment for algorithms. Will need to decide on what kind of LTI to use, for ideal relatability to the nonlinear model in research phase 2, probably use lateral LTI, which could be reduced to a dutch-roll or spiral or ... LTI. However, in research phase 2, even though in one of the meetings I mentioned that the focus should be on attitude control, it might be useful to use a 6-DOF nonlinear model of the Flying-V and not be restrictive to only an attitude model.		
WP.2.3	Implement & verify IDHP on linear model	08/03 - 17/03	Working Days: 6

Contingent WP(s): WP.2.2	Implement IDHP in Python, should most definitely take a lot of inspiration/help from past implementations. Already found the implementations from Casper & Lucas & Stefan. Verify that the implemented IDHP by comparing the kernel values with past implementations, considered verified if kernel learning curves are largely similar in shape and convergence rate. Implementation of IDHP will also partly decide the MDP of the Flying-V, which should be done such that interface with the linear and nonlinear model are identical, to eliminate work for tailoring IDHP to nonlinear model. **Relevant Research Questions**: Uses answer from question **1a,c**, adaptive nature of IDHP makes it more fault-tolerant, IDHP is superior to IHDP, and simpler to implement than IGDHP. Partly answers **3b**.		
WP.2.4	Implement & verify MsIDHP on linearmodel	18/03 - 29/03	Working Days: 10
Contingent WP(s): WP.2.2, WP2.3	Implement the Multi-step extension of IDHP. This work package will be more difficult than implementing IDHP, first of all, no previous implementations exist, secondly, it is unknown if Multi-step allows IDHP to converge or if it causes divergence. Nonetheless, the implemented MsIDHP should be verified by comparison of implemented schemes to analytically derived schemes. Verification by performance is also tricky since poor performance does not necessarily indicate erroneous implementation, could also indicate that Multi-step IDHP is worse than vanilla IDHP. **Relevant Research Questions**: Uses answer from question **1a**, improving RL algorithms with multi-step is a proven idea but novel when applied to IDHP hence is “state-of-the-art”.		
WP.2.5	Implement & verify EtIDHP on linearmodel	30/03 - 10/04	Working Days: 8
Contingent WP(s): WP.2.2, WP2.3	Implement the Eligibility trace extension of IDHP. The same comments from WP.2.4 apply here. In case the MsIDHP idea does not work, the EtIDHP idea would be tried instead since it would be interesting to produce an augmented IDHP, i.e. this package will be executed instead of WP.2.6 & WP.2.7. This could be risky, as EtIDHP is also not guaranteed to produce meaningful results, meaning time spent on WP.2.4 & WP.2.5 would have been wasted. This package could be replaced with WP.2.6 + WP.2.7, which means the result of the thesis would be spread across augmenting IDHP and the Online-Offline hybrid controller. **Relevant Research Questions**: Uses answer from question **1a**, improving RL algorithms with eligibility-trace is a proven idea but novel when applied to IDHP hence is “state-of-the-art”.		
WP.2.6	Implement & verify DSAC on linearmodel	30/03 - 06/04	Working Days: 5
Contingent WP(s): WP.2.2	DSAC would be implemented in Python, with guidance from previous implementations. Verification will be done by comparing learning curves of implemented DSAC vs baseline DSAC versions from Lucas/Peter. This WP and WP.2.7 are part of the offline-online direction of research. Which could in itself serve as the focus of this thesis, replacing WP.2.3 & WP.2.4.		
WP.2.7	Integrate DSAC and IDHP implementations	07/04 - 12/04	Working Days: 5

Contingent WP(s): WP.2.2, WP.2.3, WP.2.6	The trained actor network from DSAC will need to be used for IDHP, which requires the implementation of IDHP to be changed. **Relevant Research Questions**: Uses answer from question **1a,c,d**, an online-offline hybrid provides the fault tolerance of IDHP and the tracking performance of DSAC.		
WP.2.8	Write preliminary results chapters	13/04 - 03/05	Working Days: 15
Contingent WP(s): WP.2.8.1-2.8.3	Writing on thesis report is planned to commence after all technical work is done, but does not need to start strictly after technical work finishes. This work package is broken down for progress tracking.		
WP.2.8.1	Run implementations on evaluation cases	13/04 - 19/04	Working Days: 5
Contingent WP(s): WP.2.1-2.7	The implemented algorithms are evaluated on the evaluation cases designed in WP.2.1, a large number of runs will be conducted to ensure data reliability (Monte Carlo runs).		
WP.2.8.2	Compile and document performances	20/04 - 24/04	Working Days: 3
Contingent WP(s): WP.2.8.1	The results of evaluation cases are processed according to the criteria and metrics chosen to be analyzed in this work package. Write the results & discussion chapters. **Relevant Research Questions**: Answers question **2c** by choosing the criteria and metrics, partly answers **4b**.		
WP.2.8.3	Write accompanying texts	25/04 - 03/05	Working Days: 7
Contingent WP(s): -	Writing the introduction, methodology, derivations, and conclusion chapters. Making illustrative diagrams where suitable.		
Research Phase 2			
WP.3.1	Design nominal & faulty evaluation cases	21/05 - 24/05	Working Days: 4
Contingent WP(s): -	Decide on parameters for the flight simulation that will demonstrate a given controller's performance. Example parameters to be considered: duration of flight, onset time of fault, duration of fault, types of fault& disturbances, severity of fault & disturbances... **Relevant Research Questions**: Uses answers for questions **1b, 2a,b**. Answers question **4a**.		
WP.3.2	Obtain nonlinear model	21/05 - 27/05	Working Days: 5
Contingent WP(s): -	Compile the Simulink model of Flying-V into a form usable in Python. Use the CitAst manual for guidance, and online resources. Minimally, the model should contain attitude dynamics, possible to use full 6-DOF simulation as well. The model should be verified to be correct by comparing it's uncontrolled & controlled dynamics with the Simulink version. The interface of the model will decide largely the MDP of the Flying-V and requires considering how the RL algorithms were implemented. **Relevant Research Questions**: Answers question **4a**.		
WP.3.3	Implement/integrate comparison controllers with nonlinear model	28/05 - 06/06	Working Days: 8

Contin-gent WP(s): WP.3.2	The comparison controllers that will be used are planned to be INDI and PID, however, the choice will only be made concrete when this WP commences. Make sure the controller's interface with the nonlinear model is identical to the RL controllers if done in Python. Alternatively, this could be done in Matlab, where existing INDI implementations exist, speeding up work. **Relevant Research Questions**: Partly answers question **4c,d**, by choosing the comparison controllers.		
WP.3.4	Write scientific article and additional results chapters	07/06 - 08/07	Working Days: 22
Contin-gent WP(s): WP.3.4.1-3.4.3	Work carried out on the final document for the thesis.		
WP.3.4.1	Run all controllers on evaluation cases	07/06 - 12/06	Working Days: 4
Contin-gent WP(s): WP.2.4-2.7, WP.3.1-3.3	Retrieve data to evaluate controllers, if contingent WP's are done well this WP should be largely autonomous.		
WP.3.4.2	Compile and document performances	13/06 - 17/06	Working Days: 3
Contin-gent WP(s): WP.3.4.1	The results of evaluation cases are processed according to the criteria and metrics chosen to be analyzed in this work package. Write the results & discussion chapters. **Relevant Research Questions**: Answers question **4b,c,d**.		
WP.3.4.3	Write accompanying texts	18/06 - 08/07	Working Days: 15
Contin-gent WP(s):	Write the accompanying text of the scientific article (intro, method, conclusion ...), additional results and conclusion of thesis report, recommendations for future work.		

References

[1] Ian Savage. "Comparing the fatality risks in United States transportation across modes and over time". In: *Research in Transportation Economics* 43.1 (2013). The Economics of Transportation Safety, pp. 9–22. ISSN: 0739-8859. DOI: `https://doi.org/10.1016/j.retrec.2012.12.011`.

[2] International Air Transport Association. *Loss of Control In-Flight Accident Analysis Report*. Distributed by IATA. 2019.

[3] Killian Dally and Erik-Jan Van Kampen. "Soft Actor-Critic Deep Reinforcement Learning for Fault Tolerant Flight Control". In: *AIAA SCITECH 2022 Forum*. American Institute of Aeronautics and Astronautics, Jan. 2022. DOI: `10.2514/6.2022-2078`. URL: `http://dx.doi.org/10.2514/6.2022-2078`.

[4] Stefan Heyer, Dave Kroezen, and Erik-Jan Van Kampen. "Online adaptive incremental reinforcement learning flight control for a CS-25 class aircraft". In: *AIAA Scitech 2020 Forum*. 2020, p. 1844.

[5] D.S. Lee, D.W. Fahey, A. Skowron, M.R. Allen, U. Burkhardt, Q. Chen, S.J. Doherty, S. Freeman, P.M. Forster, J. Fuglestvedt, A. Gettelman, R.R. De León, L.L. Lim, M.T. Lund, R.J. Millar, B. Owen, J.E. Penner, G. Pitari, M.J. Prather, R. Sausen, and L.J. Wilcox.

"The contribution of global aviation to anthropogenic climate forcing for 2000 to 2018". In: *Atmospheric Environment* 244 (2021), p. 117834. ISSN: 1352-2310. DOI: `https://doi.org/10.1016/j.atmosenv.2020.117834`.

[6] M Klöwer, M R Allen, D S Lee, S R Proud, L Gallagher, and A Skowron. "Quantifying aviation's contribution to global warming". In: *Environmental Research Letters* 16.10 (Nov. 2021), p. 104027. DOI: `10.1088/1748-9326/ac286e`. URL: `https://dx.doi.org/10.1088/1748-9326/ac286e`.

[7] Gary J Balas. "Flight control law design: An industry perspective". In: *European Journal of Control* 9.2-3 (2003), pp. 207–226.

[8] C Treesatayapun. "Fault-tolerant control based on reinforcement learning and sliding event-triggered mechanism for a class of unknown discrete-time systems". In: *Nonlinear Analysis: Hybrid Systems* 50 (2023), p. 101381.

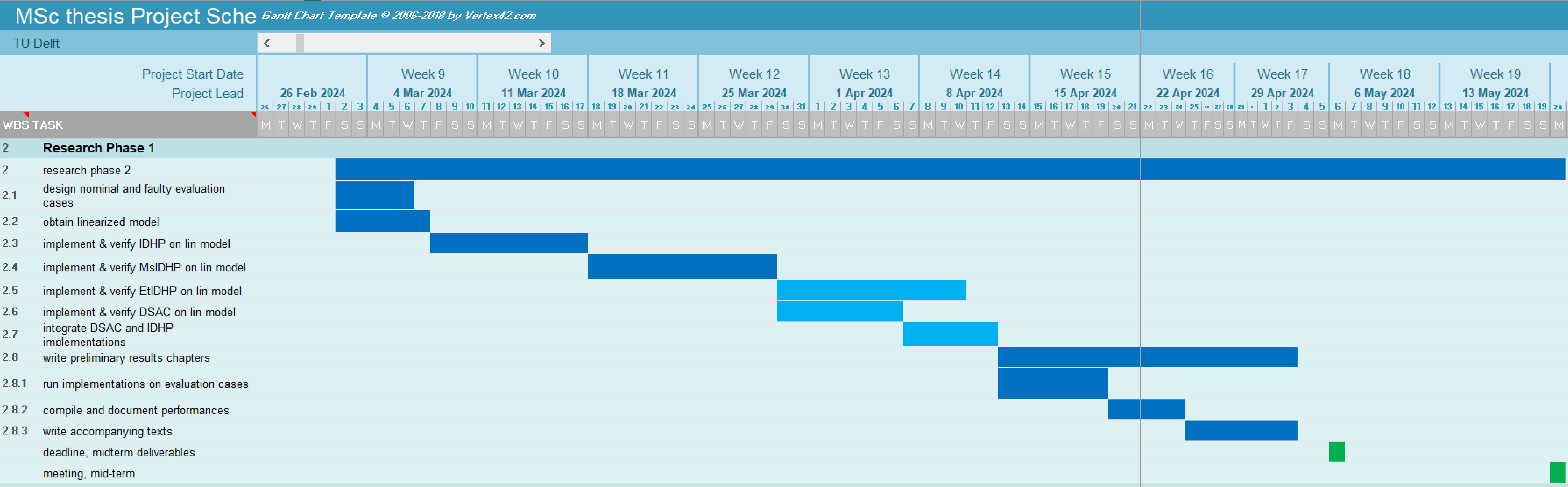
[9] Wanbing Zhao, Hao Liu, and Frank L Lewis. "Fault-Tolerant Control for the Formation of Multiple Unknown Nonlinear Quadrotors via Reinforcement Learning". In: *IFAC-PapersOnLine* 53.2 (2020), pp. 2465–2470.

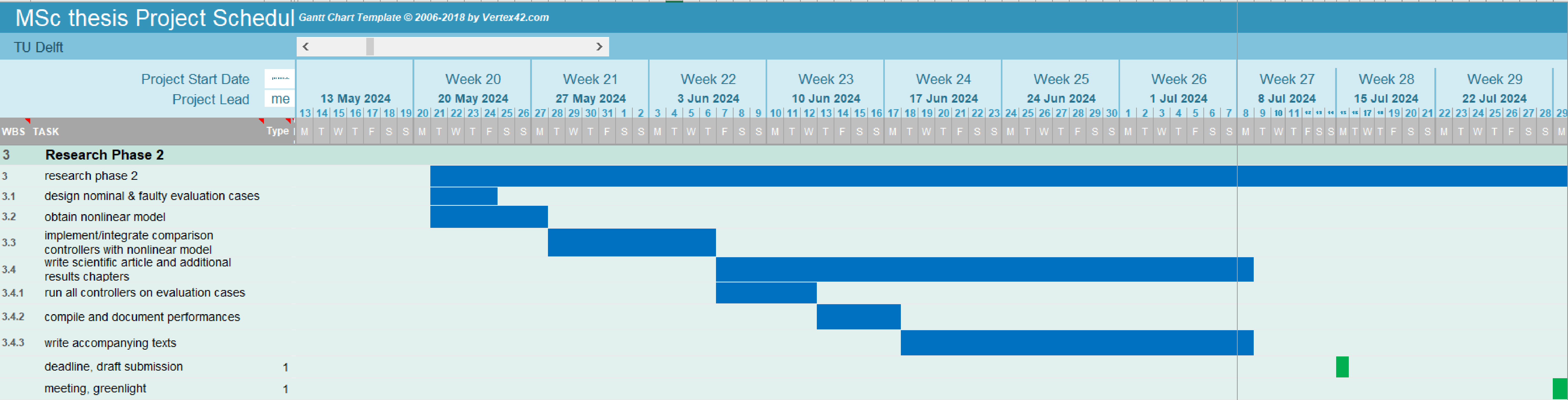
[10] Bo Zhao, Derong Liu, and Yuanchun Li. "Observer based adaptive dynamic programming for fault tolerant control of a class of nonlinear systems". In: *Information Sciences* 384 (2017), pp. 21–33.

[11] Kun Zhang, Huaguang Zhang, Zhiyun Gao, and Hanguang Su. "Online adaptive policy iteration based fault-tolerant control algorithm for continuous-time nonlinear tracking systems with actuator failures". In: *Journal of the Franklin Institute* 355.15 (2018), pp. 6947–6968.

[12] Stefan Heyer, Dave Kroezen, and Erik-Jan Van Kampen. "Online adaptive incremental reinforcement learning flight control for a CS-25 class aircraft". In: *AIAA Scitech 2020 Forum*. 2020, p. 1844.

[13] Casper Teirlinck and Erik-Jan Van Kampen. "Hybrid Soft Actor-Critic and Incremental Dual Heuristic Programming Reinforcement Learning for Fault-Tolerant Flight Control". In: *AIAA SCITECH 2024 Forum*. 2024, p. 2406.


MSc thesis Project Sche
Gantt Chart Template © 2006-2018 by Vertex42.com
TU Delft
Project Start Date
Project Lead
26 Feb 2024
Week 9
4 Mar 2024
Week 10
11 Mar 2024
Week 11
18 Mar 2024
Week 12
25 Mar 2024
Week 13
1 Apr 2024
Week 14
8 Apr 2024
Week 15
15 Apr 2024
Week 16
22 Apr 2024
Week 17
29 Apr 2024
Week 18
6 May 2024
Week 19
13 May 2024
WBS TASK
2 Research Phase 1
2 research phase 2
2.1 design nominal and faulty evaluation cases
2.2 obtain linearized model
2.3 implement & verify IDHP on lin model
2.4 implement & verify MsIDHP on lin model
2.5 implement & verify EtIDHP on lin model
2.6 implement & verify DSAC on lin model
2.7 integrate DSAC and IDHP implementations
2.8 write preliminary results chapters
2.8.1 run implementations on evaluation cases
2.8.2 compile and document performances
2.8.3 write accompanying texts
deadline, midterm deliverables
meeting, mid-term


MSc thesis Project Schedul
Gantt Chart Template © 2006-2018 by Vertex42.com
TU Delft
Project Start Date
Project Lead
me
Week 20
Week 21
Week 22
Week 23
Week 24
Week 25
Week 26
Week 27
Week 28
Week 29
13 May 2024
20 May 2024
27 May 2024
3 Jun 2024
10 Jun 2024
17 Jun 2024
24 Jun 2024
1 Jul 2024
8 Jul 2024
15 Jul 2024
22 Jul 2024
WBS
TASK
Type
3 Research Phase 2
3 research phase 2
3.1 design nominal & faulty evaluation cases
3.2 obtain nonlinear model
3.3 implement/integrate comparison controllers with nonlinear model
3.4 write scientific article and additional results chapters
3.4.1 run all controllers on evaluation cases
3.4.2 compile and document performances
3.4.3 write accompanying texts
deadline, draft submission 1
meeting, greenlight 1