10th INTERNATIONAL SYSTEMS & CONCURRENT ENGINEERING FOR SPACE APPLICATIONS CONFERENCE (SECESA 2022)

5 - 7 OCTOBER 2022

BOTTOM-UP AI-SUPPORT TO GENERATE CONCEPTUAL DESIGNS FOR CONCURRENT ENGINEERING STUDIES WITH DRL

Jan-Peter Ceglarek(1), Reinhold Bertrand(2), Redouane Boumghar(3)

(1)Technical University Darmstadt, Parametry.ai
Präsidium der Technischen Universität Darmstadt
64277 Darmstadt
Email: ceglarek@fsr.tu-darmstadt.de

(2)European Space Agency, Technical University Darmstadt
Robert-Bosch-Str. 5, Darmstadt 64293, Germany
Email: reinhold.bertrand@esa.int

(3)Parametry.ai
Frankfurt am Main, Germany
Email: red@parametry.ai

INTRODUCTION

In today's era of New Space, efficient mission design is more important than ever. Novel challenges, like the vastly growing market of Commercial off-the-shelf (COTS) components and elaborating mission ideas or business cases that remains unexplored, make the design process more and more challenging. Additionally, habits, risk aversion and efforts optimization push known and proven solutions to be preferred over the exploration of new ideas - therefore decreasing the chance of potential innovations. This could be avoided, if reliable information about components and their feasible combinations would be available, to lower the technical risk, or reduce mission costs, to minimise the financial risk - therefore promoting flight-novel components and integrations.
To improve performance, cost- and time-efficiency during the preliminary design phase of new space missions, ESA, DLR and other space mission providers or universities like the TU Darmstadt implemented dedicated facilities to conduct concurrent engineering (CE) studies. In iterative sessions a mission specific team of domain experts elaborate on a possible implementation for a novel mission over the course of one to several weeks.
While previous research in the field of AI-supported CE mainly focused on utilising extracted knowledge from historical mission designs, exploring the possibilities of an bottom-up approach was seen as a promising approach: generating mission designs based on established engineering calculations and a database of components.
In the context of a master thesis such an Artificial Intelligence (AI) tool was developed as a part of the ESALab@TU Darmstadt Concurrent Engineering Lab (CELab). The Deep Reinforcement Learning Concept Creator (DCC) can generate a CubeSat design tailored to a set of requirements, based on a database of available COTS components.

The implemented system uses Deep Reinforcement Learning (DRL), which means, it's learning strategy is Reinforcement Learning (RL) with an additional Neural Network (NN). RL gets its name from the feedback that reinforces a taken decision performed by the agent. The AI agent gains knowledge by taking *actions* in purpose-built environment, given a current *state* and observing the *environment's* reaction (*reward*). From the delta of the expected and observed behaviour, the agent can adapt its strategy and improves its performance during its training. Traditionally originating from a robotic context, the RL environment can contain simple numeric responses, but also more complex engineering calculations.
DRL is especially suited to build a bottom-up system generation approach with, since the existence of a controllable learning environment allows it to asses mission designs without the need of historical missions and to learn from its learning process. It could be shown [1], that RL is generally suited for concept generation, as intended by this work. It was further stated, that DRL (especially DQN) has significant advantages over the used RL method (Q-learning).

RELATED WORK

Artificial Intelligence (AI) has proven its value in the past to offer capable solutions and is especially reliable in data-driven situations. Besides many other approaches to support the space mission design, RL was shown to be capable of

generating CubeSat designs [1] where the decisions were just relying on regular engineering calculations. The difference to other projects in the field of AI-supported CE lies in the general approach of the system on a fundamental level. While the demonstrated work follows the aforementioned "buttom-up" approach, the remaining research to this topic followed a more "top-down" approach, using historical mission data.
Berquand et al. [2] proposed an AI system to support CE studies by providing a knowledge base of historical missions, with the idea to "provide easy and quick access to previous design decisions". As a key challenge of their work, the challenge of extracting relevant information from technical documentations, data sheets and alike is highlighted.
The proposed system would help the designing team with knowledge from previous missions using Natural Language Processing (NLP), which is a different approach, than generating designs based on engineering calculations and the experience of the designing CE team.
Buettner et al. [3] developed the iCASD tool which is capable of creating satellite designs. The work primarily focuses on component placement within the satellite, but also offers simple component selection capabilities. The iCASD project is a capable visualisation and component positioning tool, but the proposed PhD research offers increased component selection functionalities and integration in an existing MBSE/CE workflow.
In addition to the machine learning (ML) discussed here, Expert Systems have been studied as another branch of AI to support the CE process, like [4].
In summary, AI has been explored to be used for CE studies before, but the main focus relied on extracting information from historical missions, using Experts Systems or NLP to support the designing teams. The proposed research uses a method to generate systems without the challenge to analyse the documents, mails and technical descriptions of other missions. The feasibility of this approach has been first demonstrated for regular RL [1], but the usability of DRL had yet to been demonstrated.
Additionally, none of the work were made open source, so that future development on the topic of AI-supported concept generation could be ensured.

DESIGN OVERIVEW

It was the general idea of this work to create a self-taught "design expert", that is very skilled in the use of a given component database. This tool can then be used to create capable system designs with respect to a given set of mission requirements. By relying on engineering calculations, the AI was tasked to learn its designing strategy, in contrast to relying on extracted knowledge from historical mission concepts.

The DRL Concept Creator (DCC) is an AI software tool, that uses Deep Reinforcement Learning (DRL) to build satellite concepts with real Commercial-Off-The-Shelf (COTS) parts to support the CE process.
Fig 1 illustrates the underlying functionality and integration into a CE workflow. Like for any normal CE studies, initially first, rough mission requirements have to be formalised. From there on the AI selects COTS components from a database and combines it to a system concept, which is represented as a list of selected components. This concept will then be examined by the CE team and can be used for inspiration, training purposes or just a test implementation to check, how the current system design could be realised using real-world components.
The CE team has full control about the criteria and mechanisms the AI uses to comprise the system design and can adjust them by simply changing the implemented calculations. Further, when moving to the next design iteration, the mission requirements can be adjusted to represented the current state of mission design, which will be used by the tool to create an updated mission design.

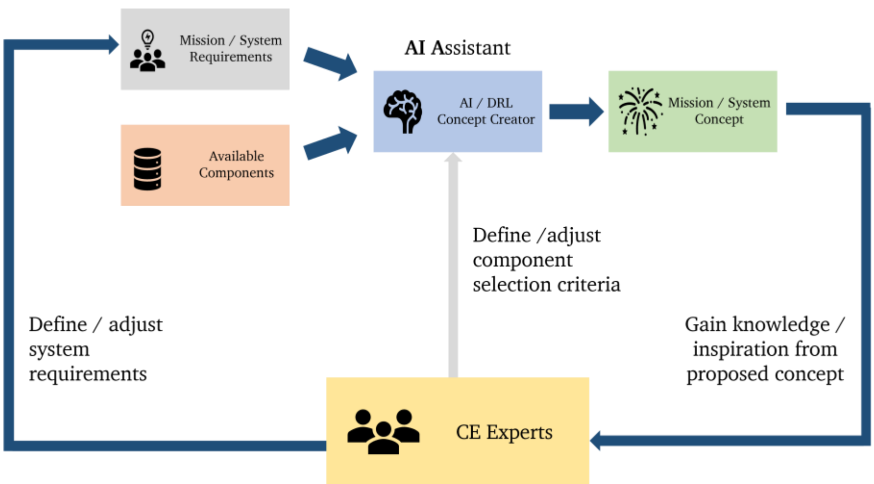

Figure 1: Schematic overview of the CE integration and working principle.

This back and forth, between the concept creation tool and the designing team, was designed to work in an iterative manner together with any regular CE process. Furthermore, the DCC tool is intended to accompany the engineering team to enhance its capabilities and not replacing it.

TECHNICAL IMPLEMENTATION

For the implementation of the DCC, three main aspects had to be designed: a tool to create the component database, the AI agent that learns the task of finding a suitable component set and the learning environment for the agent to gain experience in .
The following components of the software were built with high modularity in mind. It is therefore possible to change the component database, agent learning algorithm and environment implementation with little effort thanks to easy interfaces.

SatSearch.co is a web shop which offers a variety of COTS space equipment. Because of its clear hierarchical organisation of components and open declaration of component specification, it was selected as the source for the component database. Using the HTML parser Beautiful Soup 4, the complete SatSearch web shop were scrapped for all available components and their specifications. These information were then compiled into one CSV file and every component were given a unique ID and tags based on their SatSearch (sub)categories.

For the AI agent, Duelling Deep Q-Networks (DDQN) were chosen as the DRL algorithm because of its proven learning effectiveness [5] and exhaustive online documentation [6]. Additionally, it is a progression from the original DQN algorithm, which was recommended previous research on this topic [1]. TensorFlow was used for the implementation of the RL agent and the NN.
As illustrated in Fig. 1, the DCC takes mission requirements and creates a mission design accordingly. Therefore, the RL agent makes a component selection from the component database based on the user-defined system design constraints. The evaluation and optimisation of a design thus is done inside the learning environment: This is achieved by calculating a reward, which is a numerical number characterising the performance of a compiled design variant. The reward functions as the feedback to the design agent who can learn from this experience and adjust its component selection strategy accordingly to be able to gain a higher reward. As a result, the DCC can gradually evolve a design towards an optimum with respect to the implemented component evaluation functions.

Fig. 2 illustrates the generation of the system reward. The agent calculates an individual reward for every single component, based on technical performance calculations in the same way, the domain experts perform their component system evaluations for the CE design activity. Each component reward is based on a variety of general and domain-specific parameters such as power, mass or component-specific parameters. Lastly, the complete system gets evaluated as a on system level, before all rewards get combined into one single system reward.
The RL environment defines, which elements are supported by the DCC system and how each component gets evaluated. The current version of the software supports a reduced CubeSat system consisting of Camera, Reaction Wheel, Transceiver, Solar Panel and Battery. Limited by the 6 month timeframe of the master thesis and the prototype status of the implementation, the influences between components is only modelled by the evaluation for the battery component, which is why the reward calculation for the battery is the most sophisticated, as indicated by Fig. 2. Decisions about which elements to include in the design generation and with how many entities can be configured in a dedicated system architecture file.

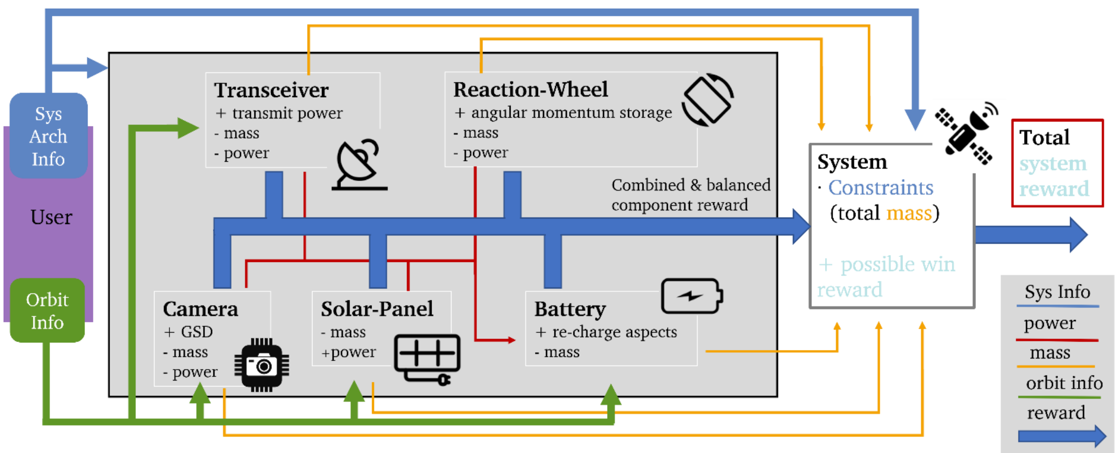

Figure 2: Overview of the complete system reward building with the considered component parameters. "+" indicating a parameter where a higher value is better and vice versa for "-"..

Just like in traditional CE studies, the tool takes propagating effects of the different components and system configurations throughout the entire design into account, although in the current prototype version, only in a basic form. Following the aforementioned modular design approach, the software is designed with other systems in mind, so that it can be easily extended to support more satellite components and used for entirely different systems as well, like the Ground Segment.

The outcome of the system generation process is shown in Tab1. Based on the implemented evaluation functions on component and system level, the tool selected the listed components from the component database.

Table 1: Complete 5 component system generated by the AI-powered DCC tool.

	Reaction Wheel	Transceiver	Camera	Solar Panel	Battery
URL	Link	Link	Link	Link	Link
Mass	21g	72g	400g	570g	125g
Consumed Power	< 0.8 W	0.18 – 3.1 W	2.6 – 4.5 W	-19.2 W	-
Bus Voltage	3.5 – 11V	12 V	4.5 – 5V	19.2 V	3.7 V
Battery Capacity					6000 mAh
GSD			39 m @ 500km		
Transmit Power		< 33 dBm			
Angular Momentum Storage	-1.5 – 1.5 mNms				
Total System Mass	1188g				
Total max System Power Consumption	8.4 W				

VALIDATION AND DISCISSION

The AI system were designed, to learn to design a satellite system given a component database and a set of constraints. To validate the success, three aspects were tested: the capability of the implement system to generate valid CubeSat designs, if the generated concepts are optimal solutions and how robust the concepts creation process is.

In general, ML applications are designed to learn a certain skill, which is hard to define correctly. The agent may be able to learn to maximise its reward function, but this does not imply this yields the intended result. Setting the correct incentives is not always obvious and it is therefore not sufficient to verify, that the system is able to reach a stable learning process (showcasing that it learned something). Instead, specific tests have to be conducted.
Validating the concept creation of the developed system was consequently done in two ways: the system creation containing 1 to 5 components were tested against the best possible component combination found by a combinatoric "brute-force" search and real-world CubeSat missions. Comparison with real missions is however just an estimation, since the DCC can only build a system with the available components from the SatSearch web shop, while real missions do not have the same limitation and can even design their own equipment.
Additionally, the trained model was tested regarding its robustness. During training, the AI agent learns its component selection strategy with regard to a specific set of constraints. Testing the trained model with altered constraints shows how well the AI can abstract from its learned scenario.

Validation showed, that the system generally works as intended. The system is able to learn a strategy for an arbitrary component amount (1 to 5 components tested in various combinations), independent of the CubeSat system configuration. Secondly, it could be demonstrated that the AI agent is not only capable of improving its component selection during its learning process, but also, that the learned strategy can lead to the optimal result.
Compared with real-world missions, the DCC tool works best for 1U CubeSat designs. This is explained by its reduced set of considered parameters and it therefore works best for systems with low complexity.
A comparison with various real-world CubeSat missions showed, that the generated designs are lighter and consume less power. This is to be expected, since the DCC tool currently does not support components like thermal isolation, structure frames or cables and connectors.
The current implementation of the DCC takes only mission architecture parameters into account. Although orbit parameters influence the evaluation of the components through parameters like the Ground Sampling Distance for the camera, the resulting trained models are not optimised for altered orbit parameters.
To address this, future research on this topic will introducing the orbit parameters as input influences to the NN alongside the system architecture information.

Table 2: Overview of the comparison between the generated designs and real-world CubeSat mission. The delta indicates the difference from the real mission to the created design: -22%: the DCC design is 22% lighter, than the real one.

Comparison Mission	Orbit Information	Included Components	Mass Delta Power Delta
OPS-Sat (3U)	a = 6871 km i = 97.47°	Reaction Wheel, Transceiver, Camera, Solar Panel, Battery	-22% -44%
UPSat (3U)	a = 6771 km i= 51.6°	Transceiver, Solar Panel, Battery	-80% -65%
EQUiSat (1U)	a = 6871 km i = 97.47°	Transceiver, Solar Panel, Battery	-38% -54%
BOBCat (3U)	a = 6871 km i = 97.47°	Transceiver, Camera, Solar Panel, Battery	-67% -44%

CONCLUSION

In context of a 6 month master thesis conducted at the CELab, TU Darmstadt, an AI system capable of CubeSat concept generation has been developed based on Deep Reinforcement Learning (DRL) .The DRL Concept Creation (DCC) tool selects suitable satellite components from a database of COTS components fitted to a set of defined system/mission requirements, just be evaluating sets of components.
With this, it could be shown that DRL can be used to build a satellite design where the decision process is based on engineering calculations utilising DRL. It was possible to select suitable components and to compose a satellite design automatically, satisfying formalised design constraints. This process resembled the trade-off studies performed during Concurrent Engineering (CE) studies, however performed by making use of AI. Feasibility of the followed system creation approach were verified with real-world CubeSat missions.
The modularity of the DCC's software design allows a simple expansion to support more components and systems, as well as adjustment to the implemented component evaluations.

The DCC tool is currently a prototype, just supporting a reduced CubeSat architecture
Future research on this topic will be conducted in a PhD research in the context of the AI for Concurrent Engineering (AI4CE) project. The focus will be on further expanding system creation and requirement definition capabilities, a tight integration into the MBSE workflow, as well as a way to introduce knowledge gained by the operating teams directly into the mission design process.
The software has been made open source under the gplv3 license: https://gitlab.com/jan-peter/drl-concept-creator

References

[1] Krijnen, Bas. "Assessment of Reinforcement Learning for CubeSat concept generation." (2020).
[2] Berquand, Audrey, et al. "Artificial intelligence for the early design phases of space missions." 2019 IEEE Aerospace Conference. IEEE, 2019.
[3] Buettner, Timothee, et al. "The intelligent Computer Aided Satellite Designer iCASD-Creating viable configurations for modular satellites." 2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS). IEEE, 2018.
[4] Murdaca, F., et al. "Artificial intelligence for early design of space missions in support of concurrent engineering sessions." 8th International Systems & Concurrent Engineering for Space Applications Conference. 2018.
[5] Hessel, Matteo, et al. "Rainbow: Combining improvements in deep reinforcement learning." Thirty-second AAAI conference on artificial intelligence. 2018.
[6] https://pythonprogramming.net/deep-q-learning-dqn-reinforcement-learning-python-tutorial/