Fachhochschule Dortmund
University of Applied Sciences and Arts
Embedded Systems Engineering

Title of your thesis

Master Thesis

Fachhochschule Dortmund
University of Applied Sciences and Arts

Author: Sheikh Muhammad Adib Bin Sh Abu Bakar
Matriculation Number: 7219310
Supervisor: Prof. Dr. rer. nat. Stefan Henkler
Co-Supervisor: You Co-Supervisor(s)
Date: May 27, 2025

Abstract

Your Abstract

Declaration

I, Sheikh Muhammad Adib Bin Sh Abu Bakar, hereby confirm, that I have written the Master Thesis at hand independently – in case of a group work: my respectively designated part of the work -, that I have not used any sources or materials other than those stated, and that I have highlighted any citations properly. .
Dortmund, May 27, 2025

Sheikh Muhammad Adib Bin Sh Abu Bakar

Contents

List of Figures

List of Tables

1

Motivation

1 General motivation

- QUESTION (scope): ”How to develop a self-adaptive system for distributed CPS?”
- Modern systems often operate in unpredictable and changing environments
- Rule-Based Adaptation :
  - Limited Flexibility
  - exhaustive offline tuning (Time-Consuming)
- Online DSE can dynamically allocate and reallocate hardware resources (for distributed CPS):
  - Makes decisions based on the current system state and goals
  - self-managing systems in uncertain or evolving environments: optimise application and hardware
  - Optimise latency, throughput, or energy efficiency
  - Adapt to changing workloads or resource availability (scheduling):
    * Task mapping (which task runs where)

    * Voltage/frequency scaling (DVFS)
    * deadline satisfaction

- QUESTION (focus): " How can online Design Space Exploration (DSE) be utilised to optimise hardware utilisation and software execution to enable self-adaptive computing systems? "

- QUESTION (focus): " How to design a lightweight model but accurate to reduce complexity? "

2 Research gap/Problem definition

- lack of standard self-adaptive definition

- modelling complex dCPS to have accurate models for DSE

- weather forecast dependent configuration for dCPS

3 Contribution

- leverage machine learning to improve models over time?

- weather dependent UAVs fleet configuration (scheduling)

4 Paper Flow

2

Related Work

3

Preliminaries

1 System

2 Self-adaptive System

In the literature, there is no standard definition for self-adaptive CPS. [][1] discussed in detail about self-adaptive CPS definition while ignoring the broader view of CPS.

Complex and self-adaptable systems need to be autonomous, system- and environment-aware, and self-controlled [][2]

[1]https://mediatum.ub.tum.de/doc/1692104/08vhwc8hsj1oacb0165por366.petrovska-ana.pdf
[2]A Review of the Principles of Designing Smart Cyber-Physical Systems for Run-Time Adaptation

4

Approach

1 Methodology

DSE-M [][1]

This section discusses the method used to find a feasible solution for a self-adaptive distributed cyber-physical system. To structure the support system in realising a self-adaptive system, a model-based system engineering (MBSE) approach is used with a strong emphasis on using SysML (Systems Modelling Language) as the primary modelling tool and practised with the guidance and standards promoted by INCOSE.

The approach is to start with a comprehension of the self-adaptive system as explained in chapter 3. To enable self-adaptive capability, many approaches have been proposed in the literature. One of the most popular approaches is to adapt the concept of MAPE-K (Model, Analysis, Plan, Execute and Knowledge) from [**??**][2] as visualised in figure 4.1. This approach could be seen as an extension to a closed-loop control system

[1] Design Space Exploration in Robotics
[2] `https://ieeexplore.ieee.org/document/7194653`

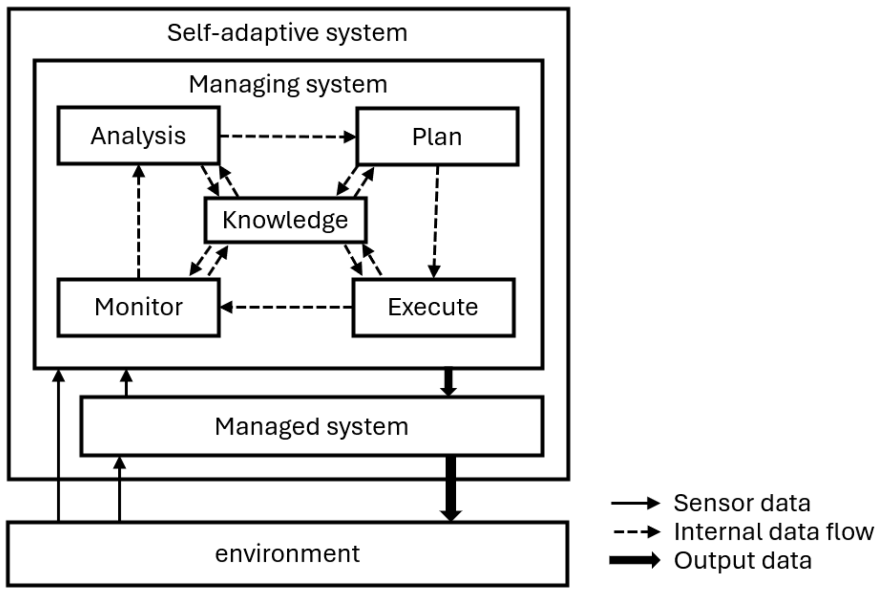

Figure 4.1: MAPE-K concept

< ... The explanation of the MAPE-K ... >

The influence of the managing system, or known as the self-adaptive component in this paper to the managed system and its dependency on the environment is visualised in Figure 4.2

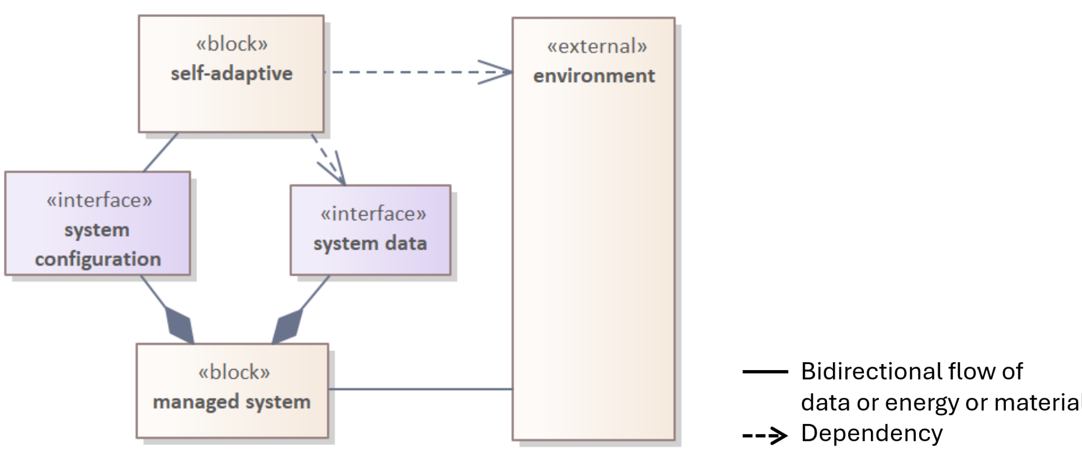

Figure 4.2: Approach

The detail of the implementation of the MAPE-K concept within the self-adaptive component is depicted in Figure 4.3

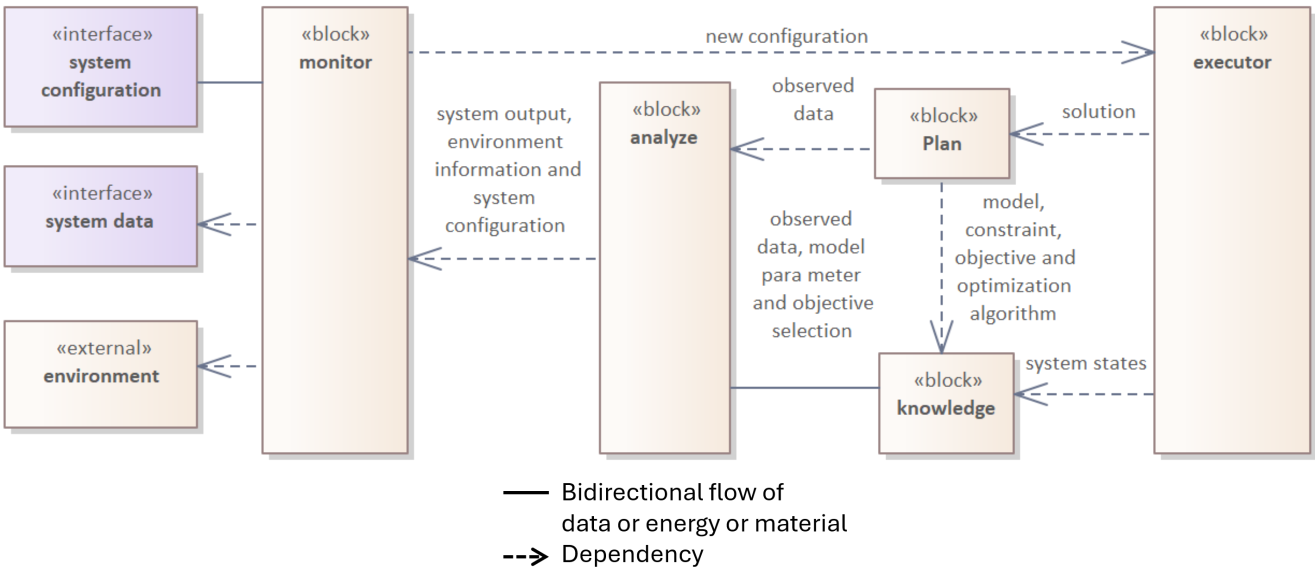

Figure 4.3: MAPE-k adaptation within self-adaptive component

< ... The explanation of the diagram 4.3 ... >

The monitor is responsible for reading and updating both the managed system configuration, application and hardware configuration. The observed data of the managed system by the monitor is used to select the appropriate objective and improve the model used for design space exploration via a machine learning algorithm, so that the model becomes better over time. Figure 4.4 shows that these processes are executed within the analysis block.

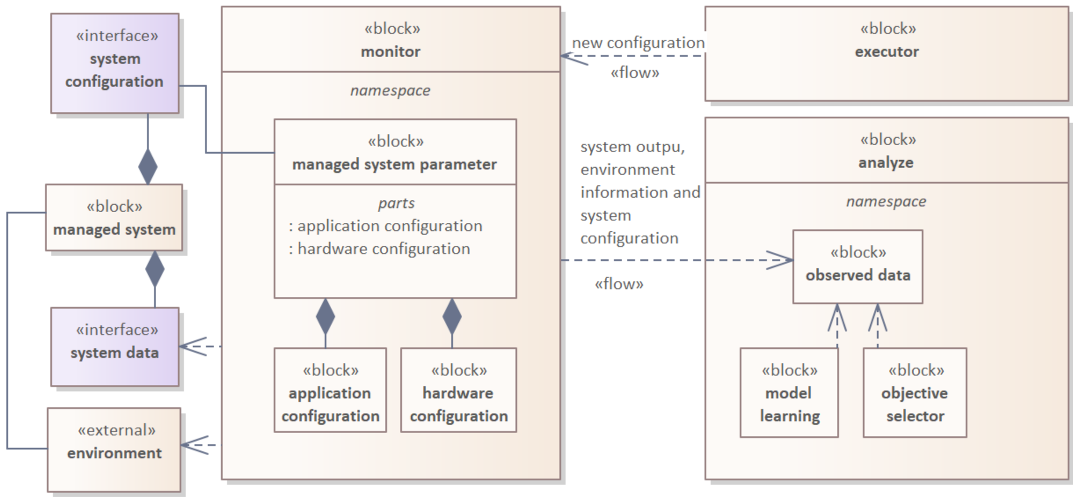

Figure 4.4: Monitor input and output

Figure 4.5 shows the internal block within the MAPE-K blocks and their dependency

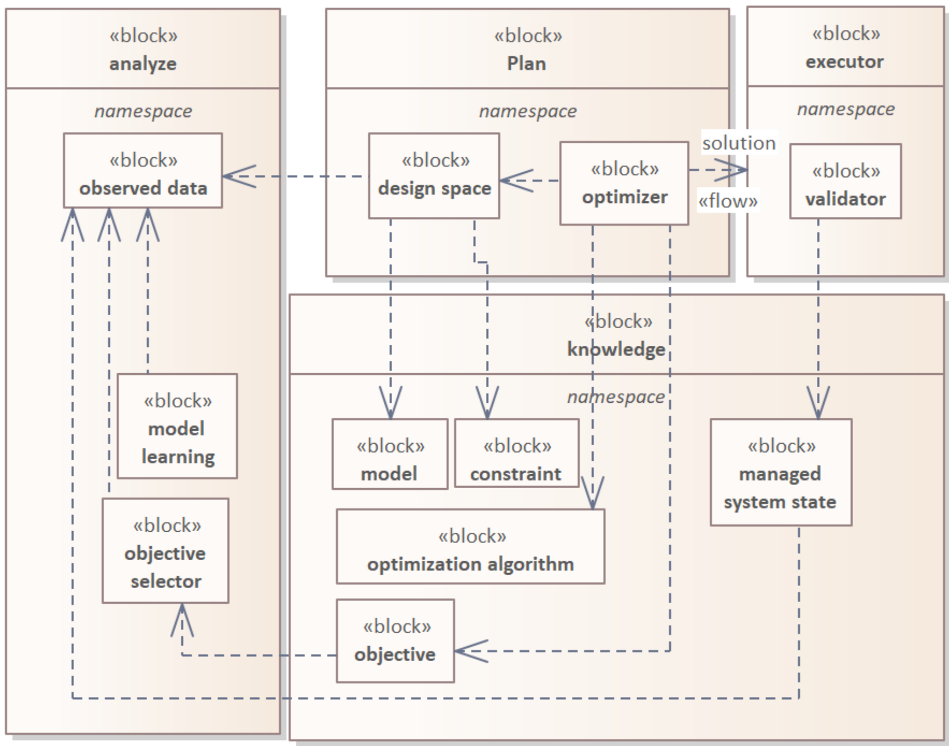

Figure 4.5: MAPE-K internal block

In the next section, the case study will be explained, which will be used as an example of a managed system and to show how the whole system will work together.

2 Case study

The theme of this case study is a smart farming system, which encompasses multiple interconnected subsystems. Given the complexity of such a system, this paper focuses on a select few subsystems that are most relevant to the requirements outlined in Figure 4.6. These requirements are specifically tailored to address the key interests and objectives of this study.

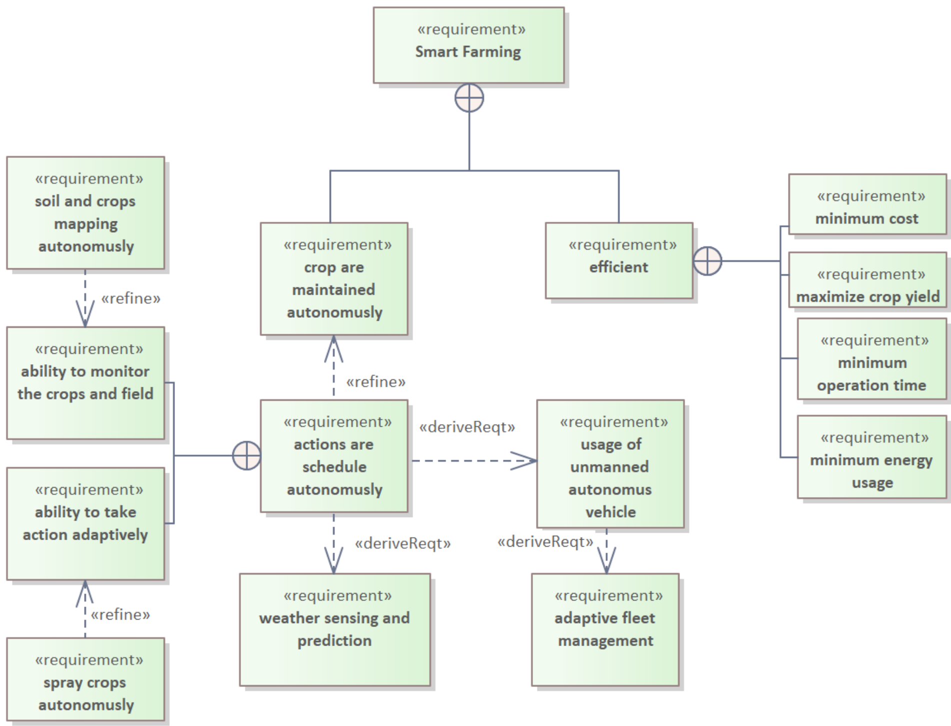

Figure 4.6: Precision farming requirements

From the defined requirements diagram, it can be seen that the system has multiple functionalities, whereas its configuration is adaptively changing depending on the weather. The main functional requirements can be simplified as shown in Figure 4.7 to achieve certain goals

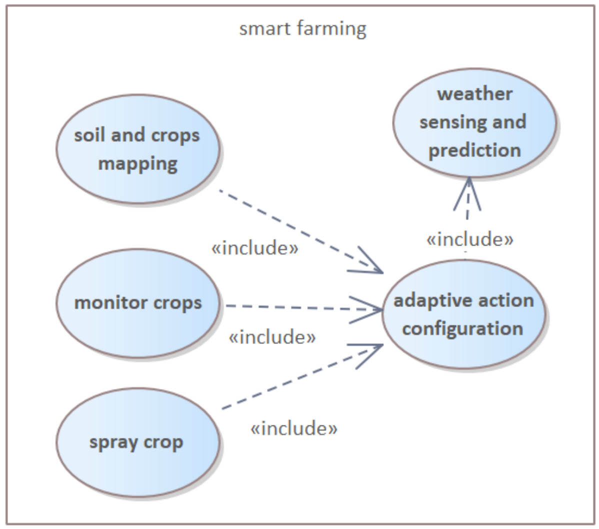

Figure 4.7: Smart farming

By analysing the requirements, the smart farming system has 3 main subsystems as shown in Figure 4.8, which are the crops monitor system, the crops treatment system and the fleet management system. In this paper, the focus is on the fleet management system, where its software and hardware configuration are dynamically changing. meaning that, in the context of Figure 4.1 and Figure 4.2, the managed system is the fleet management system. Those main subsystem has no clear boundary as they have interdependency.

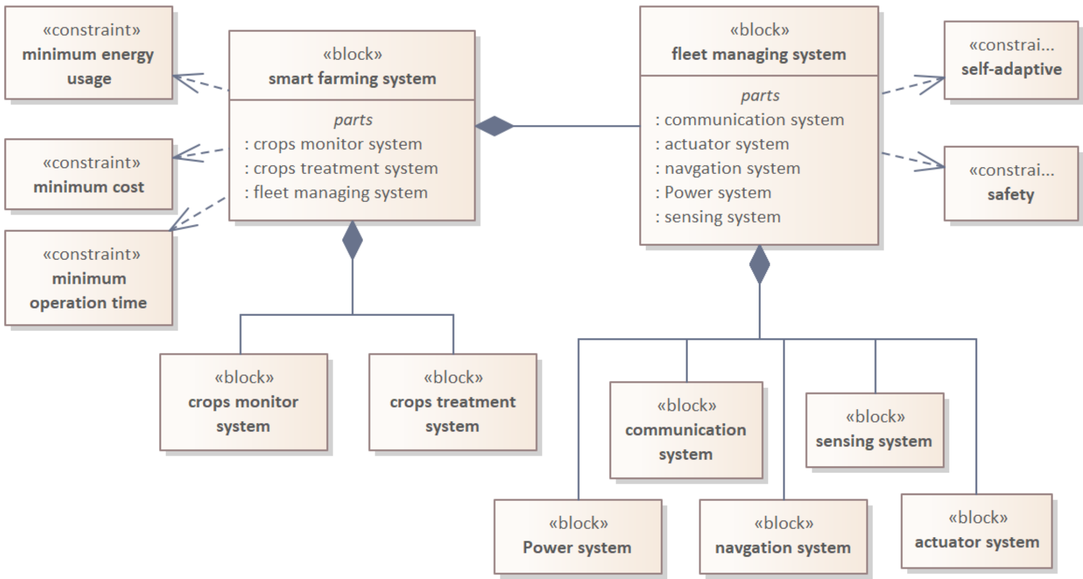

Figure 4.8: Main subsystems

The managed system consists of an edge and multiple UAVs (Unmanned Autonomous Vehicles), which are drones in this paper. The overview of the managed system is illustrated in Figure 4.9

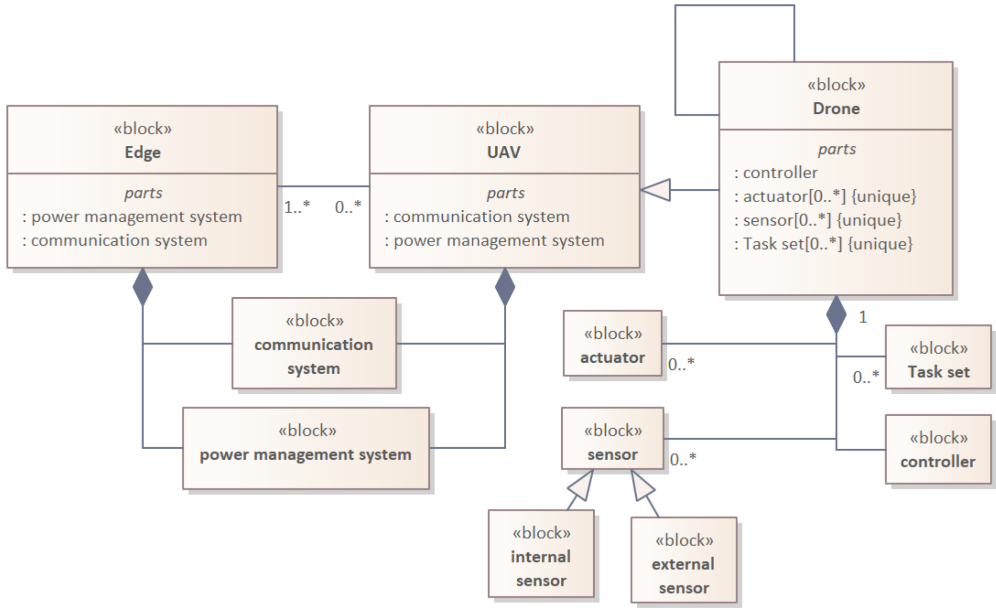

Figure 4.9: Managed System

3 Parameter and Objective

- parameter observed and parameter that should be improved
- objective used for improvement

4 Models

First, the system is divided into a few functional blocks. From that, we build the model. [][3]

$$F = f_1, f_2, ..., f_n \tag{4.1}$$

use ML to calibrate the model

relation of the model with weather can be learn from [][4]

it use k_means clustering. This will help to prune the design space by dynamically adjusting the constraint

[3]DESSERT: Design Space ExploRation Tool based on Power and Energy at System-Level

[4]https://www.researchgate.net/profile/Maksymilian-Madziel/publication/390746627_Impact_of_Weather_Conditions_on_Energy_Consumption_Modeling_for_Electric_Vehicles/links/67fc121dbd3f1930dd5d6818/Impact-of-Weather-Conditions-on-Energy-Consumption-Modeling-for-Electric-Vehicles.pdf?origin=journalDetail

Energy is the integral of power over time.This section consist of 3 main section: power model to estimate power consumption, operation power model to estimate operation time model to be used to estimate energy consumption and energy model

To estimate the energy consumed by a UAV, the factors that influence the energy consumption through required functionality are analysed. The functionality is grouped as power-dependent functions as shown in Figure 4.10. The average consumption is a part of the UAV property that will later be used by the design space construction. There are four main factors that influence the power consumption, which are, weather conditions, the height, speed and operating frequency of the UAV.

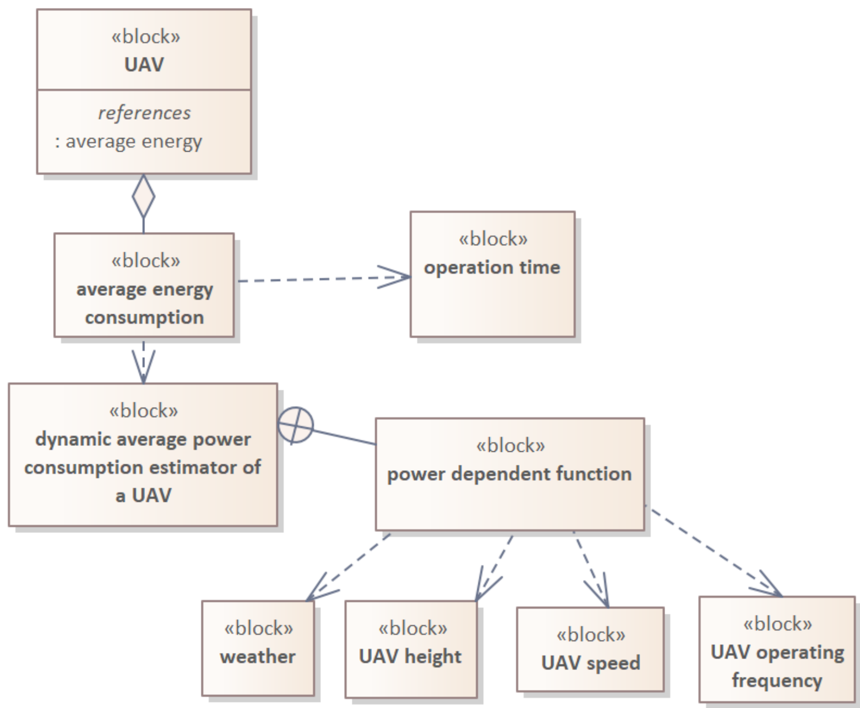

Figure 4.10: Dependency of power-dependent functions

4.1 Power model

Figure 4.11 shows the sub-block of power-dependent functions and their dependency in more detail. Basic power consumption is the power that is statically consumed either by the UAV in idle mode or active mode. The computation power consumption is the power consumed by the main processor(s) to execute a task. The sensor's power consumption is computed based on the same concept of computation's power consumption because both of them depend on the operation frequency of the UAV. The communication power consumption is also part of the computation power consumption.

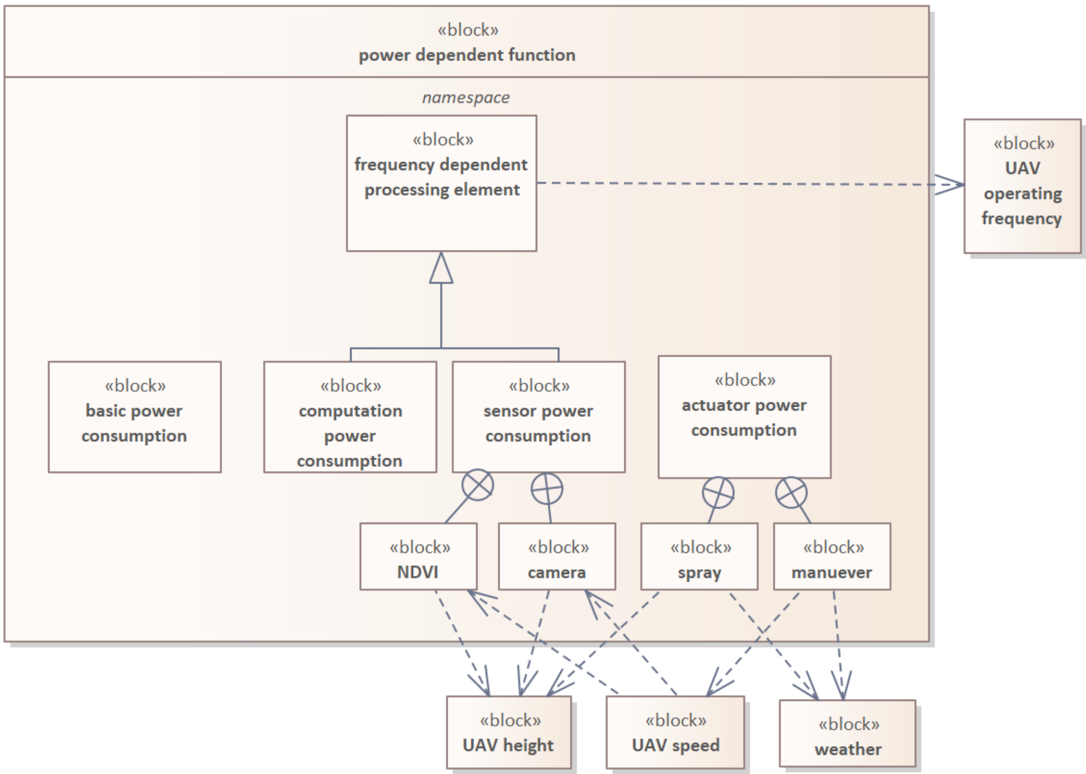

Figure 4.11: Power dependent functions

Based on figure 4.11 and equation 4.12, the power consumption can be derived as follow:

.

.

.

.

$$P_{total} = P_{basic} + P_{comp} + P_{sensor} + P_{actuator} \tag{4.2}$$

where,

$$P_{basic} = x \quad | \, x \in \mathbb{R} \tag{4.3}$$

$$P_{comp} = C \cdot V^2 \cdot f + P_{ind_{comp}} \tag{4.4}$$

$$P_{sensor} = C \cdot V^2 \cdot f + P_{ind_{sensor}} \tag{4.5}$$

$$P_{actuator} = P_{spray} + P_{manuever} \quad | \, p_{spray} \in \mathbb{R} \tag{4.6}$$

$$P_{manoeuver} = \frac{\Gamma(v_{wind} \cdot sin(\alpha) + v_i) + \Gamma_{h_0}(v_{wind} \cdot cos(\alpha)}{\eta} \tag{4.7}$$

$$\Gamma = \frac{g(m_d + m_p) \times v_a + \alpha v_a^3 + \beta \left(\frac{h}{h_{ref}}\right)(m_d + m_p)}{\delta} \tag{4.8}$$

$$\Gamma_{h_0} = \frac{g(m_d + m_p) \times v_a + \alpha v_a^3}{\delta} \tag{4.9}$$

Detailed drone (manoeuvre) energy consumption is complex as explained in [][5]

drone hover simple power estimation explained in [][6]

[][7]

4.2 Operation time model

The operation time depends on the desire resolution of sensor an covered area at a time by an actuator. Resolution of a sensor is define by coverage area per sensor pixel. The smaller the covered area per pixel, the higher the resolution is. Meaning that, the lower the position of the sensor, the higher the resolution is as illustrated in Figure 4.12. Varying the high of the drone, influence the covered area by an actuator, for example a spray, whereas the lower the drone the smaller covered area at a time causing longer time needed to cover the whole field with the consideration of constant speed. This can also be visualize using Figure 4.12.

[5] https://www.researchgate.net/publication/363896841_Quadrotor_Model_for_Energy_Consumption_Analysis/figures?lo=1&utm_source=google&utm_medium=organic

[6] https://news.quadpartpicker.com/how-to-estimate-and-calculate-drone-flight-characteristics/

[7] https://link.springer.com/article/10.1007/s11227-025-07105-0

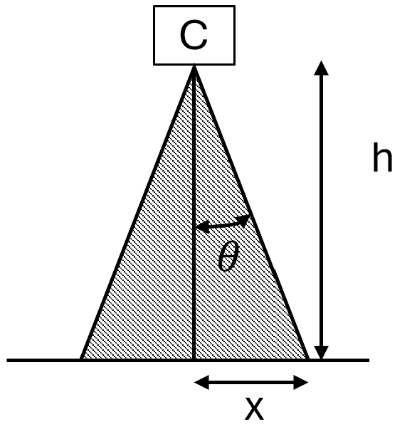

Figure 4.12: Relation between drone height, h and coverage, x

The total operating time can be define as:

$$T = (\frac{8tan^3(\frac{\theta}{2})}{\text{total area}}) \cdot \frac{h^3}{v} \tag{4.10}$$

4.3 Energy model

$$E = P * T \tag{4.11}$$

5 Model online calibration

how the model is improved over time - Parameter Estimation and Calibration

$$F = f_1, f_2, ..., f_n \tag{4.12}$$

$$S = x_1 f_1 + x_2 f_2 + ... + x_n f_n \quad | \, \forall x \in X \tag{4.13}$$

L is leaning model and I is the input (observed data and configuration)

$$X = L(I) \tag{4.14}$$

6 Online Design Space Exploration

- Online DSE
- How is it handled
- optimisation algorithm

- algorithm complexity

5

Implementation

1 Data collection: Simulation

- Subsystem simulation for parameter observation
- framework
- how observed data is handled
- how the model is improved using observed parameters

2 Model adaptation

- model improvement simulation
- framework to implement and improve the model

3 Exploration

- Exploring the design space

6

Evaluation

1 benefits of dynamic frequency

3 task, same arival time $a_i = 0 \mid i = \{0, 1, 2\}$

Task	**Energy Consumption**	
$\tau_0(1, 14)$, $\tau_1(2, 4)$, $\tau_2(3, 12)$	Total energy (on a single PE)	~ 18
	Total energy (on 2 homogeneous PE with dynamic frequency scaling and a recovery slot for fault handling	~ 14

Table 6.1: energy consumption

2 Limits for energy minimisation

Each task/processing element has a minimum frequency. below it ... high energy ¡graph¿

3 Computation complexity

4 Simulation result

¡graph f, utilization, PE , ...¿

7

Summary and Outlook

Bibliography

A

Appendix