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: June 3, 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, June 3, 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 **??**. 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`

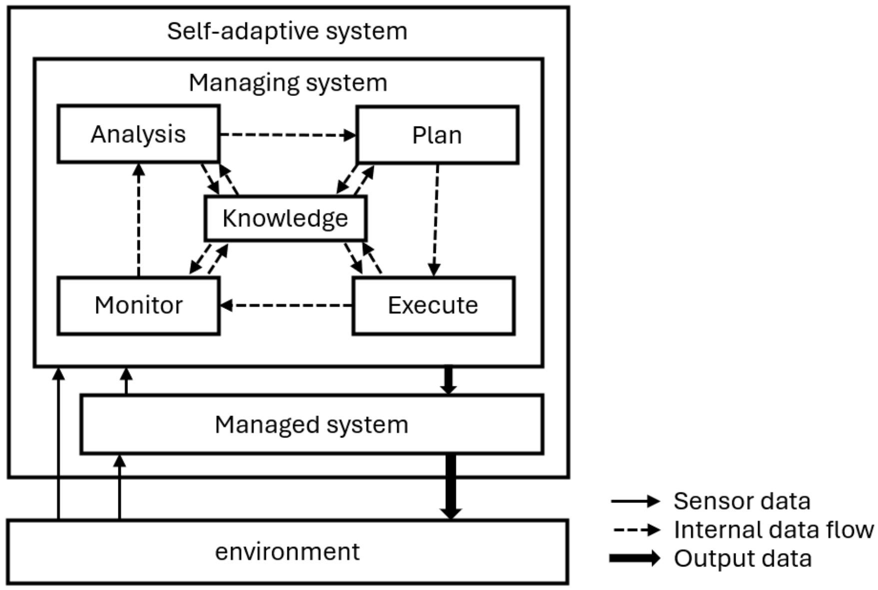

Figure 4.1: MAPE-K concept

< ... 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 **??**

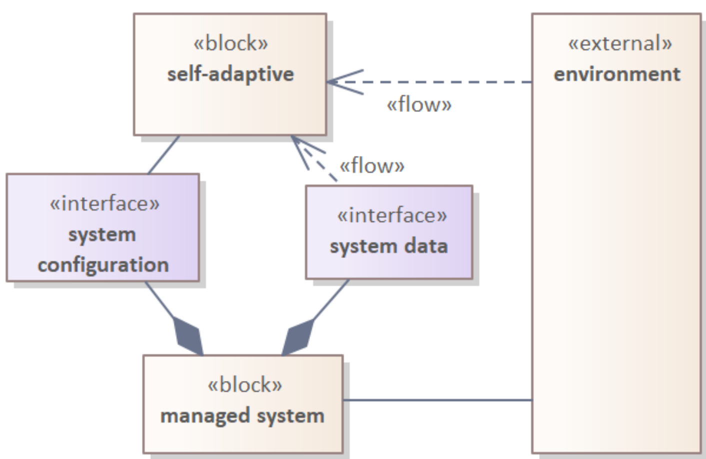

Figure 4.2: Self-adaptive as a component in a larger system view

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

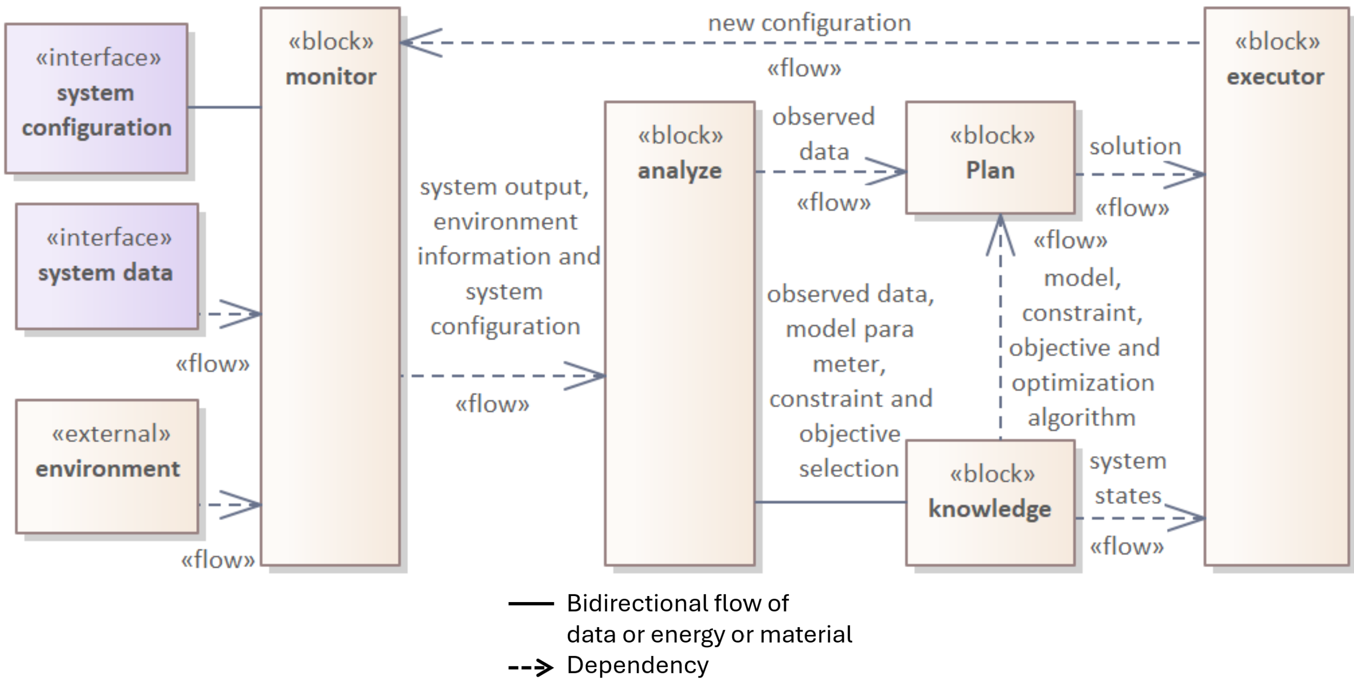

Figure 4.3: MAPE-k adaptation within self-adaptive block

< ... 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.

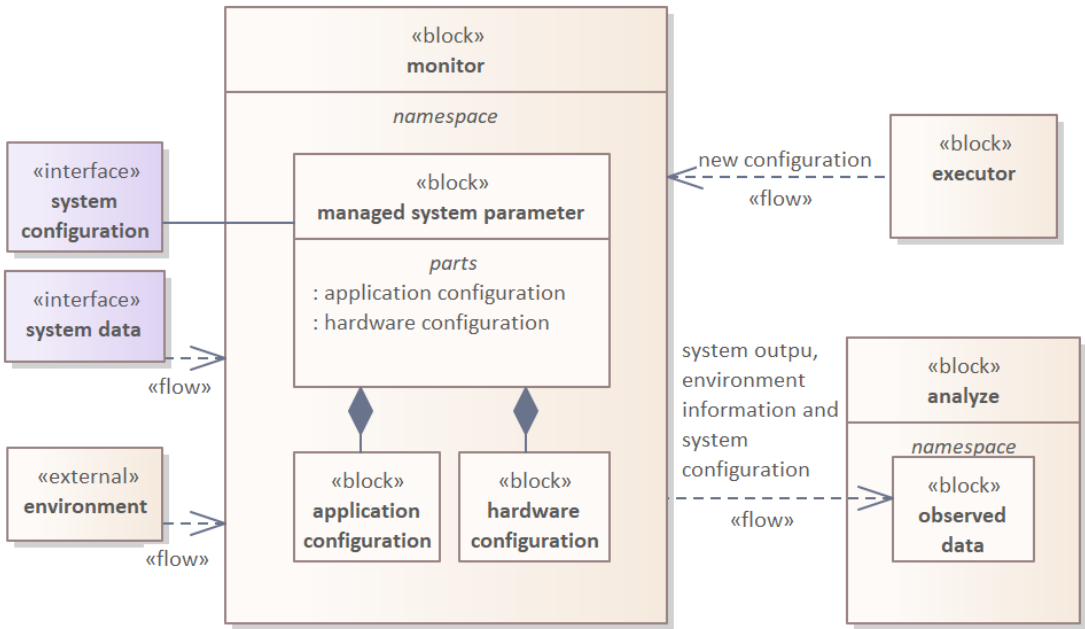

Figure 4.4: Monitor input and output

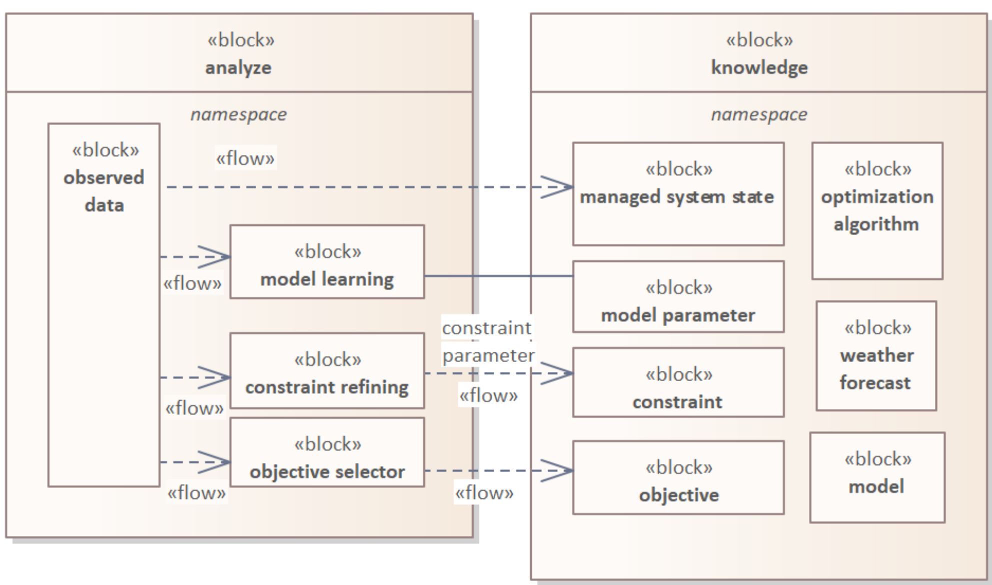

Figure 4.5: Knowledge and analysis block in self-adaptive block

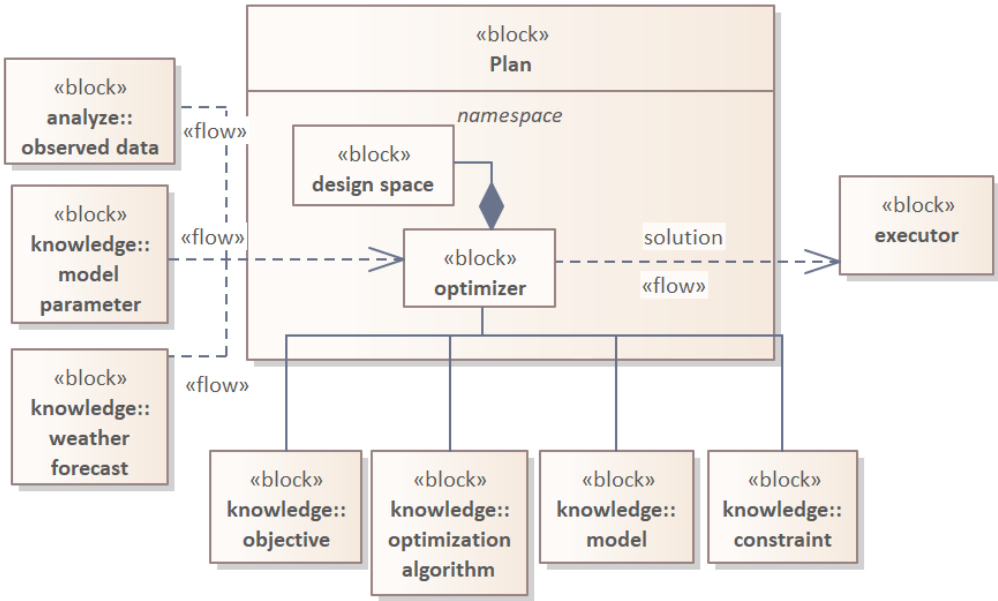

Figure 4.6: Plan block in self-adaptive block

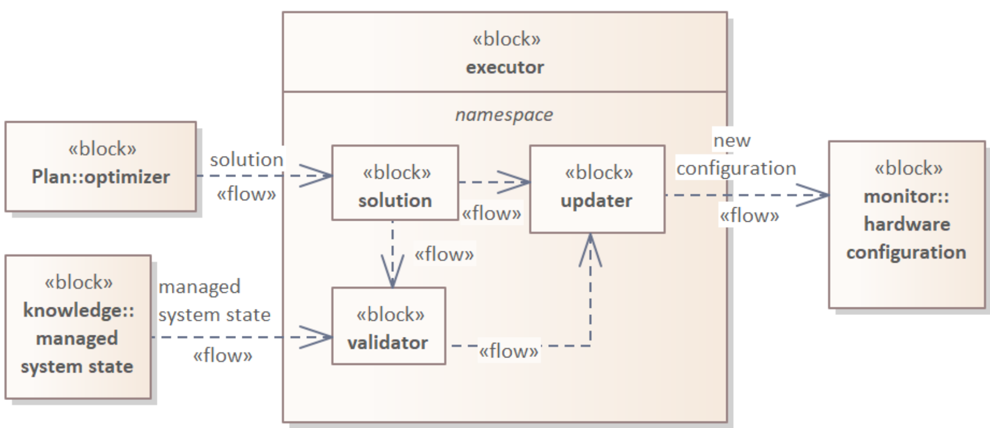

Figure 4.7: Executor block in self-adaptive block

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 4.8. These requirements are specifically tailored to address the key interests and objectives of this study.

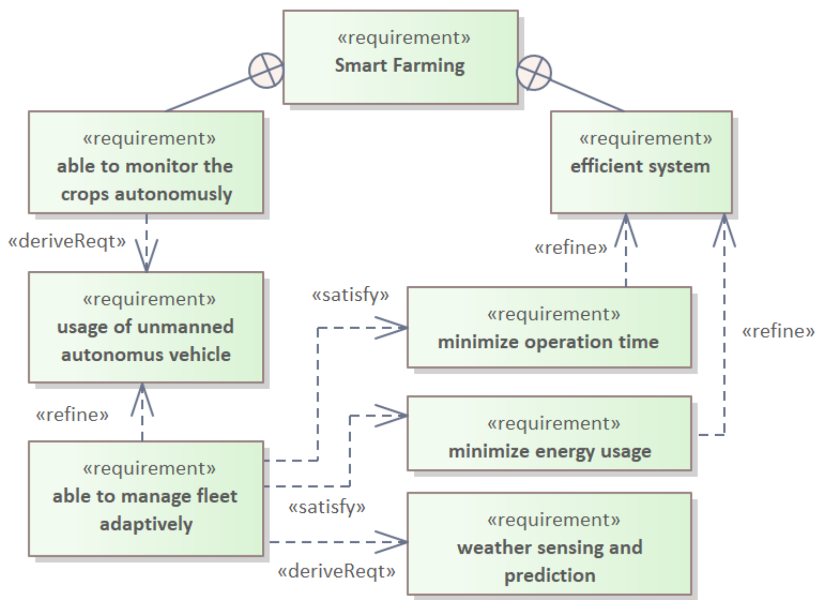

Figure 4.8: Smart farming requirements

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 4.9 to achieve certain goals

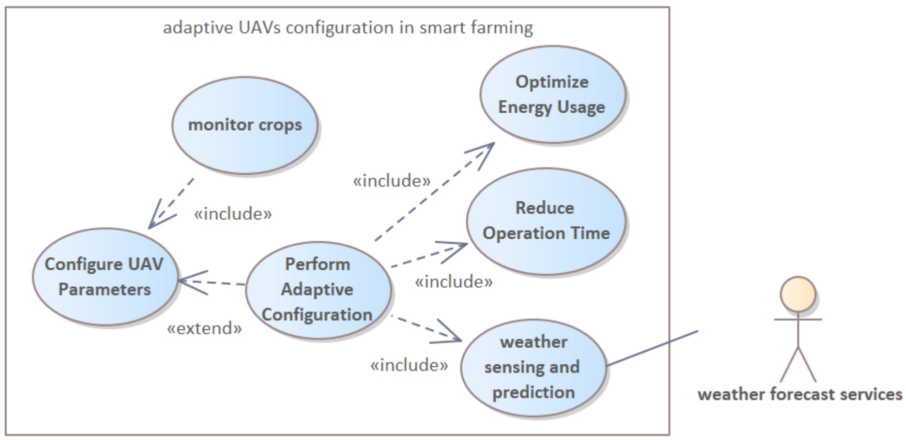

Figure 4.9: Adaptive UAVs configuration in smart farming use case

Figure 4.10 shows the logical system architecture of fleet managing system

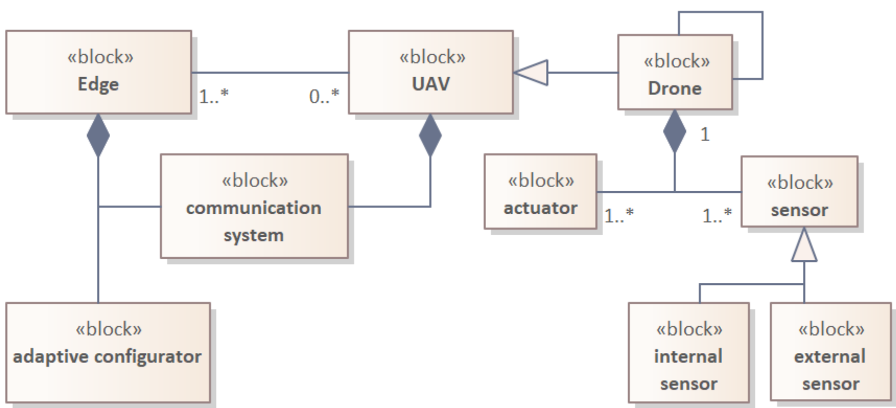

Figure 4.10: Managed system logical architecture: fleet managing system

The behaviour of the system is visualised in Figure 4.11

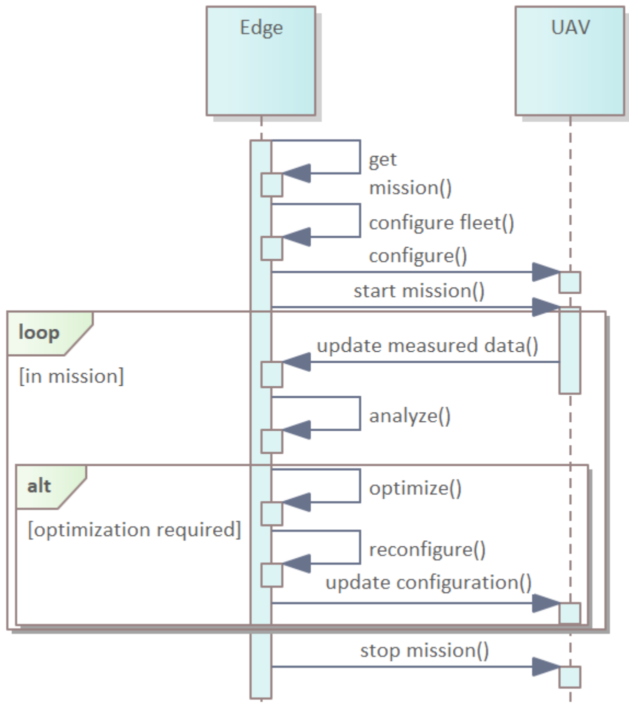

Figure 4.11: The system behaviour as a whole on the abstract level

3 Parameter and Objective

From Figure 4.8 and Figure 4.9, the objective of the implementation of self-adaptive can be listed as follow:

1. Minimization of Energy consumption
2. Minimization of operation time

The observed parameter are controlled parameter like speed and environmental information like wind speed. This parameter will be discussed in detail in incoming section and chapter. This observation is used by the self-adaptive component to optimize the controlled parameter to achieves those objectives

4 Models

This section describes the model used to construct the design space, which will later serve as the basis for optimizing controllable parameters. The modeling approach is inspired by the method presented in [][3], where the system is decomposed into a set of functional blocks, each representing a component of the overall model.

In this framework, the system F is represented as a collection of individual functions:

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

Each function f_i represents a distinct operation or behavior of the system. These functions collectively define the operational model and are subject to evaluation based on specific performance metrics.

The detailed explanation of how this functional decomposition is applied follows in the next subsection.

Moreover, the evaluation of these functions can be refined over time using machine learning techniques. This adaptive capability enables not only improved estimation and prediction accuracy for each function but also enhances the modeling of system constraints. As a result, the design space can be effectively pruned, leading to reduced computational costs.

The next subsection will focus specifically on the evaluation of the functions with respect to energy consumption, forming the foundation for modeling the system's total energy usage.

5 Energy model

In this paper, the computed energy consumption is an estimate of the average energy consumption. Based on the definition, the computed total energy can be formulated as follows:

$$E = \sum_{i}^{N_d} P \times T \tag{4.2}$$

where N_d is the total number of used drones. To estimate the energy consumed by a UAV, the factors that influence the energy consumption through required functionality are analysedand the result is shown in Figure 4.12

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

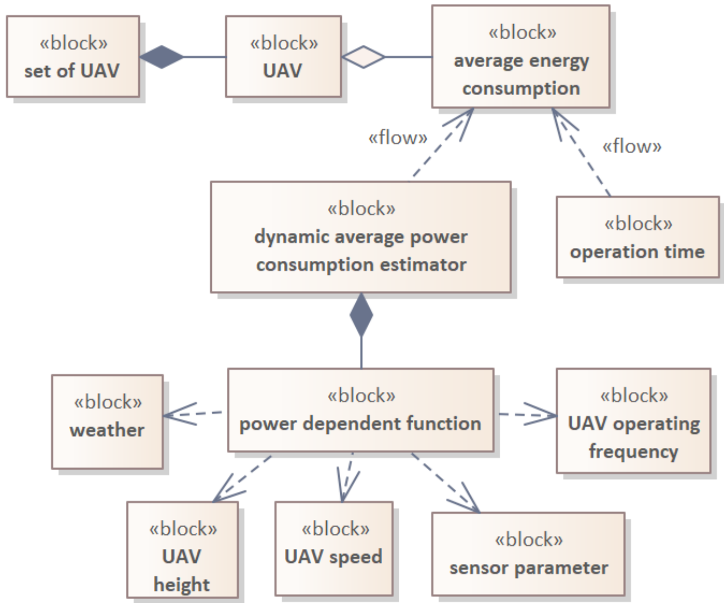

Figure 4.12: Energy model for the set of drones

The functionality is grouped as power-dependent functions. Five main factors influence the power consumption, which are weather conditions, the height, speed and operating frequency of the UAV and the parameters of the sensor attached to the UAV.

It can be seen from the diagram that the model can be split into two main models, which are the power model and the operation time model. Those models will be explained in the upcoming subsection

5.1 Power model

The drone system could be split into 5 main functionalities as shown in Figure 4.13 which are, basic ,computation, communication, sensor and actuator power consumption. To reduce the complexity, only one sensor is consider which is camera for image sensing and other basic sensing functionality like IMU sensor are considered as apart of actuator power consumption or specifically maneuvering function.

The computation power consumption is non zero if and only if there is another task that is not a fundamental task for sensing and actuating that need to be done by the drone. The computation resource by such the task is frequency dependent. Using a frequency as a parameter allow higher abstraction level to measure energy consumed by a processor.[][4]

[4] https://github.com/tgmattso/OpenMP_intro_tutorial/blob/master/omp_hands_on.pdf

A single processor can be seen as a capacitor with capacitance C with supplied voltage V. With those value, the work done can be calculated as follow:

$$W = C \cdot V^2 \tag{4.3}$$

The power can be derived with the operation frequency as in equation 4.4.

$$P = W \cdot f \tag{4.4}$$

In the considered scenario in this paper, the computation power consumption value will always be 0 as there is other task than maneuvering and image sensing.

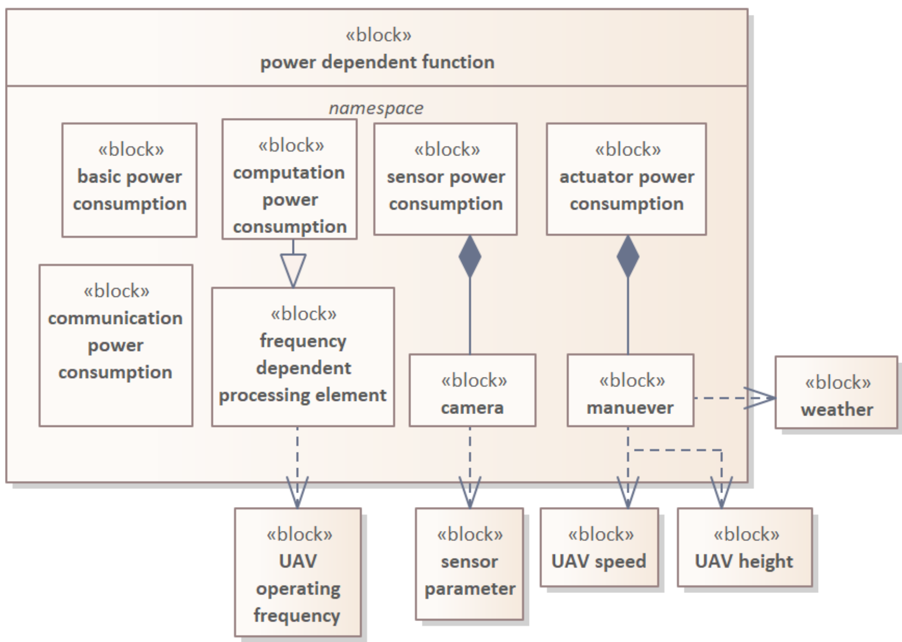

Figure 4.13: Power dependent functions

Based on Figure 4.12 and Figure 4.13, the total power consumption by a set of drones can be define as in the equation 4.20.

$$P_{total} = \sum_{i=1}^{N_{drone}} P_{basic,i} + P_{comp,i} + P_{sensor,i} + P_{actuator,i} \tag{4.5}$$

The $P_{basic,i}$ is the power of basic functionalities that cannot be avoided whenever the the drone is started or in other word power consumed in idle state. Meaning that, it is always true that $P_{basic} > 0$ and can be treat as a constant as in eqiation 4.6. The $P_{com,i}$ is the power consumed by a computing resource as explained before, where in this paper the value will always be 0.

The $P_{sensor,i}$ in this paper focused on the power consumed for image sensing including the power required to process the images. The $P_{actuator,i}$ is the power required by the maneuvering functionality.

$$P_{basic,i} = x_i \quad | \, x_i \in \mathbb{R} \tag{4.6}$$

In the considered scenario, the design of the drone is not the main focus, but the adaptive fleet configuration and its dependency. In general, focusing on improving a single drone hardware design configuration is not within the scope of this paper. For that reasons, the model used for every single functionality is more linear rather than complex model. This approach will alow the optimization process later much more less complex.

The $P_{sensor,i}$ or $P_{camera,i}$ depend only on the set FPS and number of pixel used by the sensor. This linear approach based on the analysis of the power consumed by an image sensing device with varying resolution and FPS done by [][5], where the power consumed is linearly increasing with the resolution when the resolution is above 320×240. This can be disribe using equation 4.7 and 4.8. The description of used parameter can be found in Table 4.1.

$$P_{sensor,i} = P_{camera,i} = \sigma_{t(i)} \cdot n_{pixel,i} + \omega_{t(i)} \cdot f_{fps,i} \tag{4.7}$$

$$t(i) \in T, \text{ where } |T| = N_{\text{type of drones}} \tag{4.8}$$

Symbol	**Description**
$m_{d,i}$	Drone's mass
$m_{p,i}$	Load mass
h, i	Drone's height
$h_{ref,i}$	Height reference
$v_{a,i}$	Resultant speed
g	Gravitational acceleration
$\alpha_{t(i)}$	Drone's drag coefficient factor
$\delta_{t(i)}$	Propulsion efficiency of the drone
Γ_i	Drone's thrust
v_{wind}	Wind speed
θ_{wind}	Angle of wind attack
v_i	Speed of the drone
$\eta_{t(i)}$	Power transfer efficiency
$\beta_{t(i)}$	Constant related to altitude's effect on air density

Table 4.1: List of parameters and their descriptions

The $P_{actuator,i}$ depends on the altitude of drone, wind speed, angle of wind attack, speed of drone, the mass of drone and the mass carry by the drone as

[5] Intelligent Systems and Applications-Impact of Image Sensor Output Data on Power Consumption of the Image Processing System

describe in the equation 4.9. The description of used parameter can be found in Table 4.1.

$$P_{actuator,i} = P_{manoeuver,i} = \frac{\Gamma_i(v_{wind} \cdot sin(\theta_{wind}) + v_i) + \Gamma_{h_0,i}(v_{wind} \cdot cos(\theta_{wind})}{\eta_{t(i)}} \tag{4.9}$$

[][6]

$$\Gamma_i = \frac{g(m_{d,i} + m_{p,i}) \times v_{a,i} + \alpha_{t(i)} v_{a,i}^3 + \beta_{t(i)} \left(\frac{h,i}{h_{ref,i}}\right)(m_{d,i} + m_{p,i})}{\delta_{t(i)}} \tag{4.10}$$

[][7]

$$\Gamma_{h_0,i} = \frac{g(m_{d,i} + m_{p,i}) \times v_{a,i} + \alpha v_{a,i}^3}{\delta_{t(i)}} \tag{4.11}$$

Detailed drone (manoeuvre) energy consumption is complex as explained in [][8] drone hover simple power estimation explained in [][9]

5.2 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 4.14. 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 4.14.

[6] https://link.springer.com/article/10.1007/s11227-025-07105-0
[7] comparative study on energy consumption
[8] https://www.researchgate.net/publication/363896841_Quadrotor_Model_for_Energy_Consumption_Analysis/figures?lo=1&utm_source=google&utm_medium=organic
[9] https://news.quadpartpicker.com/how-to-estimate-and-calculate-drone-flight-characteristics/

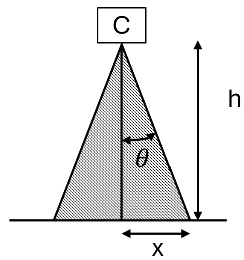

Figure 4.14: Relation between drone height, h and coverage, x

The total operating time can be define as:

$$T_i = (\frac{8tan^3(\frac{\theta_{\text{camera},i}}{2})}{\text{total area}_i}) \cdot \frac{h_i^3}{v_i} \tag{4.12}$$

5.3 Constraint

$$\text{resolution}_{max,x} < \text{resolution}_{x,i} = \frac{2 \times tan(\frac{\theta_{\text{camera},i}}{2}) \times h}{\text{pix}_{x,i}} < \text{resolution}_{min,x} \tag{4.13}$$

$$\text{resolution}_{max,y} < \text{resolution}_{y,i} = \frac{2 \times tan(\frac{\theta_{\text{camera},i}}{2}) \times h}{\text{pix}_{y,i}} < \text{resolution}_{min,y} \tag{4.14}$$

$$\frac{\text{resolution}_{y,i}}{\text{resolution}_{x,i}} = \frac{\text{pix}_{y,i}}{\text{pix}_{x,i}} \tag{4.15}$$

$$f_{fps,i} > \frac{v_i}{2 \times tan(\frac{\theta_{\text{camera},i}}{2}} \tag{4.16}$$

$$\text{maximum energy per drone}, P_{t(i)} \times T_{t(i)} < C_{\text{battery},i} \tag{4.17}$$

$$\text{total area} = \sum_{i}^{N_{drones}} \text{total\_area}_i \tag{4.18}$$

$$\text{number of drone, } N_{drone} < \text{number of available drone, } N_{available} \tag{4.19}$$

6 Model online calibration

The model explained in pervious section, section 4.4 contains parameter with uncertainties which are

- δ
- β
- η
- x
- ...

To have better estimation of overall power consumption and estimation of operation time, the value of those parameter should be improve over time. This is where the model learning block in Figure 4.5 play a big role. This block implement machine learning approch.

6.1 Machine learning model

$$F = f_1, f_2, ..., f_n \tag{4.20}$$

$$S = x_1 f_1 + x_2 f_2 + ... + x_n f_n \quad | \, \forall x \in X \tag{4.21}$$

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

$$X = L(I) \tag{4.22}$$

6.2 Dataset

[dataset][10]

6.3 Training and execution

A dataset is required to train and validate the model during the simulation of a self-adaptive system.

This paper focuses on the approach and not specifically on the case study scenario that is used to explain the approach. The dataset used to train and validate the ML model during the simulation is gathered from the literature.

The gathered dataset is not fully accurate as they are combined from different datasets of different drones, but sufficient to simulate the approach used in this paper.

[10] `https://kilthub.cmu.edu/articles/dataset/Data_Collected_with_Package_Delivery_Quadcopter_Drone/12683453`

The dataset used for basic power consumption and manoeuvre power consumption, which includes wind angle and speed parameters, is used from [][11].

The sensor focused on for the simulation is a camera whose resolution and FPS can be varied. The dataset is from [][12]

7 Online Design Space Exploration

- Online DSE
- How is it handled
- optimisation algorithm
- algorithm complexity

[11] https://kilthub.cmu.edu/articles/dataset/Data_Collected_with_Package_Delivery_Quadcopter_Drone/12683453

[12] Intelligent Systems and Applications-Impact of Image Sensor Output Data on Power Consumption of the Image Processing System

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