Energy-Aware Coverage Planning and Scheduling for Autonomous Aerial Robots

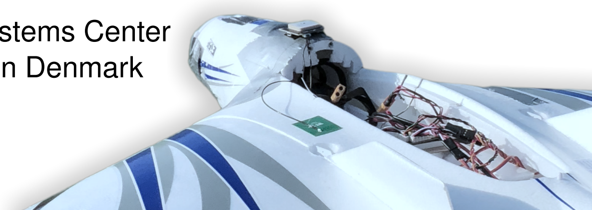
Ph.D. Thesis Defense

Adam Seewald

Unmanned Aerial Systems Center
University of Southern Denmark

January 31, 2022

Funded by H2020 TeamPlay



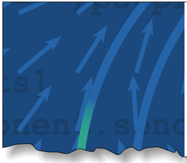
What does the title mean?

Energy-Aware Coverage Planning and Scheduling for Autonomous Aerial Robots

TEAMPLAY

- Derive energy models for all the energy components of aerial robots in autonomous use cases
- Visit efficiently every location in a given space, e.g., for perception in precision agriculture
- Simultaneously derive an optimal motion plan for the path and a scheduling policy for the computing hardware

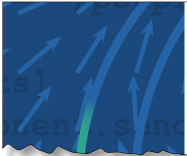
What does the title mean?



Energy-Aware Coverage Planning and Scheduling for Autonomous Aerial Robots

- Derive energy models for all the energy components of aerial robots in autonomous use cases
- Visit efficiently every location in a given space, e.g., for perception in precision agriculture
- Simultaneously derive an optimal motion plan for the path and a scheduling policy for the computing hardware

What does the title mean?



Energy-Aware Coverage
Planning and Scheduling for
Autonomous Aerial Robots

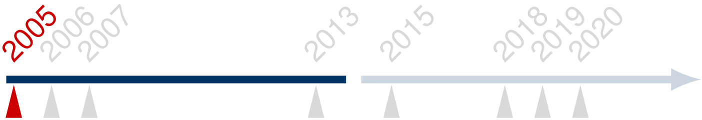
- Derive energy models for all the energy components of aerial robots in autonomous use cases
- Visit efficiently every location in a given space, e.g., for perception in precision agriculture
- Simultaneously derive an optimal motion plan for the path and a scheduling policy for the computing hardware

Why motion planning with computations scheduling?

- "[Computing energy included motion planning] *shows improved performance over the baseline and looks to be promising solution to the low-power motion planning problem*"[1]
- "*The energy budget for sensing and computing is commensurate with that of actuation*"[2]
- "*Results show that motion accounts for less than 50% of the total power consumption*"[3]

[1] Sudhakar et al. (2020) "Balancing actuation and computing energy in motion planning"
[2] Ondrúška et al. (2015) "Scheduled perception for energy-efficient path following"
[3] Mei et al. (2005) "A case study of mobile robot's energy consumption and conservation techniques"

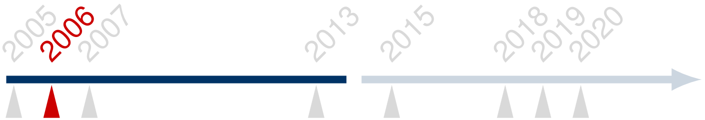
Planning-scheduling in the literature



- Mei et al. introduced mobile robot's motion and computations energies; theorized DVS, DFS, peripheral selection
- Brateman et al. optimization for DVS, DFS, and motor's speed
- Zhang et al. optimal control for DFS and robot's speed
- Sadrpour et al. plan energy prediction; theorized plan tuning

- Ondrúška et al. scheduling of path following
- Lahijanian et al. scheduling localization
- Ho et al. scheduling of navigation onto waypoints
- Sudhakar et al. scheduling of anytime planning

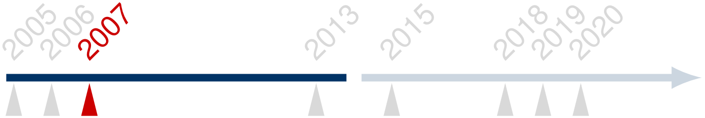
Planning-scheduling in the literature



- Mei et al. introduced mobile robot's motion and computations energies; theorized DVS, DFS, peripheral selection
- Brateman et al. optimization for DVS, DFS, and motor's speed
- Zhang et al. optimal control for DFS and robot's speed
- Sadrpour et al. plan energy prediction; theorized plan tuning

- Ondrúška et al. scheduling of path following
- Lahijanian et al. scheduling localization
- Ho et al. scheduling of navigation onto waypoints
- Sudhakar et al. scheduling of anytime planning

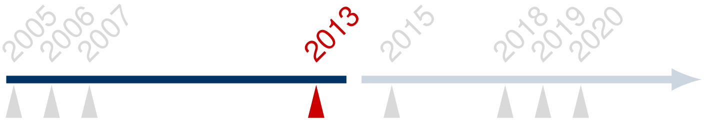
Planning-scheduling in the literature



- Mei et al. introduced mobile robot's motion and computations energies; theorized DVS, DFS, peripheral selection
- Brateman et al. optimization for DVS, DFS, and motor's speed
- Zhang et al. optimal control for DFS and robot's speed
- Sadrpour et al. plan energy prediction; theorized plan tuning

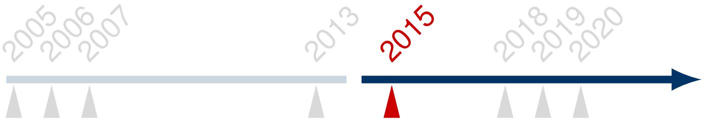
- Ondrúška et al. scheduling of path following
- Lahijanian et al. scheduling localization
- Ho et al. scheduling of navigation onto waypoints
- Sudhakar et al. scheduling of anytime planning

Planning-scheduling in the literature



- Mei et al. introduced mobile robot's motion and computations energies; theorized DVS, DFS, peripheral selection
- Brateman et al. optimization for DVS, DFS, and motor's speed
- Zhang et al. optimal control for DFS and robot's speed
- Sadrpour et al. plan energy prediction; theorized plan tuning
- Ondrúška et al. scheduling of path following
- Lahijanian et al. scheduling localization
- Ho et al. scheduling of navigation onto waypoints
- Sudhakar et al. scheduling of anytime planning

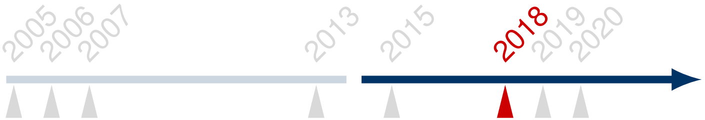
Planning-scheduling in the literature



- Mei et al. introduced mobile robot's motion and computations energies; theorized DVS, DFS, peripheral selection
- Brateman et al. optimization for DVS, DFS, and motor's speed
- Zhang et al. optimal control for DFS and robot's speed
- Sadrpour et al. plan energy prediction; theorized plan tuning

- Ondrúška et al. scheduling of path following
- Lahijanian et al. scheduling localization
- Ho et al. scheduling of navigation onto waypoints
- Sudhakar et al. scheduling of anytime planning

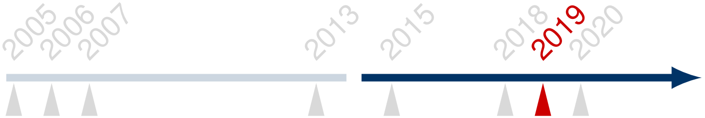
Planning-scheduling in the literature



- Mei et al. introduced mobile robot's motion and computations energies; theorized DVS, DFS, peripheral selection
- Brateman et al. optimization for DVS, DFS, and motor's speed
- Zhang et al. optimal control for DFS and robot's speed
- Sadrpour et al. plan energy prediction; theorized plan tuning

- Ondrúška et al. scheduling of path following
- Lahijanian et al. scheduling localization
- Ho et al. scheduling of navigation onto waypoints
- Sudhakar et al. scheduling of anytime planning

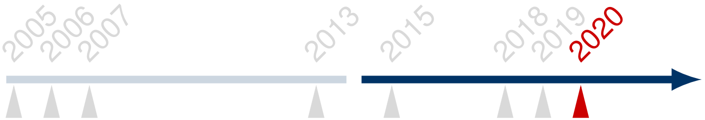
Planning-scheduling in the literature



- Mei et al. introduced mobile robot's motion and computations energies; theorized DVS, DFS, peripheral selection
- Brateman et al. optimization for DVS, DFS, and motor's speed
- Zhang et al. optimal control for DFS and robot's speed
- Sadrpour et al. plan energy prediction; theorized plan tuning

- Ondrúška et al. scheduling of path following
- Lahijanian et al. scheduling localization
- Ho et al. scheduling of navigation onto waypoints
- Sudhakar et al. scheduling of anytime planning

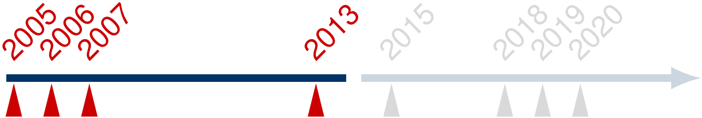
Planning-scheduling in the literature



- Mei et al. introduced mobile robot's motion and computations energies; theorized DVS, DFS, peripheral selection
- Brateman et al. optimization for DVS, DFS, and motor's speed
- Zhang et al. optimal control for DFS and robot's speed
- Sadrpour et al. plan energy prediction; theorized plan tuning

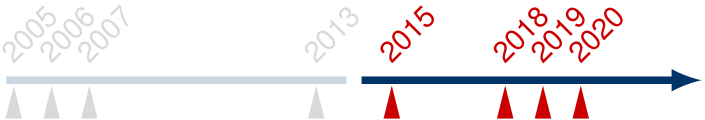
- Ondrúška et al. scheduling of path following
- Lahijanian et al. scheduling localization
- Ho et al. scheduling of navigation onto waypoints
- Sudhakar et al. scheduling of anytime planning

Planning-scheduling in the literature (2)



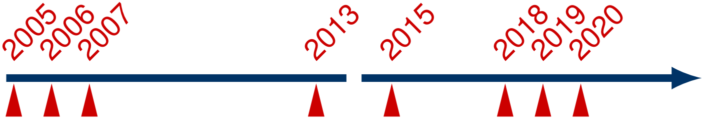
- 2000–2010 focus on hardware aspects
  - frequency and voltage
  - motor and travel velocities
- 2010–2020 focus on software aspects
  - perception, localization, navigation, and anytime planning
- Ground-based robots (Pioneer 3DX, Pack-Bot UGV, and ARC Q14)
- Different energy models and approaches
  - regression, Bayesian, and analytical
  - greedy, Pareto front, optimization, and RL

Planning-scheduling in the literature (2)



- 2000–2010 focus on hardware aspects
  - frequency and voltage
  - motor and travel velocities
- 2010–2020 focus on software aspects
  - perception, localization, navigation, and anytime planning
- Ground-based robots (Pioneer 3DX, Pack-Bot UGV, and ARC Q14)
- Different energy models and approaches
  - regression, Bayesian, and analytical
  - greedy, Pareto front, optimization, and RL

Planning-scheduling in the literature (2)



- 2000–2010 focus on hardware aspects
  - frequency and voltage
  - motor and travel velocities
- 2010–2020 focus on software aspects
  - perception, localization, navigation, and anytime planning
- Ground-based robots (Pioneer 3DX, Pack-Bot UGV, and ARC Q14)
- Different energy models and approaches
  - regression, Bayesian, and analytical
  - greedy, Pareto front, optimization, and RL

Planning-scheduling in this work

Contributions 1–3

1. Computations model based on `powprofiler`
2. Adaptation of a battery model into `powprofiler`
3. Cohesive differential periodic model

International Journal of Parallel Programming (2021) 49:136–157
https://doi.org/10.1007/s10766-019-00645-y

Check for updates

Coarse-Grained Computation-Oriented Energy Modeling for Heterogeneous Parallel Embedded Systems

Adam Seewald[1] · Ulrik Pagh Schultz[1] · Emad Ebeid[1] · Henrik Skov Midtiby[1]

Received: 12 August 2019 / Accepted: 2 November 2019 / Published online: 23 November 2019

Abstract
Limited energy availability is among the most challenging considerations developers face for heterogeneous systems and is critical for battery-powered devices. For complex systems composed of mechanical and computational units, such as drones and mobile robots, more than half of the power consumption can be due to the computational operations. Critically, these systems are often composed of many components, interacting concurrently to achieve specific functionality. As a result, power prediction and estimation can be a challenging task, especially if different computational units, such as CPU and GPU, should be modeled. In this paper, we focus on limited energy availability for mobile heterogeneous devices powered by a battery and present a coarse-grained computation-oriented energy modeling approach. Our approach predicts the energy consumption of a set of software components, in a specific configuration, executed according to a given scheduling policy. The model, determined numerically from several empirical power samples, describes the energy consumed by a software configuration and can be used for energy-aware planning and optimization from a computational point of view. It can potentially support a complex embedded system in maximizing the level of autonomy while minimizing power consumption and preserving the most appropriate amount of battery charge by finding the right rate of quality of service. Our approach is supported and validated by the design and implementation of a profiling tool. The tool abstracts computational energy behavior and describes the current battery drain as a function of all the admissible configurations.

Keywords Energy profiling · Energy modeling · Embedded platforms · Heterogeneous computing

This work is supported and partly funded by the European Union's Horizon 2020 research and innovation program under Grant Agreement No. 779882 (TeamPlay).

✉ Adam Seewald
ads@mmmi.sdu.dk

1 SDU UAS Center, University of Southern Denmark, Odense, Denmark

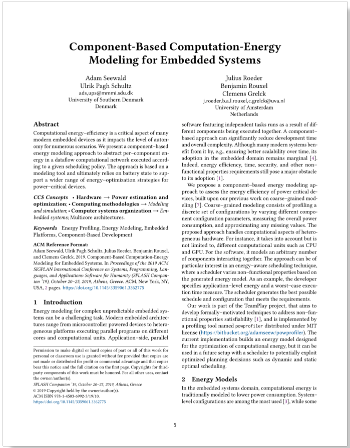
Springer

Seewald et al. (2021) "Coarse-grained computation-oriented energy modeling for heterogeneous parallel embedded systems"

Planning-scheduling in this work

Contributions 1–3

1. Computations model based on `powprofiler`
2. Adaptation of a battery model into `powprofiler`
3. Cohesive differential periodic model



Component-Based Computation-Energy Modeling for Embedded Systems

Adam Seewald
Ulrik Pagh Schultz
ads,ups@mmmi.sdu.dk
University of Southern Denmark
Denmark

Julius Roeder
Benjamin Rouxel
Clemens Grelck
j.roeder,b.a.l.rouxel,c.grelck@uva.nl
University of Amsterdam
Netherlands

Abstract

Computational energy–efficiency is a critical aspect of many modern embedded devices as it impacts the level of autonomy for numerous scenarios. We present a component–based energy modeling approach to abstract per–component energy in a dataflow computational network executed according to a given scheduling policy. The approach is based on a modeling tool and ultimately relies on battery state to support a wider range of energy–optimization strategies for power–critical devices.

CCS Concepts • **Hardware** → **Power estimation and optimization**; • **Computing methodologies** → *Modeling and simulation*; • **Computer systems organization** → *Embedded systems*; Multicore architectures.

Keywords Energy Profiling, Energy Modeling, Embedded Platforms, Component-Based Development

ACM Reference Format:
Adam Seewald, Ulrik Pagh Schultz, Julius Roeder, Benjamin Rouxel, and Clemens Grelck. 2019. Component-Based Computation-Energy Modeling for Embedded Systems. In *Proceedings of the 2019 ACM SIGPLAN International Conference on Systems, Programming, Languages, and Applications: Software for Humanity (SPLASH Companion '19), October 20–25, 2019, Athens, Greece.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3359061.3362775

1 Introduction

Energy modeling for complex unpredictable embedded systems can be a challenging task. Modern embedded architectures range from microcontroller powered devices to heterogeneous platforms executing parallel programs on different cores and computational units. Application–side, parallel software featuring independent tasks runs as a result of different components being executed together. A component–based approach can significantly reduce development time and overall complexity. Although many modern systems benefit from it by, e.g., ensuring better scalability over time, its adoption in the embedded domain remains marginal [4]. Indeed, energy efficiency, time, security, and other non–functional properties requirements still pose a major obstacle to its adoption [1].

We propose a component–based energy modeling approach to assess the energy efficiency of power critical devices, built upon our previous work on coarse–grained modeling [7]. Coarse–grained modeling consists of profiling a discrete set of configurations by varying different component configuration parameters, measuring the overall power consumption, and approximating any missing values. The proposed approach handles computational aspects of heterogeneous hardware. For instance, it takes into account but is not limited to, different computational units such as CPU and GPU. For the software, it models an arbitrary number of components interacting together. The approach can be of particular interest in an energy–aware scheduling technique, where a scheduler varies non–functional properties based on the generated energy model. As an example, the developer specifies application–level energy and a worst–case execution time measure. The scheduler generates the best possible schedule and configuration that meets the requirements.

Our work is part of the TeamPlay project, that aims to develop formally–motivated techniques to address non–functional properties satisfiability [1], and is implemented by a profiling tool named `powprofiler` distributed under MIT license (https://bitbucket.org/adamseew/powprofiler). The current implementation builds an energy model designed for the optimization of computational energy, but it can be used in a future setup with a scheduler to potentially exploit optimized planning decisions such as dynamic and static optimal scheduling.

2 Energy Models

In the embedded systems domain, computational energy is traditionally modeled to lower power consumption. System–level configurations are among the most used [3], while some

SPLASH Companion '19, October 20–25, 2019, Athens, Greece

ACM ISBN 978-1-4503-6992-3/19/10.
https://doi.org/10.1145/3359061.3362775

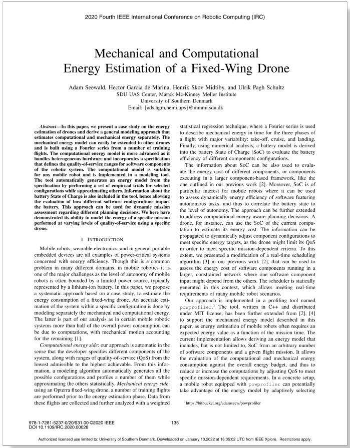
5

Seewald et al. (2021) "Coarse-grained computation-oriented energy modeling for heterogeneous parallel embedded systems"
— (2019) "Component-based computation-energy modeling for embedded systems"

Planning-scheduling in this work

Contributions 1–3

1. Computations model based on `powprofiler`
2. Adaptation of a battery model into `powprofiler`
3. Cohesive differential periodic model



2020 Fourth IEEE International Conference on Robotic Computing (IRC)

Mechanical and Computational Energy Estimation of a Fixed-Wing Drone

Adam Seewald, Hector Garcia de Marina, Henrik Skov Midtiby, and Ulrik Pagh Schultz
SDU UAS Center, Mærsk Mc-Kinney Møller Institute
University of Southern Denmark
Email: {ads,hgm,hemi,ups}@mmmi.sdu.dk

Abstract—In this paper, we present a case study on the energy estimation of drones and derive a general modeling approach that estimates computational and mechanical energy separately. The mechanical energy model can easily be extended to other drones and is built using a Fourier series from a number of training flights. The computational energy model is more advanced as it handles heterogeneous hardware and incorporates a specification that defines the quality-of-service ranges for software components of the robotic system. The computational model is suitable for any mobile robot and is implemented in a modeling tool. The tool automatically generates an energy model from the specification by performing a set of empirical trials for selected configurations while approximating others. Information about the battery State of Charge is also included in the tool, hence allowing the evaluation of how different software configurations impact the battery. This approach can be used for dynamic mission assessment regarding different planning decisions. We here have demonstrated its ability to model the energy of a specific mission performed at varying levels of quality-of-service using a specific drone.

I. Introduction

Mobile robots, wearable electronics, and in general portable embedded devices are all examples of power-critical systems concerned with energy efficiency. Though this is a common problem in many different domains, in mobile robotics it is one of the major challenges as the level of autonomy of mobile robots is often bounded by a limited power source, typically represented by a lithium-ion battery. In this paper, we propose a systematic approach based on a case study, to estimate the energy consumption of a fixed-wing drone. An accurate estimation of the system within a specific configuration is done by modeling separately the mechanical and computational energy. The latter is part of our analysis as in certain mobile robotic systems more than half of the overall power consumption can be due to computations, with mechanical motion accounting for the remaining [1].

Computational energy side: our approach is automatic in the sense that the developer specifies different components of the system, along with ranges of quality-of-service (QoS) from the lowest admissible to the highest achievable. From this information, a modeling algorithm automatically generates all the possible configurations and profiles a number of them while approximating the others statistically. *Mechanical energy side*: using an Opterra fixed-wing drone, a number of training flights are performed prior to the energy estimation phase. Data from these flights are collected and further analyzed with a weighted statistical regression technique, where a Fourier series is used to describe mechanical energy in time for the three phases of a flight with major variability: take-off, cruise, and landing. Finally, using numerical analysis, a battery model is derived into the battery State of Charge (SoC) to evaluate the battery efficiency of different components configurations.

The information about SoC can be also used to evaluate the energy cost of different components, or components executing in a larger component-based framework, like the one outlined in our previous work [2]. Moreover, SoC is of particular interest for mobile robots where it can be used to assess dynamically energy efficiency of software featuring autonomous tasks, and thus to correlate the battery state to the level of autonomy. The approach can be further extended to address computational energy-aware planning decisions. A drone, for instance, can use the SoC of the current computation to estimate its energy cost. The information can be propagated to dynamically adjust component configurations to meet specific energy targets, as the drone might limit its QoS in order to meet specific mission-dependent criteria. To this extent, we presented a modification of a real-time scheduling algorithm [3] in our previous work [2], that can be used to assess the energy cost of software components running in a larger, constrained network where one software component input might depend from the others. The scheduler is statically generated in this context, which allows meeting real-time requirements of many mobile robot scenarios.

Our approach is implemented in a profiling tool named `powprofiler`.[1] The tool, written in C++ and distributed under MIT license, has been further extended from [2], [4] to support the mechanical energy model described in this paper, as energy estimation of mobile robots often requires an expected energy value as a function of the mission time. The current implementation allows deriving an energy model that includes, but is not limited to, SoC from an arbitrary number of software components and a given flight mission. It allows the evaluation of the computational and mechanical energy consumption against the overall energy budget, and thus to reduce or increase the computations by adjusting QoS to meet specific mission-dependent requirements. In a concrete setup, a mobile robot equipped with `powprofiler` can potentially take advantage of the energy model by adaptively selecting

[1] https://bitbucket.org/adamseew/powprofiler

978-1-7281-5237-0/20/$31.00 ©2020 IEEE
DOI 10.1109/IRC.2020.00028

Seewald et al. (2021) "Coarse-grained computation-oriented energy modeling for heterogeneous parallel embedded systems"

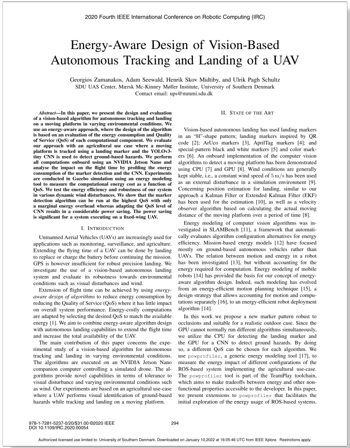
— (2019) "Component-based computation-energy modeling for embedded systems"

— (2020) "Mechanical and computational energy estimation of a fixed-wing drone"

Planning-scheduling in this work (2)

Contributions 4–6

4. Planning via variable coverage motion
5. Guidance on coverage paths
6. In-flight coverage re-planning and scheduling



2020 Fourth IEEE International Conference on Robotic Computing (IRC)

Energy-Aware Design of Vision-Based Autonomous Tracking and Landing of a UAV

Georgios Zamanakos, Adam Seewald, Henrik Skov Midtiby, and Ulrik Pagh Schultz
SDU UAS Center, Mærsk Mc-Kinney Møller Institute, University of Southern Denmark
Contact email: ups@mmmi.sdu.dk

Abstract—**In this paper, we present the design and evaluation of a vision-based algorithm for autonomous tracking and landing on a moving platform in varying environmental conditions. We use an energy-aware approach, where the design of the algorithm is based on an evaluation of the energy consumption and Quality of Service (QoS) of each computational component. We evaluate our approach with an agricultural use case where a moving platform is tracked using a landing marker and the YOLOv3-tiny CNN is used to detect ground-based hazards. We perform all computations onboard using an NVIDIA Jetson Nano and analyse the impact on the flight time by profiling the energy consumption of the marker detection and the CNN. Experiments are conducted in Gazebo simulation using an energy modeling tool to measure the computational energy cost as a function of QoS. We test the energy efficiency and robustness of our system in various dynamic wind disturbances. We show that the marker detection algorithm can be run at the highest QoS with only a marginal energy overhead whereas adapting the QoS level of CNN results in a considerable power saving. The power saving is significant for a system executing on a fixed-wing UAV.**

I. Introduction

Unmanned Aerial Vehicles (UAVs) are increasingly used for applications such as monitoring, surveillance, and agriculture. Extending the flying time of a UAV can be done by landing to replace or charge the battery before continuing the mission. GPS is however insufficient for robust precision landing. We investigate the use of a vision-based autonomous landing system and evaluate its robustness towards environmental conditions such as visual disturbances and wind.

Extension of flight time can be achieved by using *energy-aware design of algorithms* to reduce energy consumption by reducing the Quality of Service (QoS) where it has little impact on overall system performance. Energy-costly computations are adapted by selecting the desired QoS to match the available energy [1]. We aim to combine energy-aware algorithm design with autonomous landing capabilities to extend the flight time and increase the total availability of the UAV.

The main contribution of this paper concerns the experimental study of a vision-based algorithm for autonomous tracking and landing in varying environmental conditions. The algorithms are executed on an NVIDIA Jetson Nano companion computer controlling a simulated drone. The algorithms provide novel capabilities in terms of tolerance to visual disturbance and varying environmental conditions such as wind. Our experiments are based on an agricultural use-case where a UAV performs visual identification of ground-based hazards while tracking and landing on a moving platform.

II. State of the Art

Vision-based autonomous landing has used landing markers in an "H"-shape pattern; landing markers inspired by QR code [2]; ArUco markers [3], AprilTag markers [4]; and special-pattern black and white markers [5] and color markers [6]. An onboard implementation of the computer vision algorithms to detect a moving platform has been demonstrated using CPU [7] and GPU [8]. Wind conditions are generally kept stable, i.e., a constant wind speed of $5\,\mathrm{m/s}$ has been used as an external disturbance in a simulation environment [9]. Concerning position estimation for landing, similar to our approach a Kalman Filter or Extended Kalman Filter (EKF) has been used for the estimation [10], as well as a velocity observer algorithm based on calculating the actual moving distance of the moving platform over a period of time [8].

Energy modeling of computer vision algorithms was investigated in SLAMBench [11], a framework that automatically evaluates algorithm configuration alternatives for energy efficiency. Mission-based energy models [12] have focused mostly on ground-based autonomous vehicles rather than UAVs. The relation between motion and energy in a robot has been investigated [13], but without accounting for the energy required for computation. Energy modeling of mobile robots [14] has provided the basis for our concept of energy-aware algorithm design. Indeed, such modeling has evolved from an energy-efficient motion planning technique [15], a design strategy that allows accounting for motion and computations separately [16], to an energy-efficient robot deployment algorithm [14].

In this work we propose a new marker pattern robust to occlusions and suitable for a realistic outdoor case. Since the GPU cannot normally run different algorithms simultaneously, we utilize the CPU for detecting the landing marker and the GPU for a CNN to detect ground hazards. By doing so, a different QoS can be chosen for each algorithm. We use `powprofiler`, a generic energy modeling tool [17], to measure the energy impact of different configurations of the ROS-based system implementing the agricultural use-case. The `powprofiler` tool is part of the TeamPlay toolchain, which aims to make tradeoffs between energy and other non-functional properties accessible to the developer. In this paper, we present extensions to `powprofiler` that facilitates the initial exploration of the energy usage of ROS-based systems.

978-1-7281-5237-0/20/$31.00 ©2020 IEEE
DOI 10.1109/IRC.2020.00054

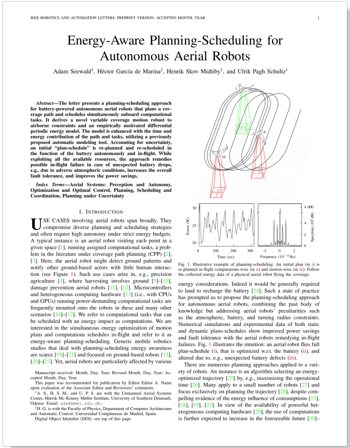
294

Zamanakos et al. (2020) "Energy-aware design of vision-based autonomous tracking and landing of a UAV"

Planning-scheduling in this work (2)

Contributions 4–6

4. Planning via variable coverage motion
5. Guidance on coverage paths
6. In-flight coverage re-planning and scheduling



Energy-Aware Planning-Scheduling for Autonomous Aerial Robots

Adam Seewald[1], Héctor García de Marina[2], Henrik Skov Midtiby[1], and Ulrik Pagh Schultz[1]

Abstract—The letter presents a planning-scheduling approach for battery-powered autonomous aerial robots that plans a coverage path and schedules simultaneously onboard computational tasks. It derives a novel variable coverage motion robust to airborne constraints and an empirically motivated differential periodic energy model. The model is enhanced with the time and energy contribution of the path and tasks, utilizing a previously proposed automatic modeling tool. Accounting for uncertainty, an initial "plan-schedule" is re-planned and re-scheduled in the function of the battery autonomously and in-flight. While exploiting all the available resources, the approach remedies possible in-flight failure in case of unexpected battery drops, e.g., due to adverse atmospheric conditions, increases the overall fault tolerance, and improves the power savings.

Index Terms—Aerial Systems: Perception and Autonomy, Optimization and Optimal Control, Planning, Scheduling and Coordination, Planning under Uncertainty

I. INTRODUCTION

USE CASES involving aerial robots span broadly. They compromise diverse planning and scheduling strategies and often require high autonomy under strict energy budgets. A typical instance is an aerial robot visiting each point in a given space [1], running assigned computational tasks, a problem in the literature under coverage path planning (CPP) [2], [3]. Here, the aerial robot might detect ground patterns and notify other ground-based actors with little human interaction (see Figure 1). Such use cases arise in, e.g., precision agriculture [4], where harvesting involves ground [5]–[10], damage prevention aerial robots [11], [12]. Microcontrollers and heterogeneous computing hardware [13] (i.e., with CPUs and GPUs) running power-demanding computational tasks are frequently mounted onto the robots in these and many other scenarios [14]–[17]. We refer to computational tasks that can be scheduled with an energy impact as computations. We are interested in the simultaneous energy optimization of motion plans and computations schedules in-flight and refer to it as energy-aware planning-scheduling. Generic mobile robotics studies that deal with planning-scheduling energy awareness are scarce [18]–[21] and focused on ground-based robots [13], [20]–[23]. Yet, aerial robots are particularly affected by various

Manuscript received: Month, Day, Year; Revised Month, Day, Year; Accepted Month, Day, Year.
This paper was recommended for publication by Editor Editor A. Name upon evaluation of the Associate Editor and Reviewers' comments.
[1]A. S., H. S. M., and U. P. S. are with the Unmanned Aerial Systems Center, Mærsk Mc-Kinney Møller Institute, University of Southern Denmark, Odense. Email: ads@mmmi.sdu.dk;
[2]H. G. is with the Faculty of Physics, Department of Computer Architecture and Automatic Control, Universidad Complutense de Madrid, Spain.
Digital Object Identifier (DOI): see top of this page.

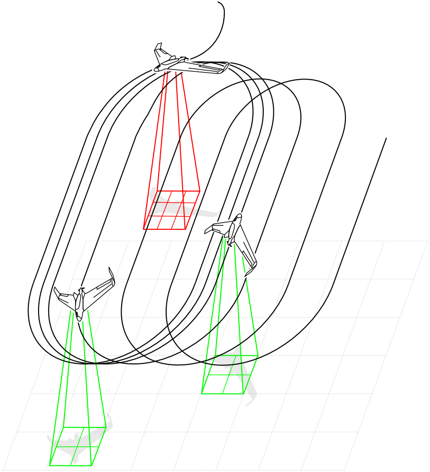
Fig. 1: Illustrative example of planning-scheduling. An initial plan (in i) is re-planned in-flight computations-wise (in ii) and motion-wise (in iii). Follow the collected energy data of a physical aerial robot flying the coverage.

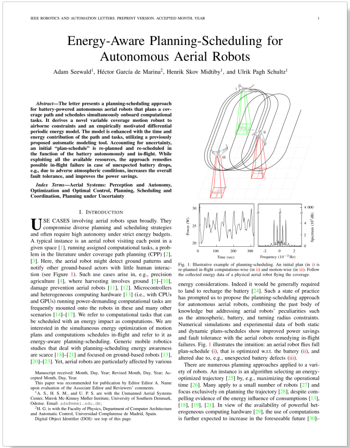
energy considerations. Indeed it would be generally required to land to recharge the battery [24]. Such a state of practice has prompted us to propose the planning-scheduling approach for autonomous aerial robots, combining the past body of knowledge but addressing aerial robots' peculiarities such as the atmospheric, battery, and turning radius constraints. Numerical simulations and experimental data of both static and dynamic plans-schedules show improved power savings and fault tolerance with the aerial robots remedying in-flight failures. Fig. 1 illustrates the intuition: an aerial robot flies full plan-schedule (i), that is optimized w.r.t. the battery (ii), and altered due to, e.g., unexpected battery defects (iii).

There are numerous planning approaches applied to a variety of robots. An instance is an algorithm selecting an energy-optimized trajectory [25] by, e.g., maximizing the operational time [26]. Many apply to a small number of robots [27] and focus exclusively on planning the trajectory [28], despite compelling evidence of the energy influence of consumptions [13], [18], [19], [21]. In view of the availability of powerful heterogeneous computing hardware [29], the use of computations is further expected to increase in the foreseeable future [30]–

Zamanakos et al. (2020) "Energy-aware design of vision-based autonomous tracking and landing of a UAV"
Seewald et al. "Energy-aware planning-scheduling for autonomous aerial robots". In preparation

Intuition behind aerial coverage planning-scheduling





IEEE ROBOTICS AND AUTOMATION LETTERS. PREPRINT VERSION. ACCEPTED MONTH, YEAR

Energy-Aware Planning-Scheduling for Autonomous Aerial Robots

Adam Seewald[1], Héctor García de Marina[2], Henrik Skov Midtiby[1], and Ulrik Pagh Schultz[1]

***Abstract*—The letter presents a planning-scheduling approach for battery-powered autonomous aerial robots that plans a coverage path and schedules simultaneously onboard computational tasks. It derives a novel variable coverage motion robust to airborne constraints and an empirically motivated differential periodic energy model. The model is enhanced with the time and energy contribution of the path and tasks, utilizing a previously proposed automatic modeling tool. Accounting for uncertainty, an initial "plan-schedule" is re-planned and re-scheduled in the function of the battery autonomously and in-flight. While exploiting all the available resources, the approach remedies possible in-flight failure in case of unexpected battery drops, e.g., due to adverse atmospheric conditions, increases the overall fault tolerance, and improves the power savings.**

***Index Terms*—Aerial Systems: Perception and Autonomy, Optimization and Optimal Control, Planning, Scheduling and Coordination, Planning under Uncertainty**

I. Introduction

USE CASES involving aerial robots span broadly. They compromise diverse planning and scheduling strategies and often require high autonomy under strict energy budgets. A typical instance is an aerial robot visiting each point in a given space [1], running assigned computational tasks, a problem in the literature under coverage path planning (CPP) [2], [3]. Here, the aerial robot might detect ground patterns and notify other ground-based actors with little human interaction (see Figure 1). Such use cases arise in, e.g., precision agriculture [4], where harvesting involves ground [5]–[10], damage prevention aerial robots [11], [12]. Microcontrollers and heterogeneous computing hardware [13] (i.e., with CPUs and GPUs) running power-demanding computational tasks are frequently mounted onto the robots in these and many other scenarios [14]–[17]. We refer to computational tasks that can be scheduled with an energy impact as computations. We are interested in the simultaneous energy optimization of motion plans and computations schedules in-flight and refer to it as energy-aware planning-scheduling. Generic mobile robotics studies that deal with planning-scheduling energy awareness are scarce [18]–[21] and focused on ground-based robots [13], [20]–[23]. Yet, aerial robots are particularly affected by various

Manuscript received: Month, Day, Year; Revised Month, Day, Year; Accepted Month, Day, Year.

This paper was recommended for publication by Editor Editor A. Name upon evaluation of the Associate Editor and Reviewers' comments.

[1]A. S., H. S. M., and U. P. S. are with the Unmanned Aerial Systems Center, Mærsk Mc-Kinney Møller Institute, University of Southern Denmark, Odense. Email: ads@mmmi.sdu.dk;

[2]H. G. is with the Faculty of Physics, Department of Computer Architecture and Automatic Control, Universidad Complutense de Madrid, Spain.

Digital Object Identifier (DOI): see top of this page.

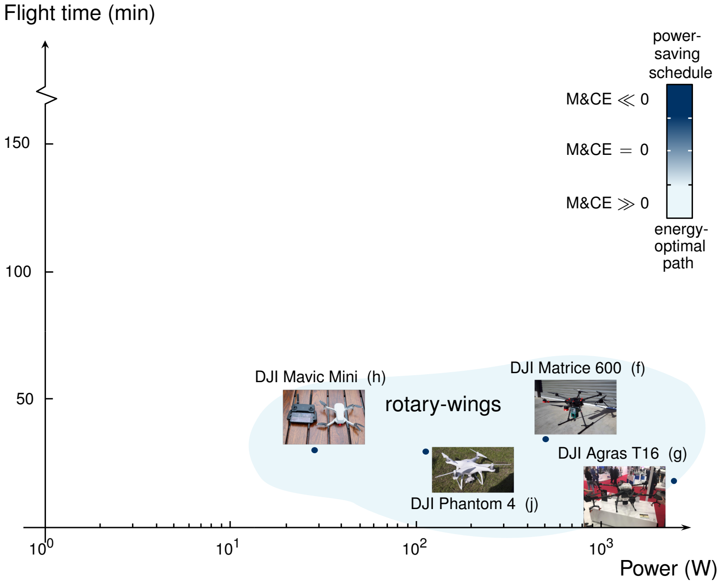
Fig. 1: Illustrative example of planning-scheduling. An initial plan (in i) is re-planned in-flight computations-wise (in ii) and motion-wise (in iii). Follow the collected energy data of a physical aerial robot flying the coverage.

energy considerations. Indeed it would be generally required to land to recharge the battery [24]. Such a state of practice has prompted us to propose the planning-scheduling approach for autonomous aerial robots, combining the past body of knowledge but addressing aerial robots' peculiarities such as the atmospheric, battery, and turning radius constraints. Numerical simulations and experimental data of both static and dynamic plans-schedules show improved power savings and fault tolerance with the aerial robots remedying in-flight failures. Fig. 1 illustrates the intuition: an aerial robot flies full plan-schedule (i), that is optimized w.r.t. the battery (ii), and altered due to, e.g., unexpected battery defects (iii).

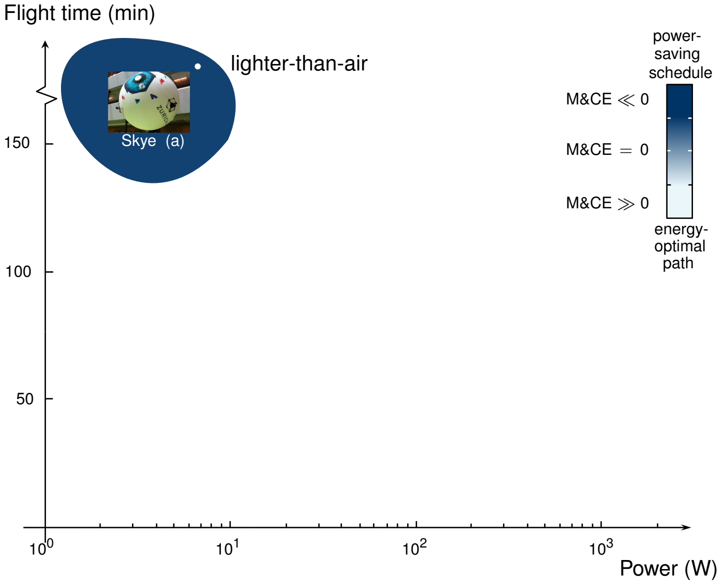
There are numerous planning approaches applied to a variety of robots. An instance is an algorithm selecting an energy-optimized trajectory [25] by, e.g., maximizing the operational time [26]. Many apply to a small number of robots [27] and focus exclusively on planning the trajectory [28], despite compelling evidence of the energy influence of consumptions [13], [18], [19], [21]. In view of the availability of powerful heterogeneous computing hardware [29], the use of computations is further expected to increase in the foreseeable future [30]–

Seewald et al. "Energy-aware planning-scheduling for autonomous aerial robots". In preparation

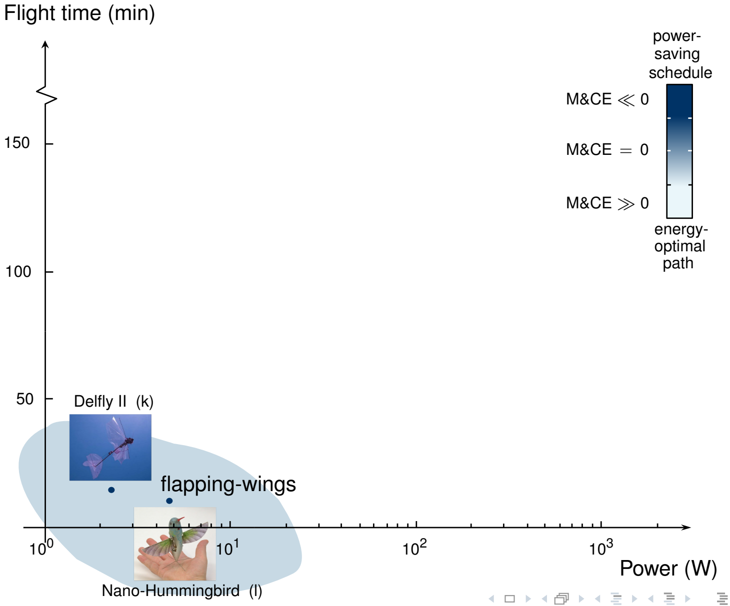
Why fixed wings?



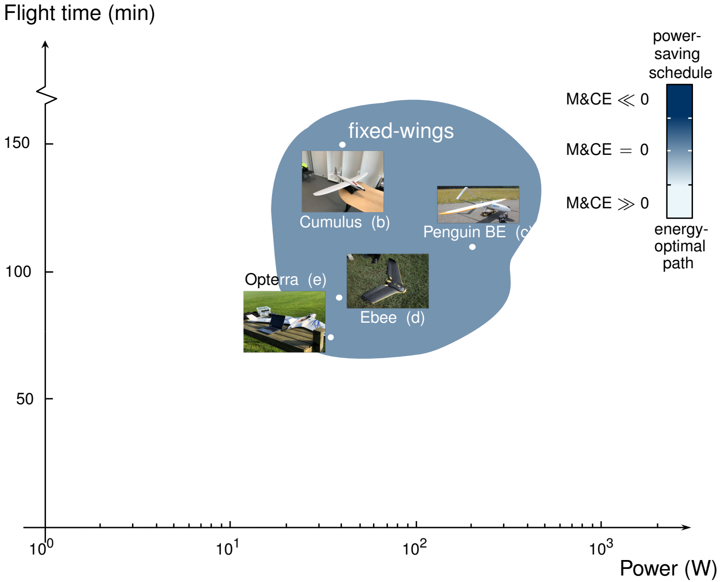
Why fixed wings?



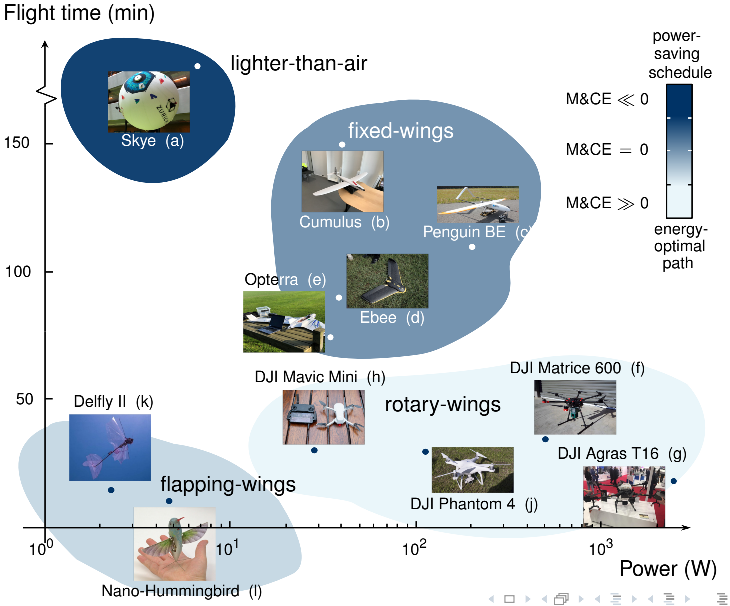
Why fixed wings?



Why fixed wings?



Why fixed wings?



Precision agriculture use case

Objective

1. Cover the field with variable coverage, e.g., alterable pattern
2. Detect ground hazard with variable computations schedule, e.g., CNN
3. Communicate detections, e.g., encryption

- There are no stationary/flying obstacles
- The aerial robot can overfly the edges of the field
- There is at least one available configuration for complete coverage

Precision agriculture use case

Objective

1. Cover the field with variable coverage, e.g., alterable pattern
2. Detect ground hazard with variable computations schedule, e.g., CNN
3. Communicate detections, e.g., encryption

- There are no stationary/flying obstacles
- The aerial robot can overfly the edges of the field
- There is at least one available configuration for complete coverage

Precision agriculture use case

Objective

1. Cover the field with variable coverage, e.g., alterable pattern
2. Detect ground hazard with variable computations schedule, e.g., CNN
3. Communicate detections, e.g., encryption

- There are no stationary/flying obstacles
- The aerial robot can overfly the edges of the field
- There is at least one available configuration for complete coverage

Precision agriculture use case

Objective

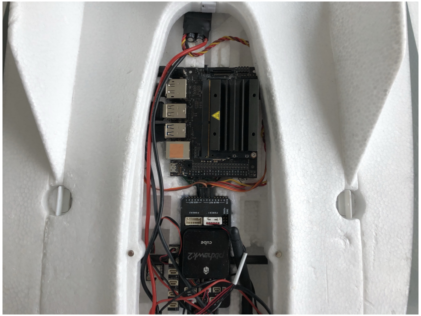
1. Cover the field with variable coverage, e.g., alterable pattern
2. Detect ground hazard with variable computations schedule, e.g., CNN
3. Communicate detections, e.g., encryption

- There are no stationary/flying obstacles
- The aerial robot can overfly the edges of the field
- There is at least one available configuration for complete coverage

Precision agriculture use case (2)

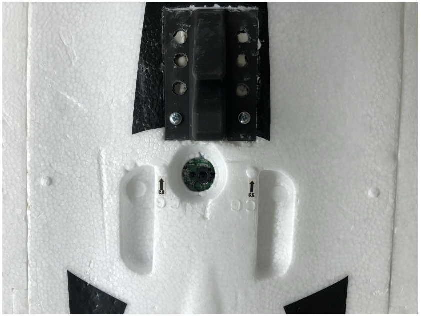
- Opterra fixed-wing aerial robot
- Apogee/v1.0 flight controller
- NVIDIA (R) Jetson Nano (TM) computing hardware
- Pixhawk 4 (R) flight controller
  - interfaces computing hardware via MAVLink
- Downward-facing RGB camera

Precision agriculture use case (2)



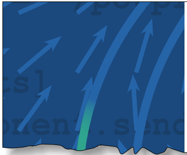
- Opterra fixed-wing aerial robot
- Apogee/v1.0 flight controller
- NVIDIA (R) Jetson Nano (TM) computing hardware
- Pixhawk 4 (R) flight controller
  - interfaces computing hardware via MAVLink
- Downward-facing RGB camera

Precision agriculture use case (2)



- Opterra fixed-wing aerial robot
- Apogee/v1.0 flight controller
- NVIDIA (R) Jetson Nano (TM) computing hardware
- Pixhawk 4 (R) flight controller
  - interfaces computing hardware via MAVLink
- Downward-facing RGB camera

Energy as a limiting factor



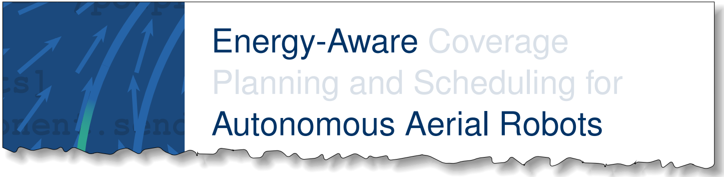
Energy-Aware Coverage Planning and Scheduling for Autonomous Aerial Robots

TEAMPLAY

- Many mobile robots have limited energy storage[4]
- Energy is often a limiting factor to improving computing performance[5]
- Motion and computations energies and battery evolutions all behave differently
  - yet, require accurate modeling to predict future instances

[4] Ondrúška et al. (2015) "Scheduled perception for energy-efficient path following"
[5] Horowitz (2014) "Computing's energy problem (and what we can do about it)"

Energy as a limiting factor

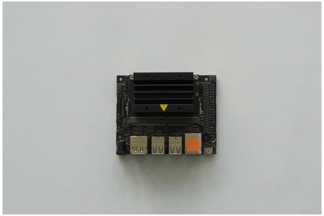


TEAMPLAY

- Many mobile robots have limited energy storage[4]
- Energy is often a limiting factor to improving computing performance[5]
- Motion and computations energies and battery evolutions all behave differently
  - yet, require accurate modeling to predict future instances

[4] Ondrúška et al. (2015) “Scheduled perception for energy-efficient path following”
[5] Horowitz (2014) “Computing’s energy problem (and what we can do about it)”

Energy model for the computations



- Modern computing hardware relies on heterogeneous elements
  - e.g., CPU(s), GPU, microcontroller(s)
  - NVIDIA (R) Jetson TK1, TX2, Nano (TM), ODROID XU3

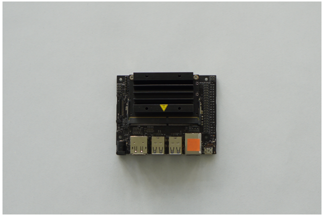
Definition

Set $c_i := \{c_{i,1}, \ldots, c_{i,\rho}, c_{i,\rho+1}, c_{i,\rho+\sigma}\}$ contains **parameters** to alter the coverage (c_i^ρ) and schedule (c_i^σ) within given bounds, i.e., different schedules $c_i^\sigma(t_j) \neq c_i^\sigma(t_{j+1})$ have different computations energies

- Precision agriculture use case
  - Variable coverage via a parameter $c_i^\rho := c_1$
  - Variable CNN schedule and encryption via two parameters $c_i^\sigma := \{c_2, c_3\}$

Energy model for the computations

- Modern computing hardware relies on heterogeneous elements
  - e.g., CPU(s), GPU, microcontroller(s)
  - NVIDIA (R) Jetson TK1, TX2, Nano (TM), ODROID XU3



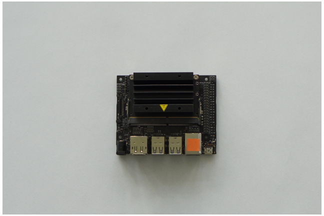
Definition

Set $c_i := \{c_{i,1}, \ldots, c_{i,\rho}, c_{i,\rho+1}, c_{i,\rho+\sigma}\}$ contains **parameters** to alter the coverage (c_i^ρ) and schedule (c_i^σ) within given bounds, i.e., different schedules $c_i^\sigma(t_j) \neq c_i^\sigma(t_{j+1})$ have different computations energies

- Precision agriculture use case
  - Variable coverage via a parameter $c_i^\rho := c_1$
  - Variable CNN schedule and encryption via two parameters $c_i^\sigma := \{c_2, c_3\}$

Energy model for the computations

- Modern computing hardware relies on heterogeneous elements
  - e.g., CPU(s), GPU, microcontroller(s)
  - NVIDIA (R) Jetson TK1, TX2, Nano (TM), ODROID XU3



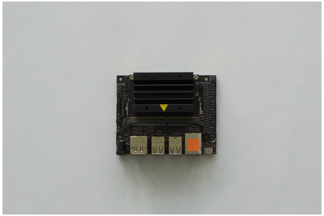
Definition

Set $c_i := \{c_{i,1}, \ldots, c_{i,\rho}, c_{i,\rho+1}, c_{i,\rho+\sigma}\}$ contains **parameters** to alter the coverage (c_i^ρ) and schedule (c_i^σ) within given bounds, i.e., different schedules $c_i^\sigma(t_j) \neq c_i^\sigma(t_{j+1})$ have different computations energies

- Precision agriculture use case
  - Variable coverage via a parameter $c_i^\rho := c_1$
  - Variable CNN schedule and encryption via two parameters $c_i^\sigma := \{c_2, c_3\}$

Energy model for the computations

- Modern computing hardware relies on heterogeneous elements
  - e.g., CPU(s), GPU, microcontroller(s)
  - NVIDIA (R) Jetson TK1, TX2, Nano (TM), ODROID XU3



Definition

Set $c_i := \{c_{i,1}, \ldots, c_{i,\rho}, c_{i,\rho+1}, c_{i,\rho+\sigma}\}$ contains **parameters** to alter the coverage (c_i^ρ) and schedule (c_i^σ) within given bounds, i.e., different schedules $c_i^\sigma(t_j) \neq c_i^\sigma(t_{j+1})$ have different computations energies

- Precision agriculture use case
  - Variable coverage via a parameter $c_i^\rho := c_1$
  - Variable CNN schedule and encryption via two parameters $c_i^\sigma := \{c_2, c_3\}$

Energy model for the computations (2)

- Models in the literature rely on analytical expression, regressional analysis, etc.
  - select software- or hardware-specific parameters
- Ours[6] relies on regressional analysis and predicts energy per each c_i^σ
- Uses a two-layer architecture–measurement/predictive

Definitions

1. Given c_i^σ, measuring device, $\mathcal{T} := [t_0, t_f]$, the **measurement layer** $\mathbf{g} : \mathbb{Z}_{>0} \times \mathbb{Z}^\sigma \times \mathcal{T} \to \mathbb{R}^3$ returns power, energy, and SoC
2. Given measuring device, $\mathcal{T}$, the **predictive layer** $g : \mathbb{Z}_{>0} \times \mathbb{Z}^\sigma \to \mathbb{R}^3$ returns power, energy, and SoC for any c_i^σ

[6] Seewald et al. (2021) “Coarse-grained computation-oriented energy modeling for heterogeneous parallel embedded systems”

Energy model for the computations (2)

- Models in the literature rely on analytical expression, regressional analysis, etc.
  - select software- or hardware-specific parameters
- Ours[6] relies on regressional analysis and predicts energy per each c_i^σ
- Uses a two-layer architecture–measurement/predictive

Definitions

1. Given c_i^σ, measuring device, $\mathcal{T} := [t_0, t_f]$, the **measurement layer** $\mathbf{g} : \mathbb{Z}_{>0} \times \mathbb{Z}^\sigma \times \mathcal{T} \to \mathbb{R}^3$ returns power, energy, and SoC
2. Given measuring device, $\mathcal{T}$, the **predictive layer** $g : \mathbb{Z}_{>0} \times \mathbb{Z}^\sigma \to \mathbb{R}^3$ returns power, energy, and SoC for any c_i^σ

[6] Seewald et al. (2021) "Coarse-grained computation-oriented energy modeling for heterogeneous parallel embedded systems"

The `powprofiler` tool

- Automated profiling and modeling
- Implements both layers
  - generates measurement layers from a discrete set of measurements
  - predictive layer with linear regression between consecutive measurement layers
- Configuration specification specifies c_i^σ and bounds

```
[settings]
frequency = 10
h =         0.01
directory = /data/matrix-exp

[components]
  [component.matrix-exp]
  src =       matrix-exp
  runtime = 5000
  range =    256, 4096, pow(2)
  range =    20, 60, 10
```

Seewald et al. (2021) "`powprofiler` computations energy modeling tool"

The `powprofiler` tool

- Automated profiling and modeling
- Implements both layers
  - generates measurement layers from a discrete set of measurements
  - predictive layer with linear regression between consecutive measurement layers
- Configuration specification specifies c_i^σ and bounds

```
[settings]
frequency = 10
h =         0.01
directory = /data/matrix-exp

[components]
  [component.matrix-exp]
  src =     matrix-exp
  runtime = 5000
  range =   256, 4096, pow(2)
  range =   20, 60, 10
```

Seewald et al. (2021) "`powprofiler` computations energy modeling tool"

The `powprofiler` tool

- Automated profiling and modeling
- Implements both layers
  - generates measurement layers from a discrete set of measurements
  - predictive layer with linear regression between consecutive measurement layers
- Configuration specification specifies c_i^σ and bounds

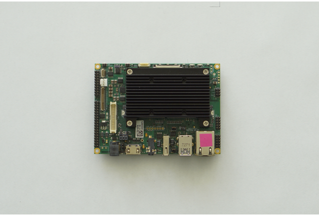
```
[settings]
frequency = 10
h =         0.01
directory = /data/matrix-exp

[components]
  [component.matrix-exp]
  src =       matrix-exp
  runtime = 5000
  range =    256, 4096, pow(2)
  range =    20, 60, 10
```

Seewald et al. (2021) “`powprofiler` computations energy modeling tool”

The `powprofiler` tool (2)

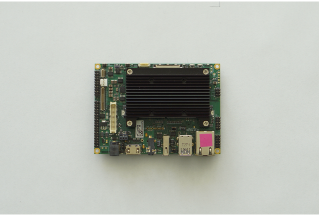
- Works with computing hardware with/ without measuring device
- Can be used with other measuring devices
  - creating a class that inherits from `sampler` functions `get_sample`, `dryrun`
- Supports ROS middleware[7]
- Can be imported as a library
  - in C/C++ adding directive `#include` with `<powprof/async.h>`



[7] Zamanakos et al. (2020) "Energy-aware design of vision-based autonomous tracking and landing of a UAV"

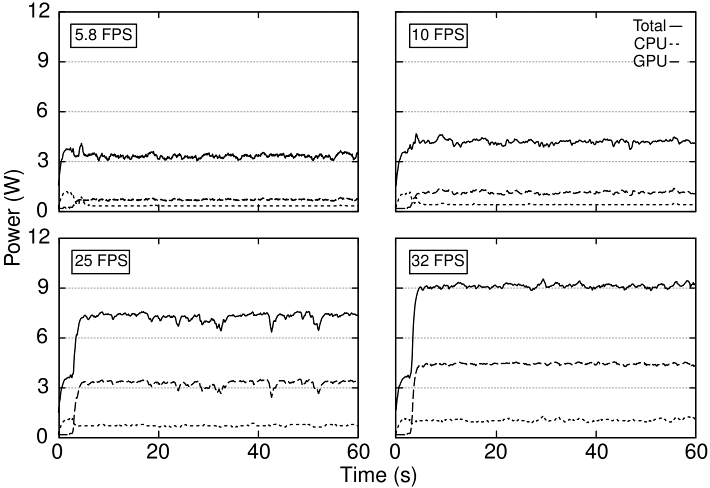
The `powprofiler` tool (2)

- Works with computing hardware with/ without measuring device
- Can be used with other measuring devices
  - creating a class that inherits from `sampler` functions `get_sample`, `dryrun`
- Supports ROS middleware[7]
- Can be imported as a library
  - in C/C++ adding directive `#include` with `<powprof/async.h>`



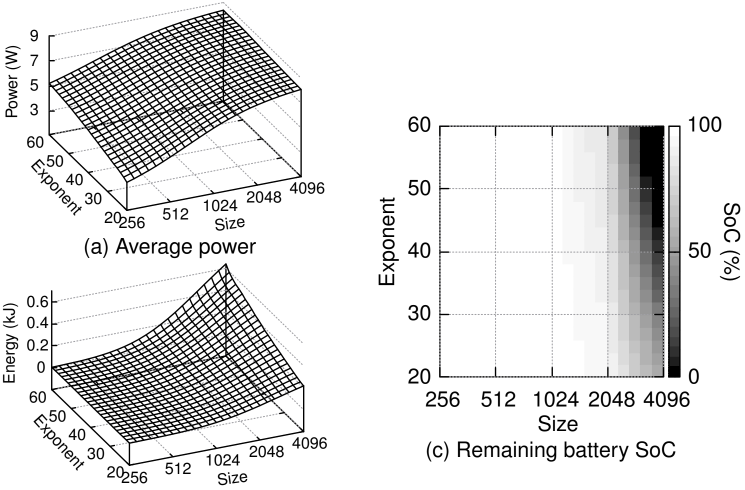
[7] Zamanakos et al. (2020) "Energy-aware design of vision-based autonomous tracking and landing of a UAV"

The measurement layer



Seewald et al. (2021) “Coarse-grained computation-oriented energy modeling for heterogeneous parallel embedded systems”

The predictive layer



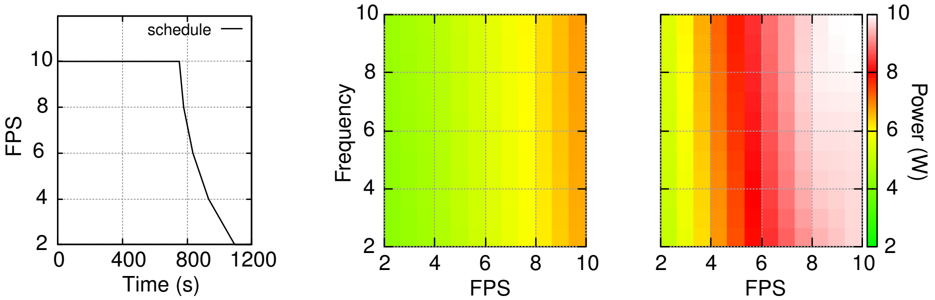
(a) Average power

(b) Overall energy

(c) Remaining battery SoC

Seewald et al. (2021) “Coarse-grained computation-oriented energy modeling for heterogeneous parallel embedded systems”

The predictive layer (2)



(a) Schedule over time (b) `sender`, `ssd-mobilenet` (c) `sender`, `pednet`

Energy model for the motion

- Relies on the energy characteristics of the coverage

Definitions

1. A **plan** Γ is FSM that contains stages $\Gamma_i := \{\varphi_i(\mathbf{p}, c_i^\rho), c_i^\sigma\}$ with parameters $c_i := \{c_i^\rho, c_i^\sigma\}$ and a “function to follow” φ_i
2. **Stage** Γ_i terminates on the **triggering point** $\mathbf{p}_{\Gamma_i}$, the plan on the **final point** $\mathbf{p}_{\Gamma_l}$

- A convenient way to define Γ is to iterate a set of stages $\Gamma_1, \ldots, \Gamma_n$
  - resulting in a periodic energy behavior

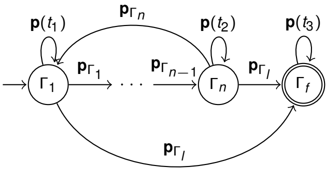
Energy model for the motion

- Relies on the energy characteristics of the coverage

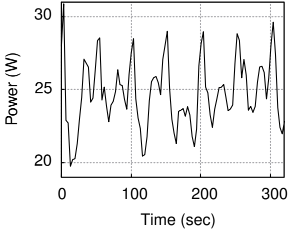
Definitions

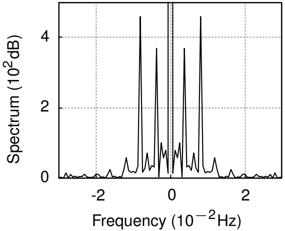
1. A **plan** Γ is FSM that contains stages $\Gamma_i := \{\varphi_i(\mathbf{p}, c_i^\rho), c_i^\sigma\}$ with parameters $c_i := \{c_i^\rho, c_i^\sigma\}$ and a "function to follow" φ_i
2. **Stage** Γ_i terminates on the **triggering point** $\mathbf{p}_{\Gamma_i}$, the plan on the **final point** $\mathbf{p}_{\Gamma_l}$

- A convenient way to define Γ is to iterate a set of stages $\Gamma_1, \ldots, \Gamma_n$
  - resulting in a periodic energy behavior



Energy model for the motion (2)



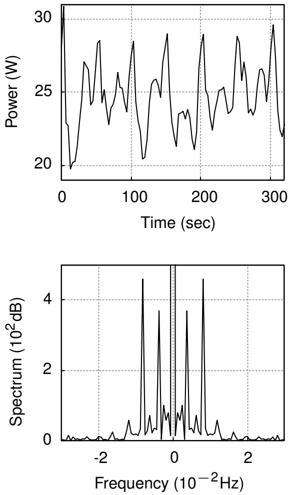


- With periodic energy behaviors
- Model via Fourier series of order r and period T

$$h(t) = a_0/T + (2/T)\sum_{j=1}^{r}\left(a_j \cos \omega j t + b_j \sin \omega j t\right)$$

  - $\omega := 2\pi/T$ angular frequency, a_0, a_j, b_j coefficients
- ODE with periodic solution to account for c_i

Energy model for the motion (2)



- With periodic energy behaviors
- Model via Fourier series of order r and period T

$$h(t) = a_0/T + (2/T)\sum_{j=1}^{r}\left(a_j \cos \omega j t + b_j \sin \omega j t\right)$$

  - $\omega := 2\pi/T$ angular frequency, a_0, a_j, b_j coefficients

- ODE with periodic solution to account for c_i

Modeling the periodic energy behavior

- Equivalent ODE with ω, T, r from the series

$$\begin{aligned}\dot{\mathbf{q}}(t) &= A\mathbf{q}(t) + B\mathbf{u}(t)\\ y(t) &= C\mathbf{q}(t)\end{aligned} \qquad A = \begin{bmatrix} 0 & 0^{1\times 2} & \dots & 0^{1\times 2} \\ 0^{2\times 1} & A_1 & \dots & 0^{2\times 2} \\ \vdots & \vdots & \ddots & \vdots \\ 0^{2\times 1} & 0^{2\times 2} & \dots & A_r \end{bmatrix}$$

$$A_j := \begin{bmatrix} 0 & \omega j \\ -\omega j & 0 \end{bmatrix} \qquad C = (1/T)\begin{bmatrix} 1 & 1 & 0 & \cdots & 1 & 0 \end{bmatrix}$$

- States are energy coefficients similar to a_0, a_j, b_j

$$\mathbf{q}(t) = \begin{bmatrix} \alpha_0(t) & \alpha_1(t) & \beta_1(t) & \cdots & \alpha_r(t) & \beta_r(t) \end{bmatrix}'$$

- Under some conditions the series and ODE are equal (Lemma 4.3.1)

Modeling the periodic energy behavior (2)

- $\mathbf{u}, B$ exploit the concept of stage $\Gamma_i := \{\varphi_i(\mathbf{p}, c_i^\rho), c_i^\sigma\}$

Observation

Given c_i at following time steps, a change in c_i results in different overall and instantaneous energies for c_i^ρ and c_i^σ respectively

- Scaling factors transform parameters into time and energy domains
  - from computations parameters to energy rely on g in `powprofiler`
  - $\mathbf{u}$ is then the change in the energy domain and B includes $\mathbf{u}$ into $\mathbf{q}$

Modeling the periodic energy behavior (2)

- $\mathbf{u}$, B exploit the concept of stage $\Gamma_i := \{\varphi_i(\mathbf{p}, c_i^\rho), c_i^\sigma\}$

Observation

Given c_i at following time steps, a change in c_i results in different overall and instantaneous energies for c_i^ρ and c_i^σ respectively

- Scaling factors transform parameters into time and energy domains
  - from computations parameters to energy rely on g in `powprofiler`
  - $\mathbf{u}$ is then the change in the energy domain and B includes $\mathbf{u}$ into $\mathbf{q}$

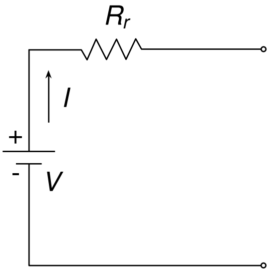
Battery model

- Predicts the state of aerial robot's li-ion battery in-flight
- Multiple models in the literature with varying complexity, accuracy, and ease of implementation
  - e.g., physical, hybrid, empirical, mixed, and abstract models
- "Rint" abstract ECM model with a given internal battery voltage (V) one resistor (R_r)[8]

[8] Mousavi et al. (2014) "Various battery models for various simulation studies and applications"

Battery model

- Predicts the state of aerial robot's li-ion battery in-flight
- Multiple models in the literature with varying complexity, accuracy, and ease of implementation
  - e.g., physical, hybrid, empirical, mixed, and abstract models
- "Rint" abstract ECM model with a given internal battery voltage (V) one resistor (R_r)[8]



[8] Mousavi et al. (2014) "Various battery models for various simulation studies and applications"

Battery model (2)

- Solving the “Rint” ECM in terms of internal load leads to

$$I = (V - \sqrt{V^2 - 4R_r y})/(2R_r)$$

- SoC changes when computations and motion require a current to be drained, i.e., $\dot{b} = -I/Q_c$[9]
  - allows to incorporate computations and motion models

Definition

Given internal battery voltage V, **output constraint** is $\mathcal{Y} := \{y \mid y \in [0, bQ_c V]\}$, bounded by maximum instantaneous energy consumption

- “Rint” ECM is implemented in the `powprofiler` tool

[9] Zhang et al. (2018) “Online estimation of battery equivalent circuit model parameters and state of charge using decoupled least squares technique”

Battery model (2)

- Solving the "Rint" ECM in terms of internal load leads to

$$I = (V - \sqrt{V^2 - 4R_r y})/(2R_r)$$

- SoC changes when computations and motion require a current to be drained, i.e., $\dot{b} = -I/Q_c$[9]
  - allows to incorporate computations and motion models

Definition

Given internal battery voltage V, **output constraint** is $\mathcal{Y} := \{y \mid y \in [0, bQ_c V]\}$, bounded by maximum instantaneous energy consumption

- "Rint" ECM is implemented in the `powprofiler` tool

[9] Zhang et al. (2018) "Online estimation of battery equivalent circuit model parameters and state of charge using decoupled least squares technique"

Battery model (2)

- Solving the "Rint" ECM in terms of internal load leads to

$$I = (V - \sqrt{V^2 - 4R_r y})/(2R_r)$$

- SoC changes when computations and motion require a current to be drained, i.e., $\dot{b} = -I/Q_c$[9]
  - allows to incorporate computations and motion models

Definition

Given internal battery voltage V, **output constraint** is $\mathcal{Y} := \{y \mid y \in [0, bQ_c V]\}$, bounded by maximum instantaneous energy consumption

- "Rint" ECM is implemented in the `powprofiler` tool

[9] Zhang et al. (2018) "Online estimation of battery equivalent circuit model parameters and state of charge using decoupled least squares technique"

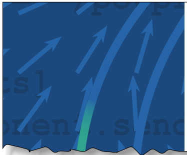
Coverage path planning

Energy-Aware **Coverage Planning** and Scheduling **for** Autonomous **Aerial Robots**

- Many approaches in the literature to cover each point in a given space
- Complete approaches divide the free space into sub-regions[10]
  - i.e., cellular decomposition
- Derive a coverage motion for each cell
  - intuitive motion is going back-and-forth, i.e., boustrophedon motion
  - turn preserving motion, i.e., Zamboni motion

[10] Choset (2001) "Coverage for robotics–A survey of recent results"

Coverage path planning



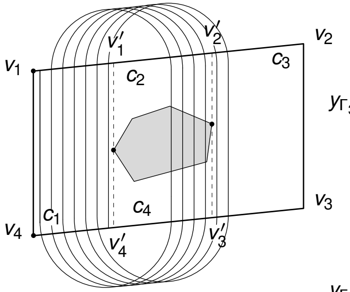
Energy-Aware Coverage Planning and Scheduling for Autonomous Aerial Robots

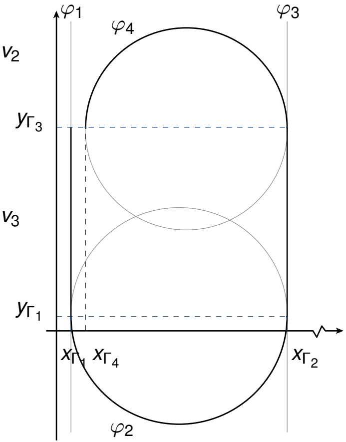
- Many approaches in the literature to cover each point in a given space
- Complete approaches divide the free space into sub-regions[10]
  - i.e., cellular decomposition
- Derive a coverage motion for each cell
  - intuitive motion is going back-and-forth, i.e., boustrophedon motion
  - turn preserving motion, i.e., Zamboni motion

[10] Choset (2001) “Coverage for robotics–A survey of recent results”

Coverage path planning (2)

- Zamboni-like motion with variable coverage
- Change in φ_4 alters the coverage

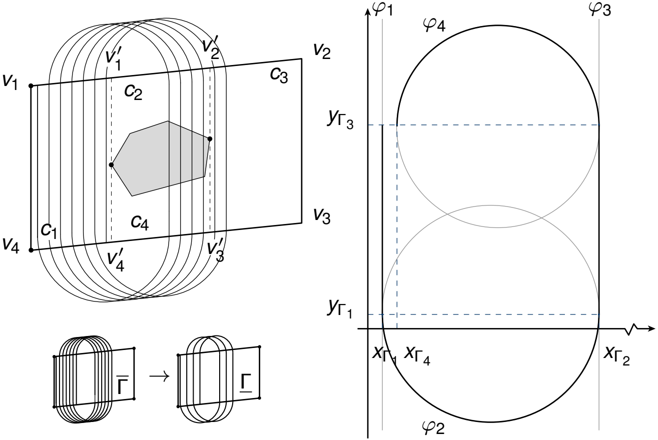




Seewald et al. “Energy-aware planning-scheduling for autonomous aerial robots”. In preparation

Coverage path planning (2)

- Zamboni-like motion with variable coverage
- Change in φ_4 alters the coverage



Seewald et al. “Energy-aware planning-scheduling for autonomous aerial robots”. In preparation

Guidance on the coverage

- Guide the aerial robot on the coverage path,
  - i.e., direction in each point to the contours of $\varphi_1, \varphi_2, \ldots$
- Vector field assigns a vector at each point $\mathbf{p}$
- Exploits a path-following vector field in the literature[11]

$$\Delta_d := E_i \nabla \varphi_i - k_e \varphi_i \nabla \varphi_i, \quad E_i = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

  - k_e the speed of convergence

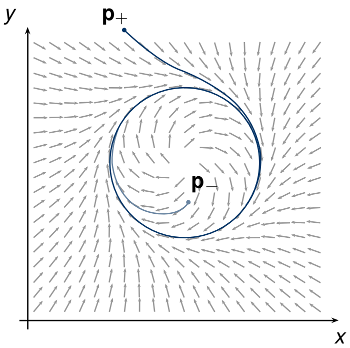
[11] García de Marina et al. (2017) "Guidance algorithm for smooth trajectory tracking of a fixed wing UAV flying in wind flows"

Guidance on the coverage

- Guide the aerial robot on the coverage path,
  - i.e., direction in each point to the contours of $\varphi_1, \varphi_2, \ldots$
- Vector field assigns a vector at each point $\mathbf{p}$
- Exploits a path-following vector field in the literature[11]

$$\Delta_d := E_i \nabla \varphi_i - k_e \varphi_i \nabla \varphi_i, \quad E_i = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

  - k_e the speed of convergence



[11] García de Marina et al. (2017) "Guidance algorithm for smooth trajectory tracking of a fixed wing UAV flying in wind flows"

Re-planning

- Returns the trajectory of c_i in Γ on $\mathcal{T} := [t_0, t_f]$ in-flight with varying conditions
- States $\mathbf{q}$ and final time instants t_f are unknown
  - Many past approaches adopt optimization on finite horizon (i.e., Brateman et al., Zhang et al., Ondrúška et al., Lahijanian et al.)
  - Output MPC for in-flight re-planning of Γ
- Remedy battery defects in, e.g., precision agriculture use case

Optimal control problem

- OCP that returns the trajectory of c_i
- Using finite horizon $t \in [t_0, T]$ with $T = t_0 + N$ for given N

$$\min_{\mathbf{q}(t), c_i(t)} l_f(\mathbf{q}(T), T) + \int_{t_0}^{T} l(\mathbf{q}(t), c_i(t), t)\, dt$$

$$\begin{aligned}
\text{s.t. } \dot{\mathbf{q}}(t) &= A\mathbf{q}(t) + B\mathbf{u}(t) \\
c_1(t) &\in [-1k, 0] \quad c_2(t) \in [2, 10] \quad c_3(t) \in \{0, 1\} \\
y(t) &\in \mathcal{Y}(t) \\
\mathbf{q}(t_0) &= \hat{\mathbf{q}}_0 \text{ given (last estimate state)} \\
b(t_0) &= b_0 \text{ given}
\end{aligned}$$

- $l(\mathbf{q}(t), c_i(t), t) = \mathbf{q}'(t)Q\mathbf{q}(t) + c'(t)Rc_i(t)$ is instantaneous cost
- $l_f(\mathbf{q}(T), T) = \mathbf{q}'(T)Q_f\mathbf{q}(T)$ is final cost
- Q, R, Q_f are positive semi-definite

Optimal control problem

- OCP that returns the trajectory of c_i
- Using finite horizon $t \in [t_0, T]$ with $T = t_0 + N$ for given N

$$\min_{\mathbf{q}(t), c_i(t)} l_f(\mathbf{q}(T), T) + \int_{t_0}^{T} l(\mathbf{q}(t), c_i(t), t)\, dt$$

$$\text{s.t. } \dot{\mathbf{q}}(t) = A\mathbf{q}(t) + B\mathbf{u}(t)$$

$$c_1(t) \in [-1k, 0] \quad c_2(t) \in [2, 10] \quad c_3(t) \in \{0, 1\}$$

$$y(t) \in \mathcal{Y}(t)$$

$$\mathbf{q}(t_0) = \hat{\mathbf{q}}_0 \text{ given (last estimate state)}$$

$$b(t_0) = b_0 \text{ given}$$

- $l(\mathbf{q}(t), c_i(t), t) = \mathbf{q}'(t)Q\mathbf{q}(t) + c'(t)Rc_i(t)$ is instantaneous cost
- $l_f(\mathbf{q}(T), T) = \mathbf{q}'(T)Q_f\mathbf{q}(T)$ is final cost
- Q, R, Q_f are positive semi-definite

Model predictive controller

- Quadratic expressions for l, l_f in the NLP
  - e.g., square of the control
  - practically, costs are negative to exploit all the resources within constraints
- Small discretization step h
  - 1/100 of a second (Euler method in $\mathcal{K}$, Runge-Kutta method in `powprofiler`)
  - different h in $\mathcal{K}$ for costs, e.g., updates the trajectory of c_i each second
- Horizon N 6 and 10 seconds
  - similar to MPC implementations in aerial robotics, e.g., 14 in Gavilan et al., 10 and 40 in Kang et al., 2 to 8 in Stastny et al., 5 in Chao et al.
- CasADi framework and IPOPT solver[12]
  - direct multiple shooting to transcribe OCP into NLP
- Linear Kalman filter derives $\hat{\mathbf{q}}_0$, flight controller b_0

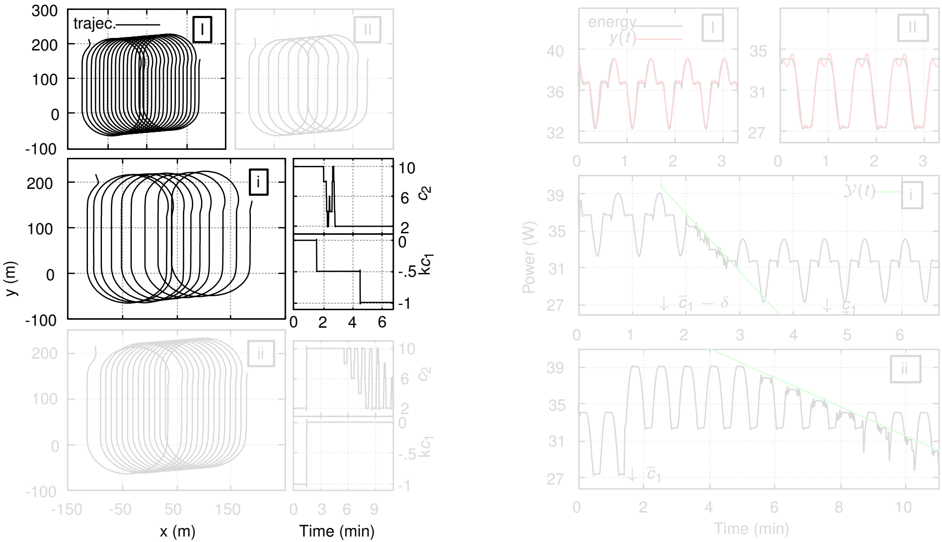
[12] Andersson et al. (2018) "CasADi: a software framework for nonlinear optimization and optimal control"

Model predictive controller

- Quadratic expressions for l, l_f in the NLP
  - e.g., square of the control
  - practically, costs are negative to exploit all the resources within constraints
- Small discretization step h
  - 1/100 of a second (Euler method in $\mathcal{K}$, Runge-Kutta method in `powprofiler`)
  - different h in $\mathcal{K}$ for costs, e.g., updates the trajectory of c_i each second
- Horizon N 6 and 10 seconds
  - similar to MPC implementations in aerial robotics, e.g., 14 in Gavilan et al., 10 and 40 in Kang et al., 2 to 8 in Stastny et al., 5 in Chao et al.
- CasADi framework and IPOPT solver[12]
  - direct multiple shooting to transcribe OCP into NLP
- Linear Kalman filter derives $\hat{\mathbf{q}}_0$, flight controller b_0

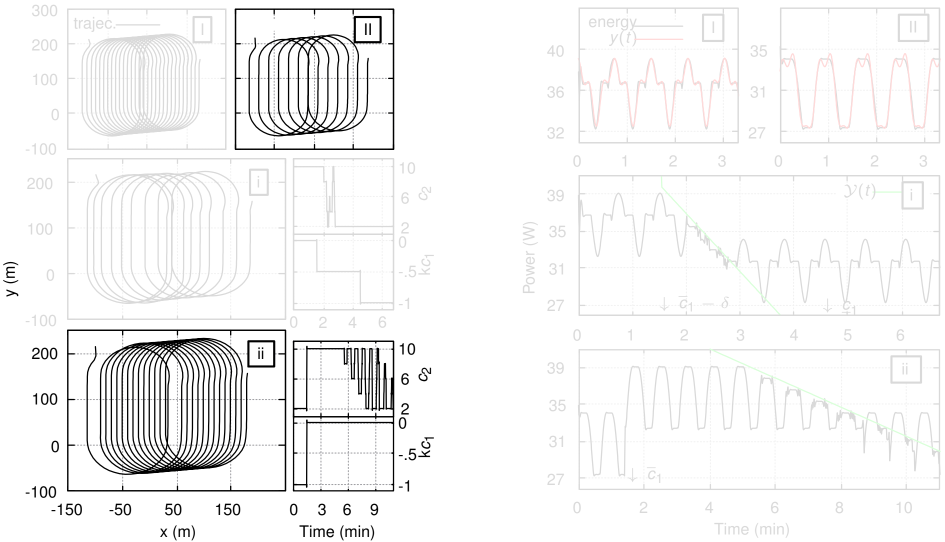
[12] Andersson et al. (2018) "CasADi: a software framework for nonlinear optimization and optimal control"

Numerical simulations



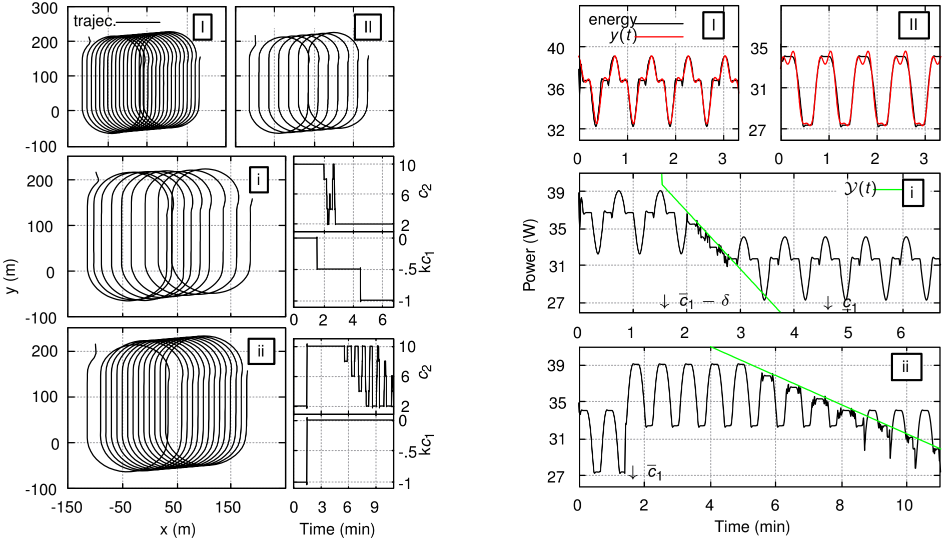
Seewald et al. “Energy-aware planning-scheduling for autonomous aerial robots”. In preparation

Numerical simulations



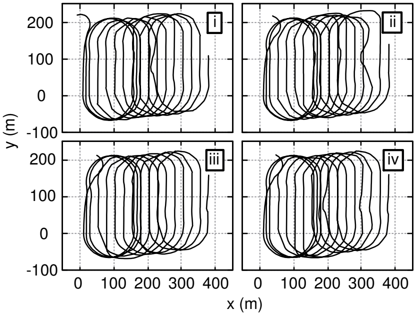
Seewald et al. "Energy-aware planning-scheduling for autonomous aerial robots". In preparation

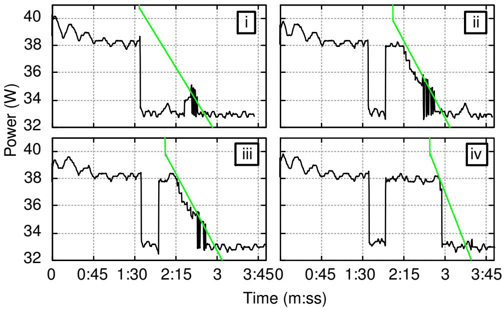
Numerical simulations



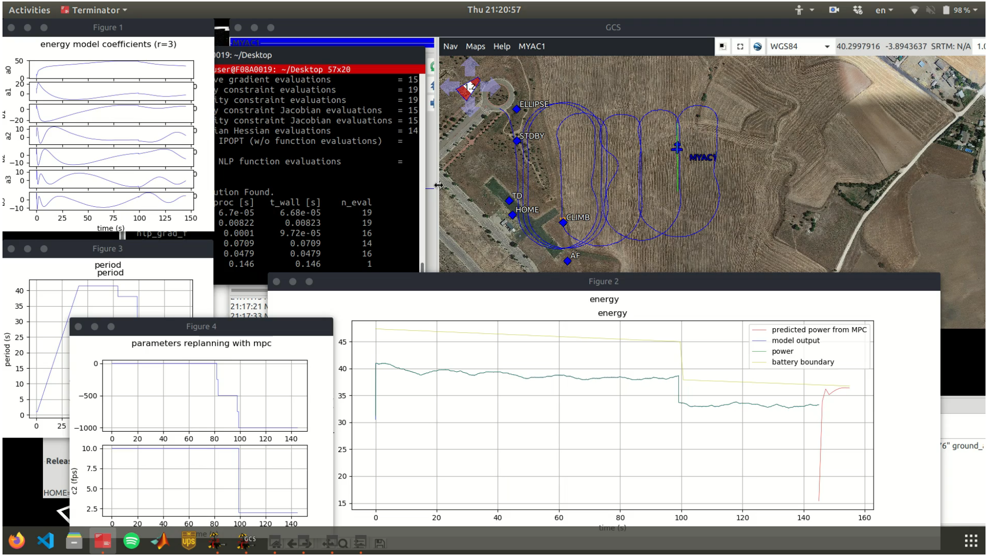
Seewald et al. “Energy-aware planning-scheduling for autonomous aerial robots”. In preparation

Paparazzi flight controller





Paparazzi flight controller (2)



Conclusions and future directions

- Computations energy and battery models based on `powprofiler`[13]
- Cohesive model for the motion that incorporates computations[14]
- Zamboni-like motion for variable coverage with constrained aerial robots
- Aerial coverage planning-scheduling for, e.g., precision agriculture[15]
- "*Planning-scheduling for aerial robots remedies possible in-flight failures, increases fault tolerance, and improves the power savings*"

[13] Seewald et al. (2021) "Coarse-grained computation-oriented energy modeling for heterogeneous parallel embedded systems"

[14] — (2020) "Mechanical and computational energy estimation of a fixed-wing drone"

[15] — "Energy-aware planning-scheduling for autonomous aerial robots". In preparation

Conclusions and future directions

- Computations energy and battery models based on `powprofiler`[13]
- Cohesive model for the motion that incorporates computations[14]
- Zamboni-like motion for variable coverage with constrained aerial robots
- Aerial coverage planning-scheduling for, e.g., precision agriculture[15]
- "*Planning-scheduling for aerial robots remedies possible in-flight failures, increases fault tolerance, and improves the power savings*"

[13] Seewald et al. (2021) "Coarse-grained computation-oriented energy modeling for heterogeneous parallel embedded systems"
[14] — (2020) "Mechanical and computational energy estimation of a fixed-wing drone"
[15] — "Energy-aware planning-scheduling for autonomous aerial robots". In preparation

Conclusions and future directions

- Computations energy and battery models based on `powprofiler`[13]
- Cohesive model for the motion that incorporates computations[14]
- Zamboni-like motion for variable coverage with constrained aerial robots
- Aerial coverage planning-scheduling for, e.g., precision agriculture[15]
- *"Planning-scheduling for aerial robots remedies possible in-flight failures, increases fault tolerance, and improves the power savings"*

[13] Seewald et al. (2021) "Coarse-grained computation-oriented energy modeling for heterogeneous parallel embedded systems"

[14] — (2020) "Mechanical and computational energy estimation of a fixed-wing drone"

[15] — "Energy-aware planning-scheduling for autonomous aerial robots". In preparation

Conclusions and future directions

- Computations energy and battery models based on `powprofiler`[13]
- Cohesive model for the motion that incorporates computations[14]
- Zamboni-like motion for variable coverage with constrained aerial robots
- Aerial coverage planning-scheduling for, e.g., precision agriculture[15]
- "*Planning-scheduling for aerial robots remedies possible in-flight failures, increases fault tolerance, and improves the power savings*"

---

[13] Seewald et al. (2021) "Coarse-grained computation-oriented energy modeling for heterogeneous parallel embedded systems"

[14] — (2020) "Mechanical and computational energy estimation of a fixed-wing drone"

[15] — "Energy-aware planning-scheduling for autonomous aerial robots". In preparation