TÉCNICO
LISBOA

UNIVERSIDADE DE LISBOA

INSTITUTO SUPERIOR TÉCNICO

Inertial uncertainty reduction through parameter estimation for microgravity free-flyer operations

Monica Jeevan Ekal

Supervisor: Doctor Rodrigo Martins de Matos Ventura

Thesis approved in public session to obtain the PhD Degree in

Electrical and Computer Engineering

Jury final classification: Pass with Distinction

2024

UNIVERSIDADE DE LISBOA

INSTITUTO SUPERIOR TÉCNICO

Inertial uncertainty reduction through parameter estimation for microgravity free-flyer operations

Monica Jeevan Ekal

Supervisor: Doctor Rodrigo Martins de Matos Ventura

Thesis approved in public session to obtain the PhD Degree in

Electrical and Computer Engineering

Jury final classification: Pass with Distinction

Jury

Chairperson: Doctor Paulo Jorge Coelho Ramalho Oliveira, Instituto Superior Técnico, Universidade de Lisboa

Members of the Committee:

Doctor António Pedro Rodrigues Aguiar, Faculdade de Engenharia, Universidade do Porto
Doctor José Alberto Rosado dos Santos Victor, Instituto Superior Técnico, Universidade de Lisboa
Doctor Zachary Manchester, Robotics Institute, Carnegie Mellon University, EUA
Doctor Rodrigo Martins de Matos Ventura, Instituto Superior Técnico, Universidade de Lisboa

Funding Institution: Fundação para a Ciência e a Tecnologia (FCT)

2024

Resumo

Os free-flyers robóticos são relevantes para futuras missões espaciais, para tarefas de manutenção em órbita, como construção e reparação. O conhecimento preciso da dinâmica é essencial para um planeamento óptimo e uma execução segura de tarefas. A execução de tarefas envolve incerteza quanto às propriedades inerciais do sistema. As propriedades inerciais também podem se alterar durante a execução, por exemplo, devido à utilização de combustível. Portanto, o objectivo desta tese é abordar a redução da incerteza paramétrica para uma execução segura e precisa de operações em órbita.

As contribuições incluem dois métodos de Identificação de Sistemas (SysID) - um que utiliza o equilíbrio energético para filtragem de dados, e um segundo método que envolve a extensão da mecânica discreta de variações para uma estimação exata, utilizando apenas as poses. Este método demonstrou ser pelo menos uma ordem mais preciso do que o uso de discretização por diferenças finitas. No entanto, a execução dedicada de sysID é demorada, pois o robô tem de obter medições informativas para estimação, explorando adequadamente o seu espaço de estados. Para além disso, pode ser inviável interromper a tarefa primária do sistema para realizar sysID.

Portanto, esta tese propõe uma estrutura de planeamento do movimento chamado RATTLE, que combina simultaneamente os objectivos do movimento para a redução da incerteza (exploração) e o cumprimento dos objectivos de controlo capitalizando sobre as estimativas mais recentes. Ao completar uma tarefa como o transporte de objectos, RATTLE melhora a precisão das estimativas, encorajando o ganho de informação e utilizando estimativas recentes para actualizar os módulos de planeamento e controlo online. A RATTLE foi testada para um cenário de transporte de carga nos robôs free-flyers da NASA a bordo da Estação Espacial Internacional, demonstrando as suas capacidades robustas de procura de informação em tempo real e de redução da incerteza paramétrica.

Palavras-chave: Incerteza paramétrica, robótica espacial, otimização de trajectórias, estimação de parâmetros inerciais, planeamento de movimentos

Abstract

Given the benefits of automation in space exploration, robotic free-flyers have a pivotal role in future space missions. Free-flying robots will soon be performing on-orbit servicing tasks such as debris removal, assembly, repair, and refueling, among others. Moreover, accurate knowledge of the dynamics is essential for planning optimal trajectories and safe task execution using model-based strategies. While the above tasks demand precision and safety, their execution often involves interaction with payload or other spacecraft; consider situations like transportation of components during assembly or docking with a satellite for a deorbiting maneuver. Uncertainty about the robot-object system, in particular, uncertainty regarding the inertial properties of the combined system, can be amplified as a result of this interaction. Therefore, the main goal of this thesis is to address the resulting parametric uncertainty to achieve safe and precise execution of on-orbit operations.

Performing a dedicated system identification (sysID) procedure to determine the inertial characteristics is fuel- and time-consuming, as the robot has to obtain informative measurements by adequately exploring its state-space. Moreover, it might not always be feasible to interrupt the system's primary task to carry out a sysID procedure. To that end, this thesis investigates a class of approaches called here as *simultaneous learning*, which attempt to combine the contrasting objectives of motion for uncertainty reduction (exploration) and the fulfillment of control objectives capitalizing on the most recent estimates (exploitation). The intent is to avoid spending time and resources on sysID if sufficient uncertainty reduction can be obtained on the fly while completing other tasks.

Two approaches to online parameter estimation were explored in building up to simultaneous learning. Contributions include using a novel energy-balance approach for measurement data filtering and the extension of discrete variational mechanics for accurate, complete rigid-body online inertial property estimation using only pose estimates. This method was shown to be at least one order of magnitude more accurate than the use of finite difference discretization.

The crux of this thesis is an uncertainty-aware motion planning framework with parameter learning, dubbed RATTLE, that weights adjustable information gain to promote online parameter estimation. En route to performing a primary task, e.g., payload transportation, RATTLE improves parameter estimation accuracy through its information incentivization approach while leveraging the amassed model knowledge to update the planning and control modules in real time. The resulting benefits are two-fold: perform excitation maneuvers safely without primary task interruption while saving fuel and time, and use the learned parameters to improve the safety and precision of the primary task execution in uncertain environments. RATTLE was tested for a microgravity cargo transportation scenario on NASA's Astrobee free-flyers aboard the International Space Station, demonstrating RATTLE's real-time robust information-seeking, motion planning, and parametric uncertainty reduction capabilities in a challenging and uncertain microgravity environment.

Keywords: Parametric uncertainty, space robotics, trajectory optimization, inertial parameter estimation, motion planning

Acknowledgments

I consider myself incredibly fortunate to have been granted the chance to combine robotics and space, both keen interests of mine, in this thesis. The five years spent in the vicinity of the Atlantic coast, in sunny Lisbon, were filled with academic and personal development thanks to the support of many individuals.

First, I am deeply grateful to my advisor, Prof. Rodrigo Ventura, for trusting me with this opportunity, guiding me, and granting me the autonomy to find my niche. I would also like to thank Prof. David Miller for the opportunity to conduct a research visit to MIT, which fostered a long-standing collaboration on uncertainty-aware motion planning. Additionally, thanks to collaborators on the ReSWARM project, Keenan Albee, Bryce Doerr, and Pedro Roque, for the research discussions (and for keeping the Slack messages flowing) and to Prof. Richard Linares. The opportunity to test our work in microgravity on the Astrobee free-flyer platform was the highlight of this thesis, and our visits to NASA Ames to perform ground tests on Astrobee prepared us well for microgravity. For this, I would like to thank the team at Ames - Brian Coltin, Marina Moreira, Ruben Garcia Ruiz, Jose Benavides, Trey Smith, and others; without their support, the ISS test sessions would never have materialized.

At Instituto Superior Técnico, I would also like to thank Prof. Pedro Lima and Prof. Isabel Ribeiro for their guidance and feedback during the INFANTE project and IRSg seminars. Further, I am thankful to Jéssica Corujeira and Rute Luz, my lab (and performing arts) colleagues, who became my second family. Thanks to friends and colleagues, including Mithun Kinarullathil, Kağan Erünsal, Prasad Antarvedi, Oscar Lima, Carlos Azevedo, Rui Bettencourt, Meysam Basiri, Rodrigo Serra, Ignacio Cabellero, Francisco Del Rio and others who made this journey one to cherish.

I would like to thank my parents for their unwavering support from a thousand miles away, as none of this would have been possible without them. And finally, my deepest thanks to Ashish for being a sounding board for my ideas and accompanying me through it all; I look forward to starting our life together.

This thesis was completed thanks to support from the following sponsors: Portuguese Science Foundation (FCT) grant PD/BD/150632/2020, the LARSyS - FCT Plurianual funding 2020-2023 and the P2020 INFANTE project 10/SI/2016. Further, research visits to MIT and to NASA Ames were made possible thanks to a seed project under the MIT Portugal Program.

Contents

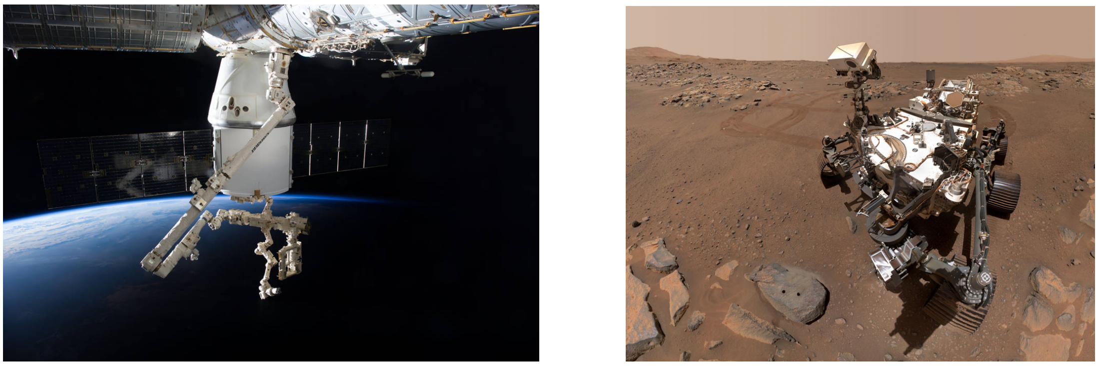
List of Tables

List of Figures

Chapter 1

Introduction

The ability to explore extreme environments and extend human capabilities in space has made robotics an indispensable part of space exploration [1–3]. Robots operating on the surface of planets, like the Perseverance rover, facilitate the examination of unknown environments by gathering samples of the Martian terrain [4] (fig. 1.1(a)). Those operating on-orbit, like NASA's International Space Station (ISS)'s Special Purpose Dextrous Manipulator system (SPDM, fig. 1.1(b)), which routinely performs maintenance on the exterior of the ISS [5], save crew time by reducing the need for spacewalks.



(a) (b)

Figure 1.1: The SSRMS preparing to remove external cargo from the Dragon resupply ship [NASA, 2014] (L), and (R), the Perseverance rover on Mars looking at the rock that it drilled samples from [NASA/JPL-Caltech, 2021]. The SSRMS is teleoperated from inside the ISS, while Perserverance is semi-autonomous with autonomous sample collection and navigation capabilities.

While astronauts have successfully retrieved and repaired satellites in the past, notably during the Hubble Space Telescope repair missions (fig. 1.2), astronaut servicing is risk-prone and cost-inefficient considering the number of space assets. Robotic on-orbit servicing (OOS) is a promising alternative, offering a wide range of capabilities such as on-orbit inspection, maintenance, assembly, and printing [6, 7]. Though the case for mission flexibility and satellite lifetime extension afforded by robotic OOS was made more than two decades ago [8, 9], autonomous OOS is still an emerging technology in its demonstration phase. Concern about mounting space debris, coupled with the possibility of life extension of satellites in the Geostationary Earth Orbit (GEO) and inspection and deorbiting of those in the

Low Earth Orbit (LEO), has created a maturing market for autonomous OOS in recent times. As a result, the space industry is making strides to advance the Technological Readiness Levels (TRLs) of OOS, often in collaboration with government space agencies. Some examples of demonstration missions planned for this decade are NASA's On-Orbit Servicing, Assembly, and Manufacturing (OSAM)-2[1] in partnership with Redwire space, the Clearspace-1 mission in collaboration with ESA and Clearspace SA[2], and Astroscale's End-of-Life Service by Astroscale (ELSA-M)[3].

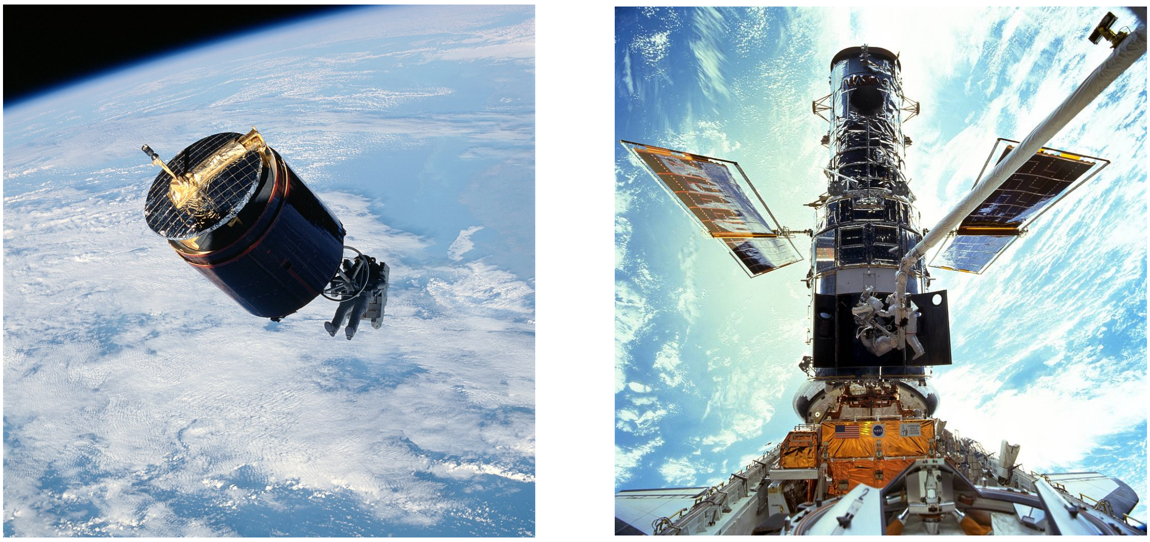


Figure 1.2: Astronauts performing on-orbit servicing. NASA Astronaut Dale Gardner using the free-flying Manned Maneuvering Unit to approach the tumbling Westar 6 satellite and retrieve it during the STS-51-A mission [NASA, 1984] (L), and (R), NASA Astronauts Steven Smith and John Grunsfeld are seen replacing the Hubble Space Telescope's gyroscopes with the help of Space Shuttle Discovery's Remote Manipulator System as an anchor. [NASA, 1999]

1.1 Assistive free-flyers

Free-flying spacecraft performing autonomous OOS will generally be equipped with robotic arms for target manipulation. Apart from the manipulator systems present on the exterior of the ISS, an assortment of free-flying robots have been deployed in space. Termed Assistive free-flyers (AFFs), these robots serve as test and demonstration platforms for on-orbit autonomy algorithms. In the long term, AFFs would work alongside crew, performing housekeeping, transportation, survey, and documentation [10].

The Autonomous Extravehicular Activity Robotic Camera Sprint (AERCam Sprint) was the first prototype of a free-flying robot as an inspection tool in space. It flew on the STS-87 mission in 1997 (Fig. 1.3), controlled by astronauts from the Space Shuttle [11]. AERCam's demonstration flight lasted 30 minutes. Fig. 1.4 shows some of the recent AFFs deployed on board the ISS. The Synchronized Position Hold Engage and Reorient Experimental Satellite (SPHERES) [12] were three satellites operational on the ISS from 2006 to 2019, used to test a range of navigation, control, and formation flying algorithms. Continuing the legacy of SPHERES, Astrobee is NASA's current free-flying test facility on the

---

[1] `https://www.nasa.gov/mission_pages/tdm/osam-2.html`

[2] `https://www.esa.int/Space_Safety/ClearSpace-1`

[3] `https://astroscale.com/elsa-m/`

Figure 1.3: The Autonomous Extravehicular Activity Robotic Camera Sprint (AERCam Sprint), a prototype free-flying television camera, as seen from the cargo bay of the Space Shuttle Colombia during mission STS-87. AERCam was teleoperated from the shuttle during the experiment. [Credit: NASA, 1997]

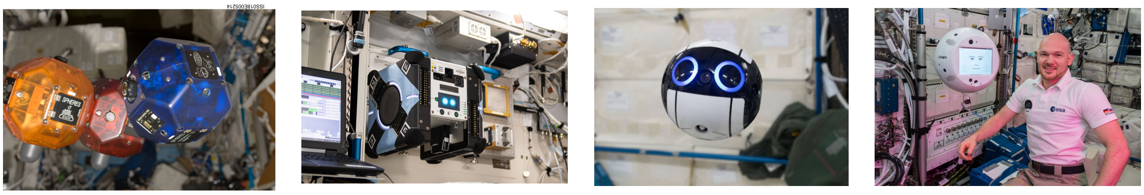


Figure 1.4: Some of the AFFs deployed on board the ISS. From left: SPHERES, CIMON, Int-ball and Astrobee [Credit: NASA, DLR and JAXA]

ISS. In addition to guidance, control, and navigation algorithms, the Astrobee platform is being used to develop caretaking capabilities for future spacecraft and stations such as the Gateway [13]. SPHERES and Astrobee are both described in further detail in Appendix B.

The Japanese Exploration Module (JEM) Internal Ball Camera or Int-ball, launched to the ISS in 2017, autonomously moves to a desired position in the JEM to capture photos and videos. Int-Ball's mission is to take over the task of photographing, which the crew spends about 10% of their task in. Apart from monitoring, future function enhancements expect it to manage inventory and provide EVA assistance [14]. DLR's Crew Interactive MObile companioN or CIMON, developed by Airbus and IBM, is a technology demonstrator flight assistance system housed in the ISS' Columbus module. Running IBM Cloud's Watson AI technology, CIMON is a voice controlled Artificial intelligence, designed to reduced crew stress via documentation, video capture, interaction, and relaying information, [4].

1.2 Motivation

To safely and precisely carry out tasks like assembly, de-orbiting, and repair, free-flyers must master manipulation and interaction with payload and other spacecraft. Further, The Global Exploration Roadmap

---

[4] `https://www.dlr.de/content/en/articles/news/2020/02/20200415_cimon-2-makes-its-debut-on-the-iss.html`

of the International Space Exploration Coordination Group (ISECG) deems NASA's Gateway[5] vital to deep space exploration, specifically to carry out operations and science outside the protection of the Van Allen belts[6]. Set to be humanity's first station in lunar orbit and uncrewed for extended periods, autonomous caretaking capabilities are crucial to the Gateway project. NASA's Integrated System for Autonomous and Adaptive Caretaking (ISAAC) project addresses this by developing monitoring, inspection, and logistics capabilities using robots like Astrobee and Robonaut on the ISS [13]. In these cases, the robots would routinely transport payloads like unpacked cargo or components for assembly and repair. Therefore, the free-flyer cargo transporation scenario and the resulting parametric uncertainty is central to this thesis.

Uncertainty, which manifests in various forms, poses a risk to safe and precise task execution by making it difficult to predict the robot's behavior. Disturbances like atmospheric drag, solar radiation pressure, and gravity gradient torques are examples of external causes of uncertainty. Additionally, unmodeled dynamic interactions or a lack of knowledge of the robot's parameters can cause uncertainty. This thesis focuses on the latter, which we call *parametric uncertainty*, primarily considering on-orbit manipulation operations. Note that this uncertainty is called epistemic in machine learning parlance, meaning that it is reducible; caused by a lack of knowledge about the system model and not due to an underlying random phenomenon (also known as aleatoric uncertainty). [15].

Inadequate knowledge of the system's inertial parameters, e.g., mass and moments of inertia is a significant source of parametric uncertainty. Even if characterized adequately, some of these parameters might change during operation, for instance, due to fuel consumption or grasping payload, thus casting doubt on their previously known values. Unlike their terrestrial counterparts, space manipulator systems also have to deal with dynamic coupling, which is a reaction of the robot base to movements in the manipulator [16]. Furthermore, large, light, and flexible structures like solar panels, antennas, and manipulators are abundant in spacecraft, so control systems must consider their flexibilities [17]. Finally, onboard fuel tanks and the associated sloshing also impact the inertial properties of the robot base and its behavior [18].

A quick consideration of the benefits of reducing parametric uncertainty motivates the need for methods to deal with them. Advanced model-based control strategies [19, 20] are usually employed for obtaining precision and safety in on-orbit servicing tasks. These methods consider the effects of the robot dynamics and thus rely on their accurate knowledge. Moreover, the kinematics of space robot manipulators depend on their dynamic parameters [21]. Uncertainty in the system model can impede the fulfillment of control objectives and even cause instability, thus posing a risk to operational safety. From the standpoint of path planning and tracking, knowledge of the system model to a certain degree of accuracy allows optimization for fuel efficiency, time, and energy consumption, among others. Besides, the inability to predict the system's behavior can also hamper state estimation and fault detection [22, 23]. Further, de-orbiting and post-capture stabilization operations could benefit from knowing the system's inertial properties, for instance, to direct thrust forces through the center of mass of the combined system [22, 24].

---

[5] `https://www.nasa.gov/gateway/overview`

[6] `https://www.globalspaceexploration.org/wp-content/isecg/GER_Supplement_Update_2022.pdf`

Methods for dealing with dynamic uncertainty in the state of the art can be broadly divided into three classes:

1. estimating the parameters to a certain level of accuracy [22–26],
2. employing control strategies that function in spite of uncertainty without explicit model learning [27–34], and
3. improving knowledge of the system model and disturbances while simultaneously attempting to control, which we term simultaneous learning [35–41].

These methods are further described in chapter 3.

1.3 Problem Statement

A key aspect of free-flyer on-orbit servicing missions is interaction with cargo and other satellites of potentially unknown dynamic properties. Whether to learn uncertain parameters or focus solely on task completion depends on the mission. After consideration of the relevance of parameter learning to advanced model-based control and trajectory optimization, this thesis takes the path of uncertainty reduction by learning. Assuming a rigid-body manipulation scenario where the robot has already grasped a cargo object leading to an increase in parametric uncertainty, the global question that this thesis aims to address is **how to get a sufficient amount of online parametric uncertainty reduction for a robotic free-flyer rigid-body manipulation scenario?**

When it comes to parameter learning, many research works such as [22, 25, 26, 42–44], perform uncertainty reduction through offline parameter estimation (this will be detailed further in chapter 3). Though this problem has been widely-researched by proposing to keep the spacecraft in free-floating mode [45–47], estimation accuracy in the face of noisy measurements (primarily acceleration) is a significant issue when it comes to system identification (sysID) in free-flying mode. Therefore, **the first objective of this thesis is to investigate methods for accurate free-flyer system identification**.

As far as simultaneous learning is concerned, this is a line of vivid robotics research, particularly learning-based control and active learning approaches [48–50]. Strategies from indirect adaptive control have also been proposed to solve this problem [51–53]. While multiple works such as [54–56] have applied self-tuning control to spacecraft, many make the free-floating assumption. Thus, simultaneous learning in robotic free-flyers is a relatively unexplored territory. Taking stock of the challenges in simultaneous learning, **the second objective of this thesis is to research uncertainty reduction through *online* simultaneous learning approaches that are not overly conservative, capable of replanning, and scalable for application to free-flyer dynamics**.

Due to the restricted accessibility of microgravity platforms, most spacecraft system identification methods are tested with simulated data or on nearly frictionless flat tables with robot hardware placed on air bearings to simulate microgravity (fig. 1.5), allowing translation in X and Y directions and rotation about the Z axis. Though the latter is an excellent way to test algorithms on hardware on the ground, it

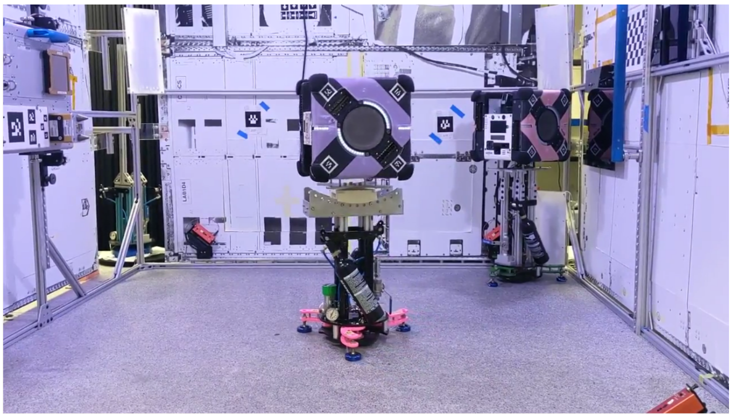


Figure 1.5: The Astrobee granite table facility at NASA Ames Research Center. Freeflyer Bsharp is placed on air-bearing to simulate microgravity in the XY plane.

limits the system's degrees of freedom. Additionally, the air carriage changes the inertial properties and, consequently, the system's behavior. NASA's Astrobee free-flyer platform (Appendix 5.1) - whose future responsibilities could include autonomous maintenance of space stations such as the ISS, for instance, via cargo transportation - can now be accessed by guest science research groups. Therefore, **the third objective of this thesis is to perform a hardware micro-gravity validation of the developed simultaneous learning approach, demonstrating its properties in real-time on resource-constrained hardware**.

Note that even though algorithms developed within the scope of this thesis are focused on rigid-body free-flyers, they make no limiting assumptions and are adaptable to other kinds of mobile robotic systems.

1.4 Contributions

This thesis has the following contributions:

1. An offline inertial parameter estimation approach using Fourier Series excitation trajectories and a novel energy-balance-based method for filtering noisy measurement data. The discrete Newton-Euler equations of motion are commonly used to formulate an estimation problem requiring measurement data such as angular accelerations, which are not readily available and often calculated using finite difference discretization, which amplifies the noise. This method aimed to improve estimation accuracy via noise filtering, thus addressing the first objective of this thesis [57–59]

2. A recursive estimation approach for generating physically feasible full-body inertial estimates formulated using the discrete variational equations of motion. In addition to requiring noisy acceleration data, the discretized Newton-Euler equations often do not accurately represent the evolution of dynamical systems. In contrast, the proposed formulation respects energy dissipation, does not need acceleration measurements and addresses the first objective of the thesis by yielding an estimate accuracy in simulation which is at least one order of magnitude larger than finite difference discretization for acceleration computation [60]

3. A real-time, information-aware framework for online parameter learning and control called RATTLE. Capable of adaptively balancing between information-seeking and goal-tracking objectives, RATTLE is designed to fulfill its primary task while gathering information about the system's parameters and incorporating this knowledge in subsequent plans and control actions en route. THe RATTLE method, therefore, addresses the second thesis objective [61, 62]

4. RATTLE was demonstrated in microgravity on the Astrobee free-flyer on board the ISS, making it one of the first autonomy algorithms to be tested on Astrobee and the first information-aware motion planning algorithm to be tested in space to the author's knowledge. The transition to hardware addressed the third objective of this thesis, resulting in reporting of hardware implementation considerations and open source contributions, both significant byproducts of the thesis [63, 64]

The following publications were a result of the work produced in this thesis:

- **M. Ekal** and R. Ventura, "On inertial parameter estimation of a free-flying robot grasping an unknown object". Published in proceedings of the 5th International Conference on Control, Decision and Information Technologies (CoDIT), 2018.
- **M. Ekal** and R. Ventura, "An energy balance based method for parameter identification of a free-flying robot grasping an unknown object". Published in proceedings of the 2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC).
- **M. Ekal** and R. Ventura, "On the accuracy of inertial parameter estimation of a free-flying robot while grasping an object". Published in the Journal of Intelligent & Robotic Systems, 2020. Impact factor 3.611, Q1.
- K. Albee, **M. Ekal**, R. Ventura, and R. Linares, "Combining parameter identification and trajectory optimization: Real-time planning for information gain". Published in proceedings of the Advanced Space Technologies in Robotics and Automation Symposium (ASTRA), 2019.
- **M. Ekal** and R. Ventura, "A dual quaternion-based discrete variational approach for accurate and online inertial parameter estimation in free-flying Robots". Published in proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), CORE rank B.
- **M. Ekal**, K. Albee, B. Coltin, R. Ventura, R. Linares, and D. W. Miller, "Online Information-Aware Motion Planning with Inertial Parameter Learning for Robotic Free-Flyers". Published in proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), CORE rank A.
- K. Albee, **M. Ekal**, B. Coltin, R. Ventura, R. Linares, and D. W. Miller, "The RATTLE Motion Planning Algorithm for Robust Online Parametric Model Improvement with On-Orbit Validation". Published in Robotics and Automation Letters, 2022. Scopus impact factor: 5.427, Q1.
- B. Doerr, K. Albee, **M. Ekal**, R. Ventura, and R. Linares, "The ReSWARM Microgravity Flight Experiments: Planning, Control, and Model Estimation for On-Orbit Close Proximity Operations". Soon to be submitted to Field Robotics.

- Instead of learning the parameters by explicitly assuming a dynamic model as discussed in sec.1.2, learning the system dynamics is an open-ended approach to dynamic uncertainty reduction. A work that uses symplectic momentum neural networks with priors based on discrete variational mechanics for end-to-end model learning of non-separable mechanical systems was proposed in S. Santos, **M. Ekal**, and R. Ventura, “Symplectic momentum neural networks-using discrete variational mechanics as a prior in deep learning”. Published in Learning for Dynamics and Control Conference (pp. 584-595), PMLR, 2020.

1.5 Thesis organization

Subsequent chapters of this thesis are organized as follows. Chapter 3 provides a review of existing literature on dealing with dynamic uncertainty. This thesis divides the state of the art into three categories: (1) Uncertainty reduction through system identification, (2) Control in the presence of uncertainty via disturbance rejection, and (3) Simultaneous learning and control. The thesis contributions are detailed in chapters 4, 5 and 6. They include two rigid body parameter identification approaches that tackle the problem of noisy acceleration measurements. The first method looks at this problem from a noise-filtering perspective, while the second formulates the estimation problem using discrete variational equations of motion, which are only a function of the pose. An information-incentivizing (re)planning framework coupled with parameter learning, called RATTLE, was also developed to address the problem of reducing parametric uncertainty online while simultaneously fulfilling the control objectives. Conclusions to this thesis are presented in chapter 7.

Chapter 2

Background

This chapter presents a summary of concepts relevant to this thesis. Starting with rigid body pose representation, a review of rigid body and free-flyer dynamics is presented. Further, central concepts in estimation, optimal trajectory design for robot identification, as well as frameworks used in this thesis, such as model predictive control (MPC) and semidefinite program (SDP), are covered. This is followed by an introduction to discrete variational mechanics and Lie Group variational integrators. The chapter concludes with a review of the key properties of quaternions and dual quaternions.

2.1 Rigid-body dynamics

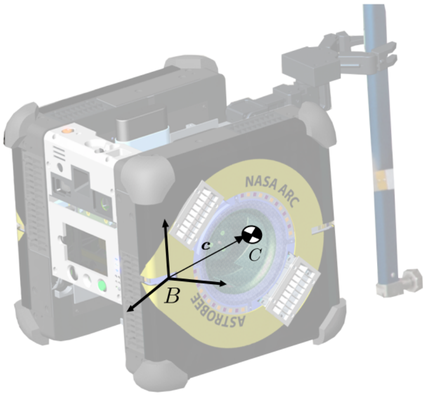


Figure 2.1: A free-flyer grasping an object. Offset $\boldsymbol{c} \in \mathbb{R}^3$ separates the center of mass, C of the combined system, and the origin of the body-fixed frame, B. [Astrobee image credits: NASA]

Consider that the origin of the body-fixed frame, B, lies at the center of mass of the robot. The Newton-Euler equations of motion of a rigid body with respect to B are [65]

$$\boldsymbol{\tau} = J\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times J\boldsymbol{\omega} \tag{2.1}$$

$$\boldsymbol{f} = m\dot{\boldsymbol{v}} \tag{2.2}$$

where (2.1) gives the relation between the torque (or moment) $\boldsymbol{\tau}$ applied to the body at its center of mass and its angular acceleration $\dot{\boldsymbol{\omega}}$. The angular velocity is denoted by $\boldsymbol{\omega}$ and the rotational inertia of the body about its center of mass is denoted by J,

$$J = \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{bmatrix} \tag{2.3}$$

In order to describe a physical system, $J \succ 0$,where $\succ 0$ denotes that the matrix is positive definite[1], and satisfies the triangle inequality constraints, i.e., $J_{11} + J_{22} \geq J_{33}$, $J_{22} + J_{33} \geq J_{11}$ and $J_{11} + J_{33} \geq J_{22}$. Note that since the inertia tensor J represents the distribution of mass relative to a reference point (usually the center of mass) on the body, (2.1) is expressed in the body frame so that J remains a constant.

Equation (2.2) gives the relation between the applied force $\boldsymbol{f}$ and the linear acceleration $\dot{\boldsymbol{v}}$ of the body's center of mass. The mass of the body is denoted by m, $m > 0$, and the linear velocity of its center of mass by $\boldsymbol{v}$.

If the origin of the body-fixed coordinate frame is not coincident with the system's center of mass (fig.2.1), the equations of motion with respect to B become [65]

$$\begin{bmatrix} \boldsymbol{\tau} \\ \boldsymbol{f} \end{bmatrix} = \begin{bmatrix} J_C + mS(\mathbf{c})S^T(\mathbf{c}) & mS(\mathbf{c}) \\ mS^T(\mathbf{c}) & m\mathbf{I}_{3\times 3} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{\omega}} \\ \dot{\boldsymbol{v}} \end{bmatrix} + \begin{bmatrix} S(\boldsymbol{\omega})\left(J_C + mS(\mathbf{c})S^T(\mathbf{c})\right)\boldsymbol{\omega} \\ mS(\boldsymbol{\omega})S(\boldsymbol{\omega})\mathbf{c} \end{bmatrix} \tag{2.4}$$

where J_C is the rotational inertia about the center of mass and $S(c)$ denotes the skew symmetric matrix

$$S(\mathbf{c}) = \begin{bmatrix} 0 & -c_3 & c_2 \\ c_3 & 0 & -c_1 \\ -c_2 & c_1 & 0 \end{bmatrix} \tag{2.5}$$

Using the parallel axis theorem, J_B, the rotational inertia about B is written as

$$J_B = J_C + mS(\boldsymbol{c})S^T(\boldsymbol{c}) \tag{2.6}$$

Therefore, (2.4) leads to the canonical form of rigid-body dynamics [65]

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \boldsymbol{\mathcal{F}} \tag{2.7}$$

where $\mathbf{q}$ signifies the generalized coordinates, $\mathbf{C}$ is the vector of Coriolis and centrifugal forces, $\mathcal{F}$ the vector of applied forces, and $\mathbf{M}$ is referred to as the mass matrix.

Comparing with (2.4), $\dot{\mathbf{q}} = \left[\dot{\boldsymbol{\omega}}^T \ \ \dot{\boldsymbol{v}}^T\right]^T$, $\boldsymbol{\mathcal{F}} = \left[\boldsymbol{\tau}^T \ \ \boldsymbol{f}^T\right]^T$, and $\mathbf{M}$ is the spatial inertia matrix,

$$\mathbf{M} = \begin{bmatrix} J_B & mS(\boldsymbol{c}) \\ mS^T(\boldsymbol{c}) & m\mathbf{I}_{3\times 3} \end{bmatrix} \tag{2.8}$$

---

[1] Positive definite matrices are symmetric matrices with positive Eigen values. Other key properties of positive definite matrices are discussed in [66]

where $\mathbf{I}_{3\times3}$ is the 3×3 identity matrix.

2.1.1 Free-flyer kinematics and dynamics

A space manipulator system is composed of deployable manipulator arms mounted on an a spacecraft base (fig 2.2). Akin to ground manipulator robots, mobile manipulators consist of joints and links. Links are generally considered to be rigid bodies, whereas joints are connections between two links that constrain their relative motion [67]. The links and joints form a serial chain, i.e., they begin at the robot's base and end at an end-effector.

2.1.2 Free-flying vs. free-floating mode of operation

Two modes of operation are usually considered for space manipulator systems. In free-flying mode the spacecraft base is also actuated in addition to the manipulator joints. The base is fully controlled in position and attitude. In free-floating mode, no external forces and torques act on the base of the spacecraft ($\mathcal{F}_e = 0$ in (2.10)). The base is free to move in reaction to the motion of the manipulator. Therefore, fuel is not used to control the spacecraft in this mode, which helps in extending the spacecraft's life. This mode is also used while manipulating safety-critical payload, in order to avoid jolting movements of the end-effector due to sudden activation of actuation [68]. Free-floating mode is usually employed in space when disturbances on the manipulator system can be ignored, and the spacecraft will not drift from its course. However, actuation is needed for course-correction and to correct for any accumulated angular momentum.

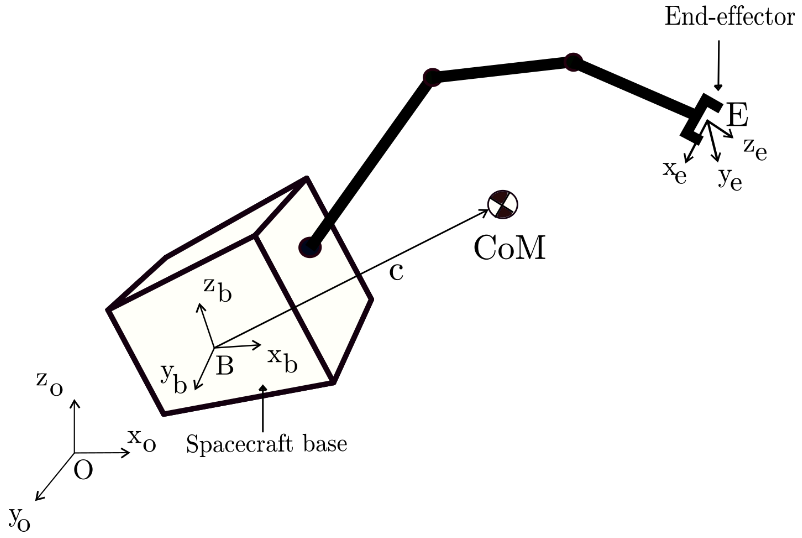


Figure 2.2: A free-floating space robot

2.1.3 Kinematic and Dynamic Equations

Kinematics performs the mapping of trajectories from operational space (end-effector position) to trajectories in joint space using the kinematic Jacobian matrix [65]. For the free-flying robotic system, the robot-base Jacobian $\mathbf{J}_b$, and manipulator Jacobian $\mathbf{J}_m$, can be used to relate the end-effector velocities $\dot{\mathcal{X}}_e = \begin{bmatrix}\boldsymbol{\omega}_e^T & \boldsymbol{v}_e^T\end{bmatrix}^T \in \mathbb{R}^6$ to the joint and base velocites, $\dot{\phi}$ and $\dot{\mathcal{X}}_b$, as [69]:

$$\dot{\mathcal{X}}_e = \mathbf{J}_m \dot{\boldsymbol{\phi}} + \mathbf{J}_b \dot{\mathcal{X}}_b \tag{2.9}$$

where $\mathcal{X}$ denotes the robot pose, and the velocities of the robot base, $\dot{\mathcal{X}}_b = \begin{bmatrix}\boldsymbol{\omega}_b^T & \boldsymbol{v}_b^T\end{bmatrix}^T$. The manipulator joint velocities $\dot{\phi}$ are chosen as generalized coordinates. The first term is related to the motion of the manipulator with respect to the base, while the second component is related to the motion of the base. The equations of motion for free-flying space manipulator systems with n joints, illustrated in fig.(2.2), can be expressed as [69, 70]:

$$\begin{bmatrix} \mathbf{M}_b & \mathbf{M}_{bm} \\ \mathbf{M}_{bm}^T & \mathbf{M}_m \end{bmatrix} \begin{bmatrix} \ddot{\mathcal{X}}_b \\ \ddot{\boldsymbol{\phi}} \end{bmatrix} + \begin{bmatrix} \mathbf{C}_b \\ \mathbf{C}_m \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mathcal{F}}_b \\ \boldsymbol{\mathcal{T}} \end{bmatrix} + \begin{bmatrix} \mathbf{J}_b^T \\ \mathbf{J}_m^T \end{bmatrix} \boldsymbol{\mathcal{F}}_e \tag{2.10}$$

where,

$\mathbf{M}_b \in \mathbb{R}^{6\times 6}$: robot base inertia matrix

$\mathbf{M}_m \in \mathbb{R}^{n\times n}$: manipulator inertia matrix

$\mathbf{M}_{bm} \in \mathbb{R}^{6\times n}$: coupling inertia matrix

$\mathbf{C}_b \in \mathbb{R}^{6}$: velocity dependent Coriolis and Centrifugal forces on the system articulated body

$\mathbf{C}_m \in \mathbb{R}^{n}$: velocity dependent Coriolis and Centrifugal forces on the fixed-base manipulator

$\boldsymbol{\mathcal{T}} \in \mathbb{R}^{n}$: Joint torques

$\boldsymbol{\mathcal{F}}_e, \boldsymbol{\mathcal{F}}_b \in \mathbb{R}^{6}$: Forces on the base and end-effector, respectively

$\mathbf{J}_b \in \mathbb{R}^{6\times 6}$: Jacobian of the robot base

$\mathbf{J}_m \in \mathbb{R}^{6\times n}$: Fixed-base manipulator Jacobian

$\ddot{\mathcal{X}}_b \in \mathbb{R}^{6}$: Acceleration of the robot base

$\mathbf{M} = \begin{bmatrix} \mathbf{M}_b & \mathbf{M}_{bm} \\ \mathbf{M}_{bm}^T & \mathbf{M}_m \end{bmatrix}$: Generalized inertia matrix

The coupling between the base and manipulator motions is evident, as both the upper and lower parts of (2.10), related to the base and the manipulator motion respectively, contain the coupling terms.

2.2 Maximum Likelihood Estimation and the Cramér-Rao Lower Bound

Consider an estimation problem of the form

$$\tilde{\mathbf{y}} = \mathbf{H}\boldsymbol{\theta} + \mathbf{v} \tag{2.11}$$

where,

$\tilde{\mathbf{y}} \in \mathbb{R}^m$: measurement vector

$\boldsymbol{\theta} \in \mathbb{R}^n$: vector of the parameters to be estimated

$\mathbf{v} \in \mathbb{R}^m$: measurement noise

$\mathbf{H} \in \mathbb{R}^{m\times n}$: regressor that maps the parameters to be estimated to the observations

The Maximum Likelihood approach results in estimates that maximize the probability of obtaining the observed set of independent and identically distributed data [71]. Considering that the distribution of $\mathbf{y}$ is conditionally independent of other measurements given the value of parameters $\boldsymbol{\theta}$, the likelihood is a probability density function (PDF) given by the product of individual likelihoods $p(\tilde{\mathbf{y}}_i|\boldsymbol{\theta})$:

$$L(\tilde{\mathbf{y}}|\boldsymbol{\theta}) = \prod_{i=1}^{q} p(\tilde{\mathbf{y}}_i|\boldsymbol{\theta}) \tag{2.12}$$

where q is the total number of PDFs. Logarithm of the likelihood function is usually used, as the natural log is a monotonic function which simplifies the exponential terms present in many PDFs. Additionally it converts products to sum. The Maximum Likelihood Estimator solves the following problem:

$$\underset{\boldsymbol{\theta}}{\text{maximize}} \quad \sum_{i=1}^{q} \log\left[p(\tilde{\mathbf{y}}_i|\boldsymbol{\theta})\right] \tag{2.13}$$

For the case of Gaussian measurement errors, $\tilde{y} \sim \mathcal{N}(0, R)$, after the terms independent of $\boldsymbol{\theta}$ are ignored, the log likelihood function is given by

$$\log\left[p(\tilde{\mathbf{y}}_i|\boldsymbol{\theta})\right] = -\frac{1}{2}\left(\tilde{\mathbf{y}} - \mathbf{H}\boldsymbol{\theta}\right)^T R^{-1}\left(\tilde{\mathbf{y}} - \mathbf{H}\boldsymbol{\theta}\right) \tag{2.14}$$

It is evident then, that the solution of this maximum likelihood problem is equal to the standard least squares solution (A.2.1) with weights $W = R^{-1}$. MLE is asymptotically efficient, which means that provided the sample size is large, it achieves the Cramér-Rao lower bound (introduced in the following section), or yields estimates with the smallest covariance possible by any estimator [71].

Cramér-Rao inequality and the Fisher Information Matrix

The Cramér-Rao inequality gives a theoretical lower bound on the expected uncertainty of the estimates. For an unbiased estimate $\hat{\boldsymbol{\theta}}$ of the estimation problem in (2.11), the Cramér-Rao inequality is given by [71]

$$P \equiv \mathbb{E}\left\{\left(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\right)\left(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\right)^T\right\} \succeq F^{-1} \tag{2.15}$$

where $\mathbb{E}\{\}$ denotes the expectation operator, P is the covariance of the estimates, $\boldsymbol{\theta}$ is the true value of the parameters and F is the Fisher Information matrix (FIM), which is a measure of the amount of information provided by the measurements on the unknown parameters

$$F = \mathbb{E}\left\{ \left[\frac{\partial}{\partial \boldsymbol{\theta}} \ln\left[p(\tilde{\mathbf{y}}|\boldsymbol{\theta})\right]\right] \left[\frac{\partial}{\partial \boldsymbol{\theta}} \ln\left[p(\tilde{\mathbf{y}}|\boldsymbol{\theta})\right]\right]^T \right\} \tag{2.16}$$

An alternative way of computing the FIM is by using the Hessian,

$$F = -\mathbb{E}\left\{ \frac{\partial^2}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T} \ln\left[p(\tilde{\mathbf{y}}|\boldsymbol{\theta})\right] \right\} \tag{2.17}$$

Using the log-likelihood of the general least squares problem with Gaussian measurement errors from (2.14) and (2.17), the FIM is given by

$$F = \mathbf{H}^T W \mathbf{H} \tag{2.18}$$

The MLE asymptotically achieves the Cramér-Rao Lower Bound and is thus an asymptotically efficient estimator. The FIM is additive across time steps, and is positive semi-definite. A recursive method of computing the FIM was presented in [72].

2.3 The linear identification problem

In identification of robotic systems, the reduced dynamic model of an n-degree rigid body robot is often expressed in the following form, which is linear in terms of the barycentric parameters[2] [74]:

$$\gamma(\boldsymbol{x}, \dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}})\, \boldsymbol{\pi} = \boldsymbol{\tau} \tag{2.19}$$

where $\gamma \in \mathbb{R}^{n \times r}$ is composed of robot states and their derivatives, while $\boldsymbol{\tau} \in \mathbb{R}^n$ is a vector of actuation inputs and $\boldsymbol{\pi} \in \mathbb{R}^r$ the vector of parameters to be estimated. As sec. 3.1.1 will discuss, methods other than the dynamical equations of motion, like the angular momentum conservation approach, also lead to a formulation linear in the parameters of interest.

Under the assumption that noise on the actuation data is Gaussian, and if the state measurements can be made free of noise for all practical purposes using appropriate filtering methods, the Maximum Likelihood estimate of (2.19) reduces to a weighted least squares estimate given by

$$\hat{\boldsymbol{\pi}} = \left(\boldsymbol{\Gamma}^T \Sigma^{-1} \boldsymbol{\Gamma}\right)^{-1} \boldsymbol{\Gamma}^T \Sigma^{-1} \mathbf{b} \tag{2.20}$$

where Σ^{-1} is the diagonal covariance matrix of the actuation data, the regressor is

$$\boldsymbol{\Gamma} = \begin{bmatrix} \boldsymbol{\gamma}(\boldsymbol{x}(1), \dot{\boldsymbol{x}}(1), \ddot{\boldsymbol{x}}(1)) \\ \vdots \\ \boldsymbol{\gamma}(\boldsymbol{x}(N), \dot{\boldsymbol{x}}(N), \ddot{\boldsymbol{x}}(N)) \end{bmatrix} \tag{2.21}$$

and

---

[2] The robot's inertial parameters, friction co-efficients, and geometric parameters such as link lengths are collectively referred to as the barycentric parameters [73]

$$\mathbf{b} = \begin{bmatrix} \tau(1) \\ \vdots \\ \tau(N) \end{bmatrix} \tag{2.22}$$

It is important to ensure that $\left(\mathbf{\Gamma}^T \Sigma^{-1} \mathbf{\Gamma}\right)$ is full-rank and well-conditioned. While rank-deficiency issues are commonly solvable using excitation trajectories (section 2.3.1), ill-conditioning of the matrix might be caused by improper scaling of the measurements and parameters. Normalization or the use of an appropriate weighting matrix can help to solve this problem. Equally important is to determine the minimal set of parameters to be identified, as redundancy can cause numerical difficulties. An appropriate grouping of the parameters can be used to determine the reduced parameters a priori, or tools such as QR decomposition and Singular Value Decomposition (SVD) of the regressor matrix can provide insight into the identifiability of the parameters once data has been collected [67].

2.3.1 Trajectory design for identification

Excitation trajectories incentivize visiting of states that ensure the full-rank or well-conditioned nature of the regressor matrix. These trajectories can be made continuous by interpolation and imposing initial and final constraints or by parameterization using Basis Functions. Sinusoids and B-splines possess certain desirable properties and are some commonly used basis functions [75, 76].

Usually, constrained nonlinear optimization is used to find excitation trajectories by optimizing an optimality criterion. For instance, [77] showed that estimate accuracy and noise immunity of a parameter identification experiment depend directly on the condition number of the input correlation matrix. The condition number is the ratio between the largest and smallest singular values of the matrix. The condition number is thus a measure of identifiability in the sense that all parameters are equally identifiable if the condition number is one.

If the information content in the measurement dataset is maximized, for instance, by maximizing any norm of the Fisher Information Matrix (FIM), then it would also correspond to minimizing the lower bound on the variance of the estimates (the FIM is addressed in detail in section 2.2). Optimization of the FIM is, therefore, a good candidate for the design of information-rich trajectories. Under the assumption of Gaussian measurement errors, the FIM for the rigid body identification problem is found to be $F = \mathbf{\Gamma}^T \Sigma^{-1} \mathbf{\Gamma}$. Measures of identifiability based on the information matrix, referred to as the alphabet optimalities [67] include A-optimality, which is maximizing the trace of the inverse FIM, $\left(\mathbf{\Gamma}^T \Sigma^{-1} \mathbf{\Gamma}\right)^{-1}$, or E-optimality, where the least eigenvalue of $\mathbf{\Gamma}^T \Sigma^{-1} \mathbf{\Gamma}$ is maximized.

2.4 Trajectory planning, control and optimization frameworks

2.4.1 Trajectory optimization

A generic continuous-time trajectory optimization problem is of the form

$$
\begin{aligned}
\underset{\boldsymbol{x}(t),\boldsymbol{u}(t)}{\text{minimize}} \quad & l_f(t_f,\boldsymbol{x}(t_f)) + \int_{t_0}^{t_f} l(t,\boldsymbol{x}(t),\boldsymbol{u}(t))dt \\
\text{subject to} \quad & \dot{\boldsymbol{x}}(t) = f(t,\boldsymbol{x}(t),\boldsymbol{u}(t)) \quad \forall t \in [t_0, t_f] \\
& \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \\
& \boldsymbol{x}_{min} \le \boldsymbol{x}(t) \le \boldsymbol{x}_{max} \\
& \boldsymbol{u}_{min} \le \boldsymbol{u}(t) \le \boldsymbol{u}_{max}
\end{aligned} \tag{2.23}
$$

where $\boldsymbol{x} \in \mathbb{R}^n$ denotes the system states and $\boldsymbol{u} \in \mathbb{R}^m$ denotes the control inputs. The first term in the cost function is referred to as the terminal cost, or Meyer term, whiles the second term is a running cost or the Lagrangian. The constraint $\dot{\boldsymbol{x}}(t) = f(t,\boldsymbol{x}(t),\boldsymbol{u}(t))$ enforces the system dynamics. Additional constraints, e.g., on the goal state, or on collision avoidance, are often included. Naturally, a numerical solution of this optimization problem requires parametrization over a discrete and finite set of numbers. Under direct transcription, a discretization approach, the states and inputs are decision variables, while the discrete dynamics are converted to explicit contraints. Using direct transcription, the continuous optimization problem in (2.23) can be discretized as

$$
\begin{aligned}
\underset{\boldsymbol{x},\boldsymbol{u}}{\text{minimize}} \quad & l_f(\boldsymbol{x}_N) + \sum_{k=0}^{N-1} l(\boldsymbol{x}_k,\boldsymbol{u}_k) \\
\text{subject to} \quad & \boldsymbol{x}_{k+1} = \mathcal{F}(\boldsymbol{x}_k,\boldsymbol{u}_k) \quad \forall k \in [0, N-1] \\
& \boldsymbol{x}_{k=0} = \boldsymbol{x}_0 \\
& \boldsymbol{u}_k \in \mathcal{U} \quad \forall k \in [0, N-1] \\
& \boldsymbol{x}_k \in \mathcal{X} \quad \forall k \in [0, N-1]
\end{aligned} \tag{2.24}
$$

More details on discretizing a continuous optimization problem can be found in [78–80].

2.4.2 Model Predictive Control

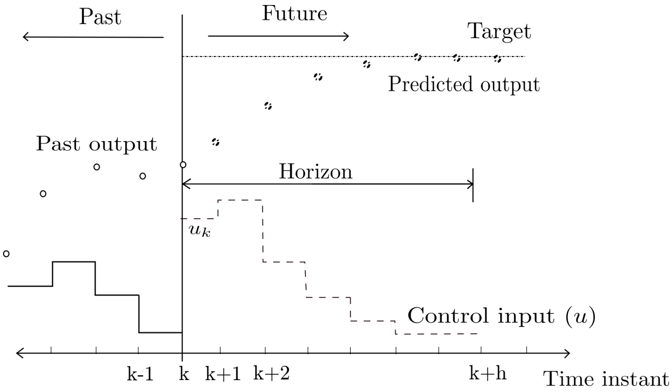


Figure 2.3: Discrete model predictive control scheme

Widely used in the process control industry, Model predictive control (MPC) consists of solving an optimal control problem at each time step k, over horizon h [81]. Out of the series of inputs $\boldsymbol{u}_t, \boldsymbol{u}_{t+1} ... \boldsymbol{u}_{t+h-1}$

calculated after optimization, only the first input $\boldsymbol{u}_t$ is executed. The horizon is then shifted by one time step and the optimal trajectory is recomputed. A key property of Model Predictive Control is the ability to impose state and input constraints directly as a part of the control problem. A discrete MPC problem is of the form

$$\begin{aligned} \underset{\mathbf{x},\mathbf{u}}{\text{minimize}} \quad & \sum_{k=0}^{h-1} \boldsymbol{x}_{t+k}^T Q \boldsymbol{x}_{t+k} + \boldsymbol{u}_{t+k}^T R \boldsymbol{u}_{t+k} \\ \text{subject to} \quad & \boldsymbol{x}_{t+k+1} = A\boldsymbol{x}_k + B\boldsymbol{u}_k, k = 0, ..., h-1 \\ & \boldsymbol{u}_{min} \leq \boldsymbol{u}_{t+k} \leq \boldsymbol{u}_{max}, k = 0, ..., h-1 \\ & \boldsymbol{x}_{t+k} \in \mathcal{X}_{free}, k = 0, ..., h-1 \end{aligned} \tag{2.25}$$

where h is the length of the horizon and $Q \succ 0$ and $R \succ 0$ are positive definite gain matrices. The states are denoted by $\boldsymbol{x}$, the control inputs by $\boldsymbol{u}$, while $\mathcal{X}_{free}$ is the obstacle-free configuration space.

2.4.3 Semidefinite programming

Semidefinite programs (SDP) are convex optimization problems in which a linear function is minimized subject to the constraint that an affine combination of symmetric matrices is positive semidefinite [82]. A general SDP is of the form

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad & c^T x \\ \text{subject to} \quad & A(x) \succeq 0, \ \ A(x) \in \mathbf{S}^{n_A} \end{aligned} \tag{2.26}$$

where $x \in \mathbb{R}^n$, $\mathbf{S}^{n_A}$ is the space of $n_A \times n_A$ symmetric matrices, and $A(x)$ depends affinely on x,

$$A(x) = A_0 + x_1 A_1 + x_2 A_2 + ... + x_k A_k, \ \ A_i \in S^{n_A} \tag{2.27}$$

The inequality $\succeq$ indicates that $A(x)$ is positive semidefinite, i.e., $z^T A(x) z \geq 0$ for all $z \in \mathbb{R}^{n_A}$.

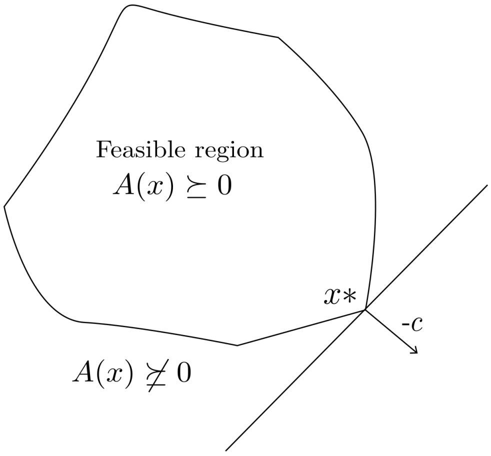


Figure 2.4: Solution of an SDP, $\mathbf{x} \in \mathbb{R}^2$, based on [82]

As SDPs are convex, a variety of fast solvers are available to find the solution (if one exists) efficiently. Several standard problems like linear and quadratic programming can be cast as an SDP. Fig.2.4 illustrates an SDP with $x \in \mathbb{R}^2$ and $A(x)$ representing the constraints. This example has one optimal point $x*$. In simple terms, the solution is obtained by moving as far as possible in the direction of $-c$ while staying in the feasible region.

If enforcing Linear Matrix Inequality (LMI) constraints is of interest, Schur complements can be used to represent a Quadratic Program (QP) as an SDP. For instance, consider the following constrained least squares problem

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad & ||Rx - z||^2 \\ \text{subject to} \quad & M(x) \succeq 0 \end{aligned} \tag{2.28}$$

The cost function $||Rx - z||^2$ is converted into a Linear Matrix Inequality (LMI) constraint, thus converting the quadratic objective into an SDP with the addition of a slack variable s.

$$\begin{aligned} \underset{x \in \mathbb{R}^n,\ s}{\text{minimize}} \quad & s \\ \text{subject to} \quad & \begin{bmatrix} s & (Rx - z)^T \\ (Rx - z) & \boldsymbol{I} \end{bmatrix} \succeq 0 \\ & M(x) \succeq 0 \end{aligned} \tag{2.29}$$

2.5 Periodicity, harmonics, and the Fourier Series

A signal is periodic if there exists a constant $T > 0$ where for any x in the domain of the function f [83]

$$f(x + T) = f(x) \tag{2.30}$$

Consider the following example of simple periodic function,

$$y = A \sin(\omega x + \phi) \tag{2.31}$$

where A denotes the amplitude, ω the frequency, and ϕ represents the phase; A, ω and ϕ are constants. The period of this function is $T = \frac{2\pi}{\omega}$. The function y is also the solution of the differential equation arising from the simplest type of oscillatory motion, or the simple harmonic motion (SHM) [83].

SHM may result from a point mass M, of mass m, moving along a straight line under the influence of a restoring force F whose magnitude is proportional to the distance x of the mass from a fixed point, i.e., $F = -kx$. This leads to the differential equation whose solution is given by (2.31), with $\omega = \sqrt{\frac{k}{m}}$

$$\frac{d^2x}{dt^2} = -\omega x \tag{2.32}$$

Rewriting (2.31) using the sine angle of sum trigonometric identity[3], we obtain

$$A \sin(\omega x + \phi) = a \cos \omega x + b \sin \omega x \tag{2.33}$$

---

[3] $\sin(A + B) = \sin A \cos B + \cos A \sin B$

where $a = A\sin\phi$ and $b = A\cos\phi$. Therefore (2.31) is expressed as a sum of sine and cosine waves of the same period but different magnitudes. Extending this discussion, we consider a signal $f(x)$ with $\omega = 1$ for simplicity

$$S_N = A + \sum_{n=1}^{N} \left(a_n \sin(nx) + b_n \cos(nx)\right) \tag{2.34}$$

where A is a constant. The signal is a sum of sines and cosines with different magnitudes, and frequencies which are positive interger multiples, or harmonics, of the fundamental frequency $n = 1$. The period of the harmonic n is $\frac{T}{n}$. Though S_N is a more complex signal than y, it a sum of periodic functions and therefore itself periodic with frequency equal to the fundamental frequency, ω and period $T = \frac{2\pi}{\omega}$. Series with the structure of S_n are called trigonometric series.

Consider an integrable function $f(x)$ with period 2π ($\omega = 1$), which can be expanded into the following infinite trigonometric series:

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \sin nx + b_n \cos nx) \tag{2.35}$$

Here, the constant A is written as $\frac{a_0}{2}$ for notational simplicity. The coefficients a_0, a_n and b_n can be obtained by integrating $f(x)$ and its product with $\sin(nx)$ and $\cos(nx)$ in the following manner

$$\begin{aligned} a_n &= \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\cos(nx)dx \qquad (n = 0, 1, \cdots) \\ b_n &= \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\sin(nx)dx \qquad (n = 1, 2, \cdots) \end{aligned} \tag{2.36}$$

If all coefficients can be obtained from the single integrable function $f(x)$ in this manner, then a_0, a_n and b_n are called the **Fourier coefficients** of $f(x)$, and the trigonometric series is called its **Fourier series** [83, 84]. An important assumption here is that the trigonometric series converges and sums to $f(x)$.

Note that the complex or exponential form is a representation of the Fourier Series alternative to the trigonometric form. It can be obtained by using Euler's formula which relates trigonometric and exponential functions, $e^{(i\varphi)} = \cos(\varphi) + i\sin(\varphi)$, where e denotes the exponential and $i = \sqrt{-1}$. We can then write the following

$$\cos(nx) = \frac{e^{inx} + e^{-inx}}{2}, \qquad \sin(nx) = \frac{e^{inx} - e^{-inx}}{2i}, \tag{2.37}$$

Substituting (2.37) in (2.35) and rearranging yields the complex form [83]

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{inx} \tag{2.38}$$

with the coefficients c_n given by

$$c_n = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)e^{-inx}dx \qquad (n = 0, \pm 1, \pm 2, \cdots) \tag{2.39}$$

Note the straightforward conversion between the coefficients

$$c_0 = a_0, \quad c_n = (b_n - ia_n)/2, \quad c_{-n} = (b_n + ia_n)/2 \qquad n = 1, 2, 3, \cdots \tag{2.40}$$

2.5.1 The Discrete Fourier Transform

As (2.34) shows, the Fourier Series is a signal in the time-domain expressed as a sum of its frequency components. The Discrete Fourier Transform (DFT) enables the separation of a timeseries, i.e., a series of data samples, in its component frequencies, making it a valuable spectral analysis tool [85]. However, as real data is often noisy and might not be periodic, a general and finite form of (2.37) is used for discrete signals. Note that there exist multiple representations of the discrete Fourier Series and DFT in literature. Here, we use the one from [86, 87].

Let X_k denote the kth sample in a timeseries of length N. For the purpose of the discrete Fourier Series, the signal is considered to have a period of N, with a resulting fundamanetal frequency ω_0 of $\frac{2\pi}{N}$. A_n is the nth coefficient or the complex Fourier amplitude of the frequency n, given by the DFT

$$A_n = \sum_{k=0}^{N-1} X_k e^{-i2\pi nk/N} \qquad n = 0, \cdots, N-1 \tag{2.41}$$

An inverse DFT operation returns the timeseries as:

$$X_k = (1/N) \sum_{n=0}^{N-1} A_n e^{i2\pi kn/N} \qquad k = 0, \cdots, N-1 \tag{2.42}$$

Comparing (2.42) to (2.38) highlights the adaptations made for the discrete and finite case. in (2.42), the timeseries X_k is approximated with N harmonics, yielding the same number of coefficients as the samples in the timeseries. An important property of the DFT is that the DFT returns the least squares fit of the truncated Fourier Series.

2.6 Discrete variational mechanics

The discrete variational equations of motion follow the derivation of the continuous equations of motion, but on a discrete level [88]. Analogous to the states q and $\dot{q}$ of the continuous mechanics, the generalized coordinates in discrete mechanical systems at time step k are q_k and q_{k+1}. The evolution of the states is described using a time-step $h \in \mathbb{R}$. A simplified derivation of the discrete Euler-Lagrange (EL) equations while drawing parallels with the well-known continuous EL equations is presented here. This subject is treated in-depth in [88].

Consider the Lagrangian of a system $L(q, \dot{q})$. In the standard Lagrangian mechanics approach, an action integral is formed by integrating the Lagrangian along a curve $q(t)$. Taking its variational derivative while keeping the endpoints of $q(t)$ fixed (in this case at zero, $\delta q(T) = \delta q(0) = 0$, where δ represents the variation), and using integration by parts leads to

$$\delta \int_0^T L(q(t), \dot{q}(t)) \mathrm{d}t = \int_0^T \left[\frac{\partial L}{\partial q} \cdot \delta q + \frac{\partial L}{\partial \dot{q}} \cdot \delta \dot{q} \right] \mathrm{d}t \tag{2.43}$$

$$= \int_0^T \left[\frac{\partial L}{\partial q} - \frac{\mathrm{d}}{\mathrm{d}t} \left(\frac{\partial L}{\partial \dot{q}} \right) \right] \cdot \delta q \, \mathrm{d}t \tag{2.44}$$

Imposing the condition that the variation in the action should be zero for all values of δq results in the Euler-Lagrange equation,

$$\frac{\partial L(q, \dot{q})}{\partial q} - \frac{\mathrm{d}}{\mathrm{d}t} \left(\frac{\partial L(q, \dot{q})}{\partial \dot{q}} \right) = 0 \tag{2.45}$$

The continuous equations are then discretized, for instance, by using finite differences to approximate the derivatives,

$$\dot{q} \approx \frac{q(k+h) - q(k)}{h} \tag{2.46}$$

Then, Ordinary Differential Equation (ODE) integration methods could be used to solve eq.(2.45) and propagate the states in time. However discretizing the continuous equations in this manner does not preserve properties such as symplecticity[4] or energy preservation of the continuous system.

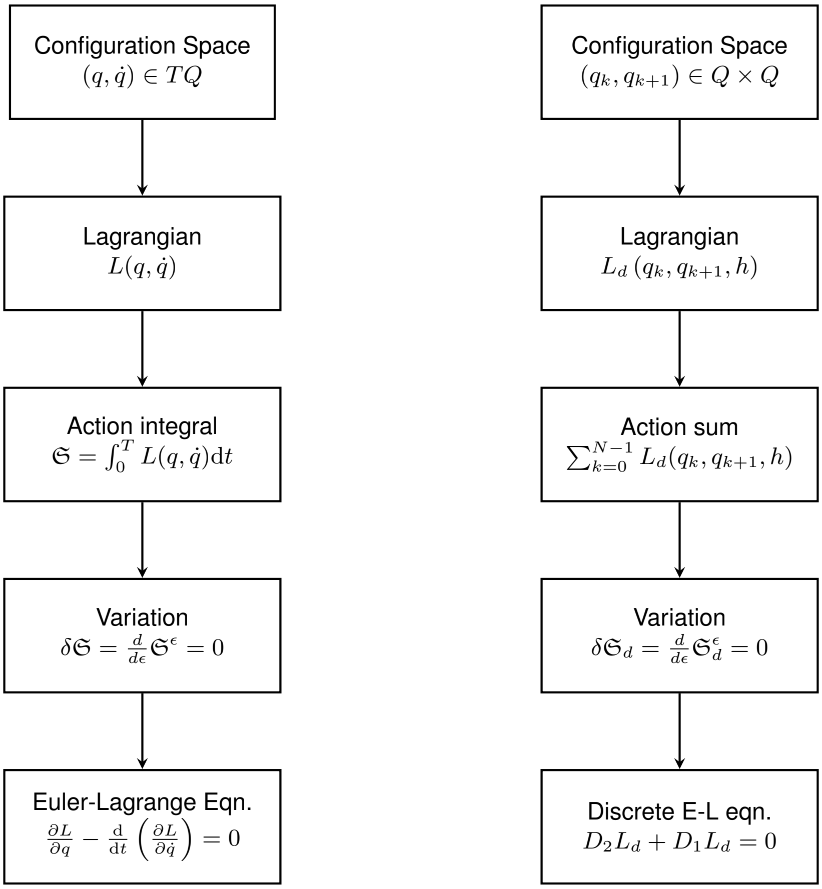


Figure 2.5: Derivation of the continuous and discrete equations of motion, an abridged version of the diagram in [88] Here, TQ represents the tangent bundle of the differentiable manifold Q.

---

[4] Hamiltonian systems of differential equations used to describe the evolution of physical systems posses the structure-preserving property of symplecticity. Mathematically, this means that their flow on the state space conserves a particular two-form. In addition, symmetry actions in the phase space lead to a conservation of momentum maps [88].

Instead of discretizing the continuous equations of motion, the discrete variational mechanics equations are derived starting from the discrete Lagrangian $L_d(q_k, q_{k+1}, h)$. This can be thought of as an approximation of the action integral along the curve between q_k and q_{k+1}, for instance, by using finite differences (2.46). Analogous to the continuous case, variations of the action sum are calculated. The endpoints are fixed ($\delta q_0 = 0$ and $\delta q_N = 0$) and discrete integration by parts is used, resulting in

$$\delta \sum_{k=0}^{N-1} L_d(q_k, q_{k+1}, h) = \sum_{k=0}^{N-1} \left[D_1 L_d\left(q_k, q_{k+1}, h\right) \cdot \delta q_k + D_2 L_d\left(q_k, q_{k+1}, h\right) \cdot \delta q_{k+1}\right] \tag{2.47}$$

$$= \sum_{k=1}^{N-1} \left[D_2 L_d\left(q_{k-1}, q_k, h\right) + D_1 L_d\left(q_k, q_{k+1}, h\right)\right] \cdot \delta q_k \tag{2.48}$$

where $D_i, i = 1, 2$ denotes derivative with respect to the i-th component of L_d. Requiring that the variations of actions must be zero for any choice of δq_k, the discrete Euler-Lagrange equations are obtained. For each k, the equations are of the following form

$$D_2 L_d\left(q_{k-1}, q_k, h\right) + D_1 L_d\left(q_k, q_{k+1}, h\right) = 0 \tag{2.49}$$

These equations give a map

$$F_{L_d} : (q_k, q_{k+1}) \mapsto (q_{k+1}, q_{k+2}) \tag{2.50}$$

This can be thought of as a one-step integrator for the system, starting with initial conditions (q_0, q_1). The discrete equations of motion for forced systems can be found using the Lagrange-D'Alembert's principle, and be represented as

$$D_2 L_d\left(q_{k-1}, q_k, h\right) + D_1 L_d\left(q_k, q_{k+1}, h\right) + f_{k-1}^{+} + f_k^{-} = 0 \tag{2.51}$$

where f_{k-1}^{+} and f_k^{-} are the discrete generalized forces. These integrators have the advantages of being

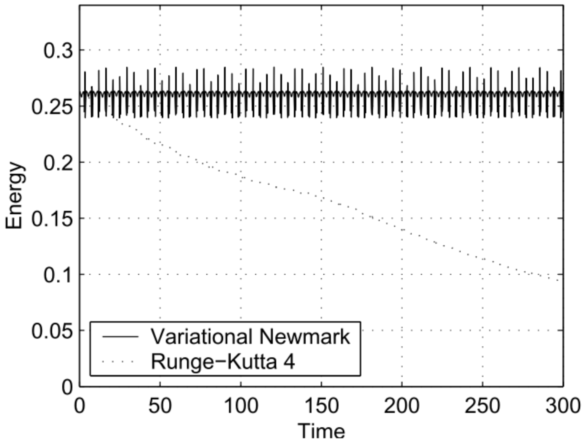


Figure 2.6: An illustration of the bounded energy behaviour of variational integrators shown for a conservative system. Unlike Range-Kutta (dashed line), the variational Newmark integrator does not artificially dissipate energy [88]

computationally efficient and stable. They also yield numerical solutions that match those given by the continuous equations of motion. Owing to the variational method of derivation, they are symplectic and respect energy dissipation (fig.2.6) and momentum conservation, even in long term simulations. They elegantly handle holonomic constraints [89] and can be used to design optimal control methods that inherit the advantages of forward integration [90]. Techniques for scalability and recursive computation of these integrators have also been developed using special coordinates [91].

2.6.1 Lie Group Variational Integrators

The evolution of many physical systems, including rigid bodies, is properly defined on a manifold and only locally defined on a Euclidean state space $\mathbb{R}^n$ [92]. The rigid body configuration space has a Lie group[5] structure, called the special Euclidean group, $\mathrm{SE}(3)$. As illustrated earlier, general numerical integration methods like the Range-Kutta do not preserve this Lie group structure or the symplectic and momentum conservation properties of the Lagrangian or Hamiltonian dynamics. Lie group variational integrators (LGVI) are a subclass of variational integrations which discretize the equations of motion for rigid bodies [94–96]. These integrators preserve the structure of the Lie group as the system evolves, without the need for constraints or reprojection. Additionally, they are symplectic, momentum-preserving, and exhibit good total energy behavior. Lie group variational integrators, specifically dual quaternion variational integrators, are used to describe rigid body dynamics in Chapter 5.

---

[5] Mathematically, a Lie group is a topological group that is also a smooth, differentiable manifold. Key properties of Lie groups useful for practical application are summarized in [93]

Chapter 3

Literature Review

This chapter covers research adressing the subject of maneuvering with parametric uncertainty, dividing the state-of-the-art into parameter estimation, disturbance-rejecting control, and learning during task execution (simultaneous learning). Of these, parametric uncertainty in spacecraft is mainly handled by offline estimation processes or disturbance rejecting control considering a known level of uncertainty bounds. Simultaneous learning, when proposed, is in the form of indirect adaptive control approaches, where an estimator is combined with an adaptive controller. Further, a limited number of in-space demonstrations of these algorithms exist. The presented literature study reveals a need for real-time, safe and scalable learning approaches suitable for implementation on physical systems.

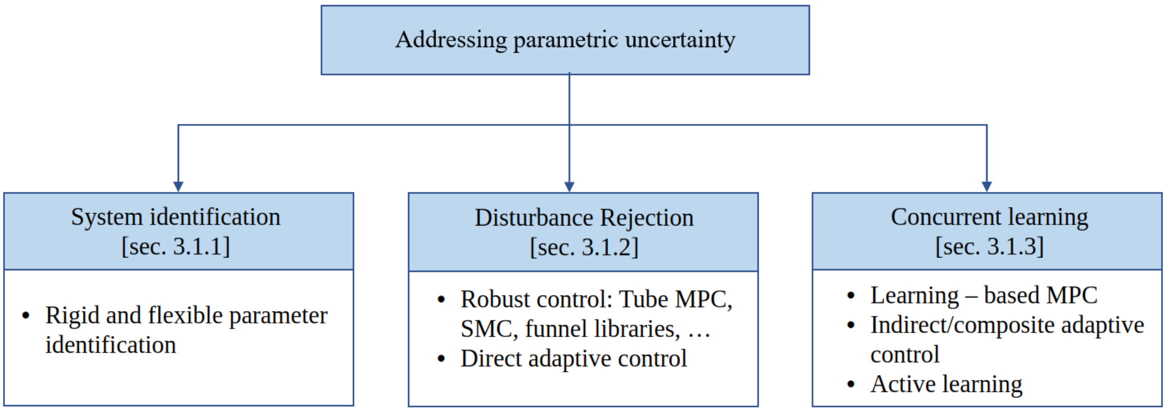


Figure 3.1: Classification of methods for coping with parametric uncertainty

3.1 Parametric uncertainty reduction

Existing methods for reducing model uncertainty in dynamical systems are classified into three broad groups. The first class of methods performs upfront system identification (sysID). It uses optimal excitation design (sec.2.3.1) to facilitate accurate parameter estimation. Excitation trajectories yield information-rich and well-conditioned measurement data, thus facilitating the process of accurate estimation. The controller then uses the learned system model for precise task execution. The second group of methods tackles this problem by treating the effects of uncertainty as unwanted, bounded disturbances and designs disturbance-rejecting control techniques to achieve control objectives despite them. The third and final group can be seen as an intersection between the two previous groups. It attempts to learn the dynamic parameters online to use them in controlling the system. The result is an improvement in the knowledge of the system model while also fulfilling the primary tasks. Robust control approaches can sometimes be conservative, and simultaneous learning methods can combine optimality and get maximum performance from the system without being overly cautious.

3.1.1 Upfront system identification

As the name suggests, upfront sysID performs complete system characterization via offline or recursive approaches. In addition to performing accurate model-based control, sysID might be relevant for optimal planning or fuel budgeting. Visual system identification approaches exist [97, 98] and are vital for remote motion estimation of target and the eventual nullifying of momentum. However, identification methods that use motion models are more suited for a free-flyer payload transport scenario, as more accurate estimates can be obtained by measuring the system's reaction to forces, provided an appropriate motion model is used.

Rigid body system identification

For upfront sysID, a linear dependence on the inertial parameters of interest is formulated using the equations of motion. To reiterate (2.19), this formulation is

$$\gamma(\boldsymbol{x}, \dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}})\boldsymbol{\pi} = \boldsymbol{\tau} \tag{3.1}$$

where $\gamma \in \mathbb{R}^{n \times r}$ is composed of robot states and their derivatives, while $\boldsymbol{\tau} \in \mathbb{R}^n$ is a vector of forces and torques, and $\boldsymbol{\pi} \in \mathbb{R}^r$ the vector of parameters to be estimated. As sec.2.3 shows, this estimation problem can be solved in closed form using weighted least squares. In literature, methods based on the Newton-Euler (NE) equations of motion have been used to tackle this problem in the case of terrestrial fixed manipulators, [74, 99, 100] and quadrotors [101] among others. This approach has also been extended to the identification of free-flying spacecraft. A key challenge in using this approach is that the NE equations depend on accelerations, which are noisy when measured. Often, localization algorithms filter and combine data obtained from multiple sensors to provide pose estimates. This data might need to be differentiated to obtain measurements of the accelerations. Performing numerical derivatives/ finite difference approximations or integration operations on measured states amplifies noise, and additional filtering steps are needed. Workarounds in past research include using second-order filtering to eliminate dependence on accelerations [21, 25], or using an integrated version of the Euler equations [102].

On the other hand, the method of conservation of angular momentum can avoid these issues. It uses the equations of momentum balance, which solely depend on velocities. Free-floating mode is

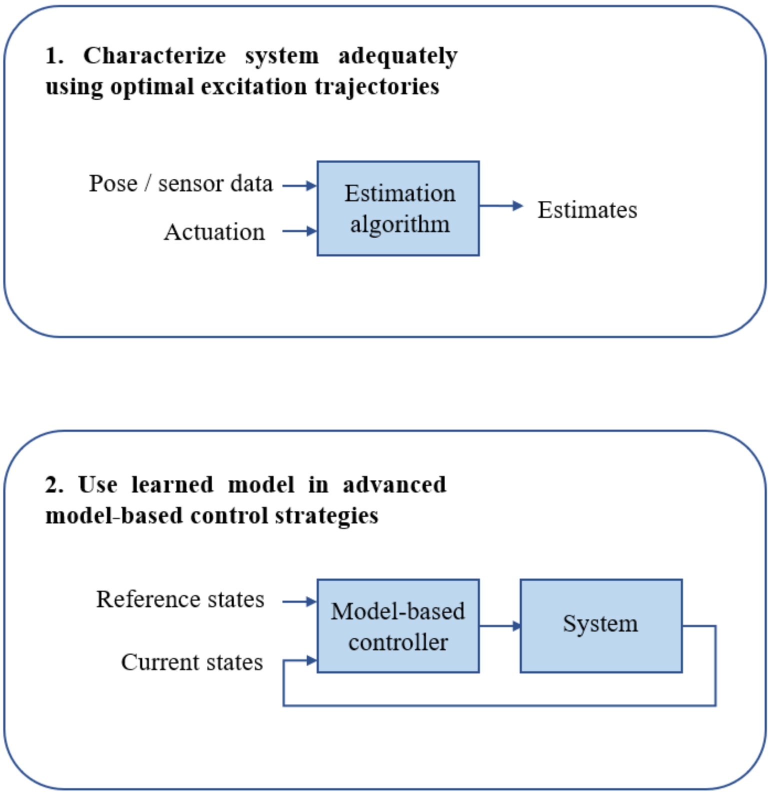


Figure 3.2: Upfront sysID aims to adequately characterize the system before precise task execution.

considered, which saves thruster fuel, and only the manipulator joints are excited. Moreover, any unknown/unmodeled forces like friction will not affect the system identification results. Some or all of the robot parameters can be identified using the Angular Momentum Conservation principle (AMC). For instance, the use of manipulator motion to excite the base in free-floating mode for inertial parameter estimation was proposed in [22]. The dynamic parameters of the arm are assumed to be known. By modeling the disturbance from gravity gradient torques and using measurement data from the reaction wheels, a reduced set of parameters (mass and moments of inertia) of the manipulator links and the robot base was estimated in [26]. This method was validated on telemetry data obtained from the Japanese ETS-VII satellite. The parameters of the object handled by the robot were estimated using the momentum formulation in [42]. More recently, works such as [45] estimate the complete inertial parameters of the robot by first solving a single-body problem (the joints of the arm are locked, and thrusters are fired). Then, a two-body problem is solved by unlocking only one joint at a time in free-floating mode. Particle Swarm Optimization (PSO) is used for solving the estimation problem. In works such as [46] and [47], a minimum set of parameters linear with respect to the momentum dynamics is derived and estimated. In the obtained estimation problem, the angular momentum is expressed as [46]

$$Y_h(\boldsymbol{x}, \dot{\boldsymbol{x}})\boldsymbol{\pi}' = \mathbf{h}_{CM} \tag{3.2}$$

where $\mathbf{h}_{CM}$ is the angular momentum, and Y_h is the regressor matrix composed of robot states $\boldsymbol{x}, \dot{\boldsymbol{x}}$.

The reduced set of parameters is represented by $\boldsymbol{\pi}'$. This set of parameters is enough to reconstruct the complete system dynamics for motion planning and control. A treatment similar to eq. 2.19 can then be used to solve the estimation problem. Both works use momentum control devices to maintain non-zero angular momentum. Results using the NE and AMC methods to estimate the robot base's inertial parameters and a grasped payload have been compared in [24]. The momentum formulation resulted in the most accurate estimates as it does not depend on acceleration measurements.

Instead of the discretized continuous equations of motion, a more recent approach based on discrete variational mechanics [103] employs the quaternion discrete variational equations for rotational dynamics from [104] to estimate spacecraft inertia. Analogous to the continuous equations of motion, variational integrators are derived from the discrete action sum, which is formed using the discrete Lagrangian. Moreover, these equations do not contain accelerations. Further, integrators derived from discrete variational mechanics are known for their energy and momentum conservation properties, and high-quality linearizations [105]. A detailed study of discrete variational mechanics is given in sec. 2.6. A linear dependence on the parameters of interest at time step k formulated using discrete variational mechanics has the following general form

$$\boldsymbol{H}(\boldsymbol{x}_k, \boldsymbol{x}_{k+1})\boldsymbol{\pi} = \boldsymbol{T}(h, \boldsymbol{\tau}_{k+1}) \tag{3.3}$$

where $h \in \mathbb{R}$ is the time-step, $\boldsymbol{\tau}$, the actuation forces and torques, and $\boldsymbol{x}_k$ and $\boldsymbol{x}_{k-1}$ represent the robot states at time k and $k+1$, analogous to the continuous mechanics states $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$.

Identification of flexible parameters/sloshing

A large number of spacecraft are equipped with flexible appendages. The presence of flexibilities in joints and links, fuel sloshing, solar panels, and antennas violates the assumption of a rigid body. Estimating properties of spacecraft with flexible appendages and sloshing and generating appropriate excitation maneuvers have been addressed [44, 106, 107]. In order to model fuel sloshing, multi-mass spring damper or pendulum equivalent models are used, while finite element methods are used to model link flexibilities. Joint flexibilities have been represented using spring-damper models. Estimation of flexible structures like trusses and beams, which could be pertinent to an on-orbit assembly problem, has been previously handled using the finite element method [108] or distributed parameter models [109], among others.

3.1.2 Disturbance rejecting control

Broadly, disturbance rejecting control arrives at a control solution for a system without accurately knowing its model parameters. This class of approaches is goal-driven, ensuring that the reference is tracked by, for instance, considering the worst bounds of uncertainty or updating the control law online in response to disturbances.

Robust control

Robust control methods aim to reject the effects of external disturbances or parametric uncertainties by means of a control law that guarantees stability for bounded disturbances. Nonlinear sliding mode control (SMC) [110] has been used for spacecraft attitude tracking in the presence of disturbances and bounded parametric uncertainties in works such as [27] and [28]. The discontinuous nature of the SMC control law causes high-frequency oscillations of the control variable, also known as chattering. Therefore, a high-performant sliding mode control might have significant fuel-consumption [29]. Higher order sliding mode control approaches are used to help with chattering alleviation [111].

Model predictive control (MPC) (sec. 2.4.2), which originated from process control industries, has been used in a variety of applications relating to dynamical systems. However, the performance of MPC depends on the system model, which makes it vulnerable to model inaccuracies and disturbances. Robust MPC approaches have been proposed to combat this. RMPC is a feedback control that optimizes over a control policy - a sequence of control laws - and ensures cost minimization over all possible uncertainty realizations. However, determining such a control policy is usually computationally challenging, and simplifying approximations are often used [30], [112].

Invariant tubes and funnel libraries

Tube-based MPC breaks down the RMPC problem into an open-loop control for nominal dynamics and a feedback controller, which ensures that the system response remains in a Robust Control Invariant (RCI) tube around the nominal trajectory. Constraint tightening is performed to account for uncertainty in the nominal constraints. The robustness is maximized by minimizing the tube size given control constraints and uncertainty bounds. In [31], a linear tube MPC is employed in a rendezvous scenario between a chaser and a target of uncertain dynamics. A feedback controller which minimizes the size of the robust positively invariant set is determined. This controller rejects disturbances caused by bounded, additive uncertainties in predicted target motion, guaranteeing feasibility and exponential stability within the tube of trajectories. Many tube MPC approximations are sub-optimal and conservative due to the complexity of computing invariant tubes and the inability to capture state-dependent uncertainty. Dynamic tube MPC has been proposed, where the tube geometry and open-loop trajectory are optimized simultaneously [113]. The result is online optimization of tube width as new obstacles are observed and the application of high-bandwidth control only when necessary. This approach has been extended to include set membership identification (SMID) to guarantee monotonic uncertainty reduction in model parameters, thus resulting in an Adaptive DTMPC scheme. Offline computation of a set of funnel libraries for robots with highly nonlinear and complex dynamics, operating in unknown environments with bounded dynamic uncertainties and disturbances, has been proposed in [114]. Starting from a finite library of open loop trajectories, a stabilizing controller that minimizes the worst-case reachable set of the system for bounded levels of uncertainty and disturbances is computed. Sum-of-squares programming (SOS) [115] is used to obtain finite dimensional semidefinite programs from the infinite-dimensional problem which involves searching over spaces of functions. The result is an outer approximation of the reachable set, or funnel, within which the closed loop system is guaranteed to remain. Once the funnel library is computed, motion plans are formed by sequential composition of funnels at runtime. In contrast, [116] uses contraction theory to compute invariant tubes and an offline tracking controller designed using control contraction metrics. Their framework is suited for real-time planning in densely cluttered environments and does not involve a computationally expensive offline calculation of funnel libraries.

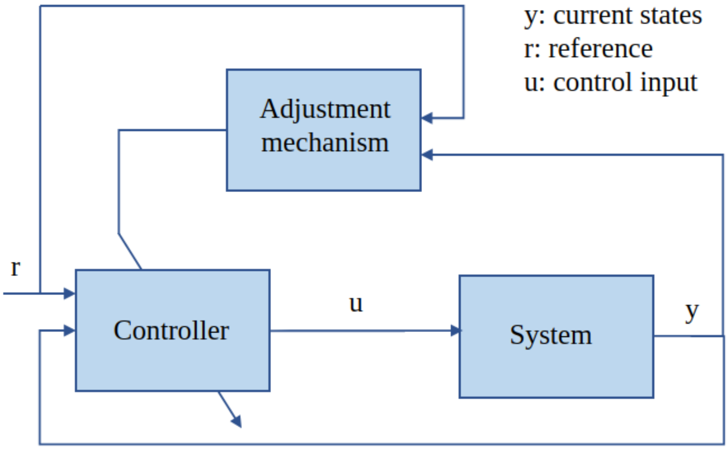


Figure 3.3: Representation of direct adaptive control, a disturbance rejecting control scheme. The adjustment mechanism may or may not contain a reference model. The control parameters are adjusted only to the extent necessary for tracking, and do not require system parameter convergence.

Adaptive control

In contrast to robust control, adaptive control copes with uncertainty by changing the control law online to ensure that the control objectives are met [117], [118]. Adaptive control is a combination of control and estimation framework, where system or controller parameters are adapted online and used for computing the control input. It is superior to robust control in dealing with uncertainties in constant or slowly-varying parameters and does not require a priori information of the uncertainty bounds [117]. Two main approaches exist for constructing adaptive controllers: Model Reference Adaptive Control (MRAC) and Self-Tuning Control (STC) [117, 119]. In MRAC, a reference model specifies the desired output of the control system, and an adaptation mechanism adjusts the parameters such that the plant's response matches that of the reference model. Unlike the adaptation mechanism used in MRAC, which is affected by choice of the control law, STC usually uses control and estimation frameworks that are independent of each other. The control gains are computed based on the results of the estimation. MRAC and STC can be designed by either estimating the plant parameters first and then computing control parameters (indirect scheme), or estimating the control parameters directly by parametrizing the plant model in terms of control parameters (direct scheme). Note that the parameter estimates will only converge to their actual values as long as the reference signal satisfies certain conditions, such as persistent excitation.

This section discusses direct adaptive control methods focusing on tracking without requiring parameter convergence. These methods are usually employed for dealing with disturbances that are hard to model or whose effects cannot be neglected, for instance, the effects of joint friction and damping in robotic space manipulators [34]. These methods have been explored in the context of in-orbit proximity operations, in the presence of significant inertial uncertainties such as free-flyer robotic post-capture stabilization [32] and undocking with tests on the SPHERES platform [33]. In [120], impedance-based adaptive control has been used for post-grasp stabilization by considering the grasped target as a disturbance force on the end-effector.

Model-free reinforcement learning

In Reinforcement Learning (RL), an agent learns goal-driven behaviors through its trial-and-error interactions with the environment [121]. Though RL finds its roots in the computer science community and the interaction of living organisms with the environment, it is connected with adaptive optimal control methods [122], and is increasingly being adapted to find optimal control policies. Model-free or non-parametric reinforcement learning (RL) approaches, where the agent learns policies that directly map states to actions through interaction with the environment, then fall under the disturbance rejecting control category. A growing number of such approaches are being proposed for space robotic systems with complex dynamics [123], for instance, [124], where a model-free policy was learned for online trajectory planning of a dual-arm free-floating space robot.

3.1.3 Concurrent learning

Concurrent or simultaneous learning attempts to reconcile the principles of pure system identification and disturbance-rejecting control by learning the parameters during task execution. Even though characterized adequately, the parameters of a system might change online for reasons such as fuel consumption or payload deployment [56]. Concurrent learning enables the system to learn its parameters online without interrupting primary task execution. Further, the quantities estimated on the go are often used to inform the control scheme, adapting it to current uncertainty levels.

Concurrent learning with adaptive control

Indirect adaptive control approaches offer the possibility of obtaining convergent estimates if the trajectories are persistently exciting. This is achieved by combining an estimator with an adaptive controller. For instance, in [54], a self-tuning prediction error-based adaptive Jacobian controller is used for trajectory tracking in the presence of dynamic and kinematic uncertainties for a robot in free-floating mode. The control scheme is combined with two Cushioned Floor (CF) estimators [55] for simultaneously estimating the robot's dynamic and kinematic parameters. In [56], the concurrent learning approach is used for controlling a robot in free-floating mode with unknown dynamic parameters. A self-tuning transposed Jacobian controller (for control in Cartesian space) is used to reject disturbance from accumulated reaction wheel momentum while estimating the system parameters using the Angular Momentum Conservation (AMC) method. A composite adaptation law that leverages MRAC and Extended Kalman Filter (EKF) is used in [52] to accomplish concurrent adaptive control and parameter estimation. The system continues to improve the estimates even after the tracking error has been eliminated, leading to a less energetic characterization and stabilization of the system. Further, works such as [53] have used nonlinear adaptive control to guarantee global asymptotic convergence of the position tracking error for spacecraft formation flying in the presence of unknown, constant or slow-varying parameters and disturbances. to achieve this, the controller contains an adaptation law which estimates the unknown parameters online.

Learning based model predictive control

As previously mentioned, Model Predictive Control optimizes the predicted motion of the vehicle over a time horizon. However, real dynamical systems operate with uncertainty, and usually, MPC does not use learned models or account for model uncertainty while applying constraints. Therefore, constraints could be violated if highly-accurate models are not used [125]. A class of methods that combines non-parametric learning with MPC addresses this problem by learning either the complete system model or a disturbance model online. This reduces the time spent on learning and designing trajectories offline, and maximum performance from the system can be obtained. Tools such as Gaussian process regression, locally Weighted Projection Regression, or Support Vector Regression [126] are commonly used for learning. The controller acts upon the composition of the nominal model and the learned dynamics. Constraint tightening ensures that constraint formulation takes uncertainty into account.

This approach has been applied to highly non-linear and complex models such as an autonomous racing car [35], [36]. In [125], a robust learning controller that is conservative in the initial trials and optimal and high-performance in the later trials when model uncertainty is reduced is proposed. Another group of methods intersects robust or adaptive control theory with MPC and model learning, thus ensuring that constraint satisfaction and stability guarantees are met as the MPC performance improves. A learning-based tube MPC (LBMPC) is proposed in [37]. If the system is sufficiently excited, then the LBMPC control action probabilistically converges to that of an MPC computed using the true dynamics. A Concurrent Learning Model Predictive Control approach (CL-MPC) proposed in [38] guarantees simultaneous system stability and exponential estimate convergence with excitation for a finite time.

Model-based reinforcement learning

Most works belonging to this category perform model and policy learning. Usually, the model is learned using the nonlinear approximation tools mentioned in section 3.1.3, and this model is used to find a (locally) optimal control policy that minimizes the total cost. Examples of this are works such as [39], and [127], where each trial of the algorithm roughly consists of the following phases: learning a simulation-based control policy by augmenting the nominal model with improvements based on real-world observations, and data collection and performance evaluation in the real world. A major limitation of reinforcement learning is its need for a large amount of data for training. Performing numerous trials in the real world during the exploration phase is not only time-consuming but could potentially be unsafe (using a simulator in model-based RL is thus widespread). Therefore, data-efficient and safe RL are burgeoning

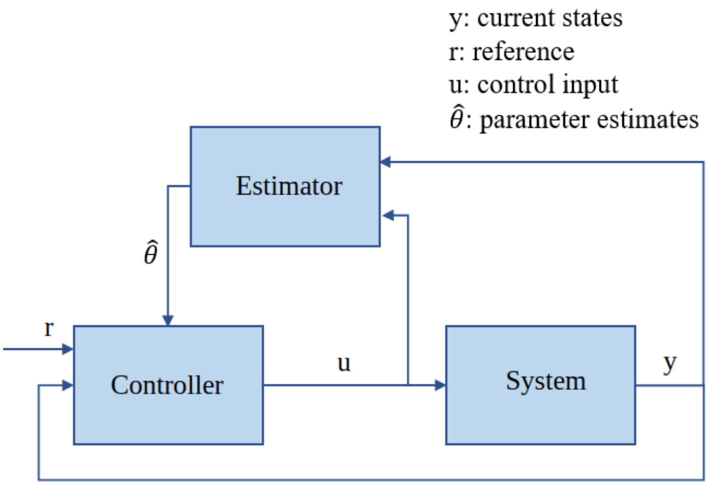


Figure 3.4: Schematic of the concurrent or simultaneous learning approach. The controller is continually updated with the latest model estimates.

domains of RL research.

Active learning

The idea behind active learning or optimal experimental design is that efficiency and greater accuracy can be achieved in the learning process if the algorithm is allowed to choose the data from which it learns [128]. For instance, active RL approaches are being proposed to mitigate the need for large amounts of data [40], where, instead of random sampling, the algorithm actively selects potentially informative samples during the exploration phase. Similarly, information seeking motion plans are often used in combination with parameter or state estimation algorithms with the aim of gathering information-rich measurements to improve estimation accuracy.

Active learning and dual control

Dual control [129] methods aim to perform system identification and control simultaneously. Following the classification in [130], dual control methods are typically explicit or implicit. The former class of approaches explicitly induces information-seeking behavior or estimate convergence (for instance, by maximizing Fisher information in the cost function). In this approach, the interaction between the controller and the dynamic parameter estimator is similar to that of a self-tuning controller: the estimator uses the control inputs and the measurements, and the controller uses the latest estimates to update the dynamic model. For example, a Fisher Information maximizing trajectory is tracked using closed-loop receding-horizon control in [49].

The implicit class of methods approximates the stochastic optimal control problem. The resulting control policy is thus inherently information-seeking. In [41], Partially Observable Markov Decision Processes (POMDPs) are used to plan control policies online that optimally trade-off parameter estimation by penalizing posterior covariance and fulfillment of control objectives. This problem is posed as a Bayes-Adaptive Markov Decision Process (BAMDP) in [50], where Monte Carlo tree search (MCTs) plans control actions that reduce model uncertainty[1]. This approach could be used in real-time, similar to [41]. In [131], the problem of obtaining information-gathering motion plans in belief space (the space of probability distributions of the state, over which POMDPs are defined) is posed as a covariance-free optimization (partial collocation) problem, leading to about four hundred times reduction in computing time as compared to planning approaches that include covariance in the state.

3.2 Research gap

Updating current knowledge of the system model is a choice that depends on the mission objectives. Having touched upon some of the benefits of model learning in section 1, particularly when it comes to model-based control and optimal planning, this thesis assumes a scenario where model learning is relevant and focuses on the research problem of uncertainty reduction. From that point of view, disturbance rejection approaches do little to improve the system model, aiming primarily to ensure stability and reduce tracking error in the presence of uncertainties. Approaches based on concurrent learning adaptive control also perform model learning so that the closed-loop system model converges to the reference model, thus recovering its stability and transient properties. However, optimal solutions under state and input constraints are usually challenging to obtain with adaptive control [38]. On the other hand, Model Predictive Control is a powerful tool for online optimization and enforcing constraints, suitable for operation under dynamic environments. However, MPC's stability and constraint satisfaction cannot be guaranteed in the presence of uncertain dynamics. Learning-based NMPC approaches remedy this by using the learning model to capture the current dynamics online for the NMPC to optimize. This also alleviates the need for over-cautious control.

Many learning-based NMPC methods use Gaussian Process Regression (GPR) for model or uncertainty learning. GPR is a potent and flexible tool for accurate function approximation in high dimensional space. However, this comes at the cost of computational complexity, which scales cubically with the number of training examples for GPR [132]. The combination of GP and NMPC has been experimentally validated for scenarios where either the GP was learned over multiple trials or offline and updated online. Developing online and computationally efficient learning approaches is a subject of active research. Approximations such as sparse GP and local GP, using a moving window, or maintaining an online dictionary of data points, have been proposed to speed up the computation [35, 48]. Real-time capabilities of learning-based MPC have not been the focus of past research. Moreover, implementing active learning and dual control in a constrained system is an open question, as most learning-based MPC methods use passive learning [130].

As the tradeoff between exploration and exploitation is often heuristic, explicit active learning approaches provide a certain flexibility, i.e., information-seeking behavior can be activated when needed. Moreover, these approaches are usually not computationally expensive. On the other hand, implicit approaches account for stochasticity in the dynamics, are capable of optimally trading off exploration and

---

[1] Since the underlying formulation of reinforcement learning problems is commonly a Markov Decision Process (MDP), implicit active learning approaches that use extensions of MDPs such as POMDPs and BAMDPs could also be classified under active RL

exploitation but are computationally expensive and not easily scalable to higher dimensions [130]. To the author's knowledge, works on implicit active learning have been tested in simulation and on real robots restricted to planar dynamics.

As the literature shows, the problem of model uncertainty manifests in multiple applications involving robotic systems. This thesis considers the free-flyer cargo transportation scenario, where the robot has grasped an object of uncertain inertial properties, resulting in parametric uncertainty of the combined system. This application is motivated by guest science access to NASA's Astrobee free-flyer and its relevance to autonomous on-orbit tasks tasks like assembly and cargo unpacking. Considering these applications, a **real-time** and **scalable** learning approach is of interest. Given a dynamic environment where the robot would work alongside other free-flyers and crew, the ability to replan would be essential. At the same time, safety during model learning is important. Concurrent adaptive control methods provide stability and convergence guarantees, while in RL and learning-based MPC, approaches such as safety filters [130] and probabilistic shields [133] have been proposed for safety during the learning phase. To the author's knowledge, **safety** has not been the focus of real-time implicit and explicit dual control approaches. Since these approaches often incorporate the latest estimates into the controller models, safety during the transient phase and in the face of estimator instability would also be beneficial in addition to safety during exploration.

To that end, this thesis aims to develop explicit active-learning approaches that are real-time, scalable and uncertainty-aware. Leveraging the availability of the Astrobee free-flyer platform for guest science use, a micro-gravity demonstration of the developed approach is envisioned. Further, the state of the art reveals that inertial parameter estimation of free-flying robots using the equations of motion suffers from the issue of noisy measurements, accelerations in particular. Using the angular momentum conservation approach [46] circumvents this, but requires the robot to be in free-floating mode. This is not feasible for robots operating in clutted, dynamic environments. Therefore, investingating free-flyer parameter estimation methods that produce accurate estimates inspite of noisy measurements is another research objective that this thesis aims to address.

3.3 Microgravity autonomy demonstrations

Due to the limited, albeit growing, accessibility of microgravity platforms, few demonstrations of robotic autonomy algorithms have been performed in space. These demonstrations have either used experimental free-flying platforms deployed on board the International Space Stations (ISS) or data gathered from spacecraft. In the experiments that tested parameter learning, the focus has been on the estimation problem and not on information-aware learning. For instance, in [26], the authors used telemetry data obtained from the ETS-VII satellite to estimate a reduced set of inertial parameters of a space manipulator. Works such as [97, 98] used the Synchronized Position Hold Engage Reorient Experimental Satellites (SPHERES) onboard the ISS as a testbed for evaluating vision-based navigation and estimation algorithms, estimating properties of a spinning target. Measurements obtained from the ROKVISS robotic arm installed outside the ISS were used to estimate friction and stiffness parameters in space

[134]. Given that Astrobee was deployed on the ISS recently (the first Astrobee, Honey, had its first flight in April 2019), the software payload for the experiments presented in this thesis was one of the first ones to make significant changes to Astrobee's autonomy stack. To the author's knowledge, this was the first time an information-aware inertial parameter estimation framework was tested in microgravity.

Chapter 4

Offline inertial estimation using Fourier-based noise filtering

Accurate knowledge of a system's dynamic properties considerably enhances the precision and safety of in-space task execution using model-based control strategies. Inertial properties such as mass and center of mass, even if known perfectly, may vary during operation, creating the need for on-board estimation. As discussed in sec. 3.1.1, the equations of motion approach for rigid-body estimation enables a formulation of an estimation problem linear in the parameters of interest. The drawback of this approach is its requirement of acceleration measurements and actuation forces and torques, which can be noisy and difficult to measure. In contrast, approaches that use the conservation of angular momentum do not need acceleration measurements, but the linear problem formulation is more convoluted. Works like [22, 42, 47] propose operating the robot in free-floating mode and using the momentum conservation approach, but do not estimate all parameters of the space manipulator system. If reaction wheels are use to generate a non-zero angular momentum, then the dynamic coupling between the robot base and the manipulator (sec.2.1.2) permits complete identification in the free-floating mode [46].

Considering that the free-floating mode might be infeasible for cluttered, dynamic, intra-vehicular environments, and that reaction wheels may not be available for systems operating in such environments, this section presents an approach for free-flyer inertial parameter estimation developed based on the Newton-Euler equations of motion. As previously noted, raw acceleration data are often noisy. Further, angular accelerations needed for the estimation algorithm are not measured in practice and are obtained by numerical differentiation of measured velocities, thus amplifying the noise content in the signal. To address this, the proposed approach harnesses the continuous nature of the Fourier Series by performing analytical differentiation of the measured data. For this, the robot tracks multiple cycles of an optimized Fourier Basis excitation trajectory (trajectory design for identification is addressed in section 2.3.1). The obtained data is then low-pass filtered using an energy-balance approach to determine the high-frequency cut-off. The purpose of the energy balance approach is two-fold: keep out the high-frequency noise but retain high-frequency data associated with non-linearities in the robot model. Parts of this work were published in [57–59].

The proposed method is inspired by [74] and [100], where a periodic Fourier Series excitation was used to enable time-domain averaging, filtering, and analytical differentiation of the measurement data for maximum-likelihood parameter identification of ground-based manipulators. While the excitation trajectory used in [74] was planned in the joint space, the proposed approach plans the excitation trajectory in the six-dimensional free-flyer robot workspace. The commanded trajectories are band-limited in both cases, i.e., they only consist of a known and finite number of Fourier series harmonics. Specifying the bandwidth of the trajectories allows, for instance, to prevent or induce excitation of the robot's flexible modes as needed. It also facilitates noise-filtering, as all high-frequency harmonics outside the bandwidth can be attenuated. To ensure that the executed trajectory is band-limited, [74] uses a band-limited position-feedback controller. However, due to the non-linearities of rigid-body dynamics in the SE(3), as well as the potential existence of other unmodeled effects, no guarantees on the band-limited nature of the executed trajectory can be provided for the free-flyer case. The proposed approach, therefore, differs from [74] in the measurement filtering phase - an appropriate number of Fourier Series harmonics that filter out the high-frequency noise but fit the measurement data is computed.

The task of on-orbit payload transportation is considered as the underlying scenario for the proposed approach. The robot has grasped an uncharacterized payload leading to an increase in uncertainty about the inertial properties of the robot-object system, and the goal is to determine the inertial parameters of the complete system. For simplicity, the grasp and the cargo are both assumed to be rigid. The manipulator is assumed to be fixed and the excitation is provided to the robot base.

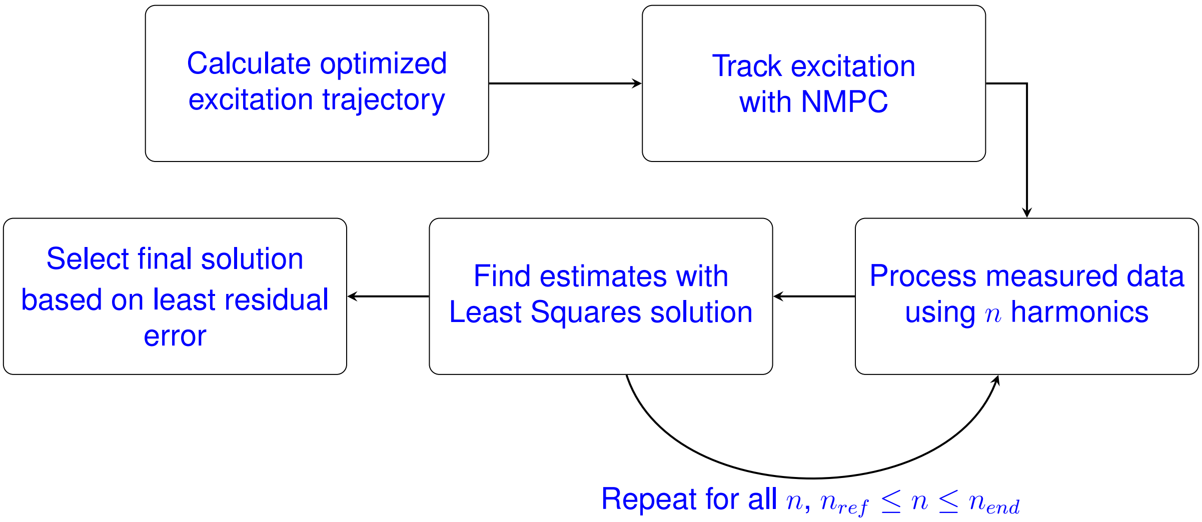


Figure 4.1: An overview of the parameter estimation method - TODO add the new figure

4.1 Formulation of the estimation problem

Starting with the rigid body dynamics in section 2.1, and using (2.6), we obtain

$$\begin{bmatrix} \boldsymbol{\tau} \\ \boldsymbol{f} \end{bmatrix} = \begin{bmatrix} J & mS(\mathbf{c}) \\ mS^T(\mathbf{c}) & m\mathbf{I}_{3\times 3} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{\omega}} \\ \dot{\boldsymbol{v}} \end{bmatrix} + \begin{bmatrix} S(\boldsymbol{\omega})J\boldsymbol{\omega} \\ mS(\boldsymbol{\omega})S(\boldsymbol{\omega})\mathbf{c} \end{bmatrix} \tag{4.1}$$

where J represents the inertia in the body frame, J_B, and $S()$ represents the skew-symmetric matrix operator. Rearranging (4.1) leads to

$$\begin{bmatrix} \boldsymbol{\tau} \\ \boldsymbol{f} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3\times1} & -S(\dot{\boldsymbol{v}}) \\ \dot{\boldsymbol{v}} & S(\dot{\boldsymbol{\omega}}) + S(\omega)S(\omega) \end{bmatrix} \begin{bmatrix} m \\ m\mathbf{c} \end{bmatrix} + \begin{bmatrix} J\dot{\boldsymbol{\omega}} + S(\boldsymbol{\omega})J\boldsymbol{\omega} \\ \mathbf{0}_{3\times1} \end{bmatrix} \tag{4.2}$$

Further, $J\boldsymbol{\omega}$ can be written as $G(\boldsymbol{\omega})\,\mathbf{J}$, where $G()$ performs the following operation

$$G(\boldsymbol{\omega}) = \begin{bmatrix} \omega_x & \omega_y & \omega_z & 0 & 0 & 0 \\ 0 & \omega_x & 0 & \omega_y & \omega_z & 0 \\ 0 & 0 & \omega_x & 0 & \omega_y & \omega_z \end{bmatrix} \tag{4.3}$$

and

$$\mathbf{J} = \begin{bmatrix} J_{11} & J_{12} & J_{13} & J_{22} & J_{23} & J_{33} \end{bmatrix}^T \tag{4.4}$$

Similarly, $J\dot{\boldsymbol{\omega}}$ can be written as $G(\dot{\boldsymbol{\omega}})\,\mathbf{J}$. This yields a relation linear in the inertial parameters,

$$\begin{bmatrix} \boldsymbol{\tau} \\ \boldsymbol{f} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3\times1} & -S(\dot{\boldsymbol{v}}) & G(\dot{\boldsymbol{\omega}}) + S(\boldsymbol{\omega})G(\boldsymbol{\omega}) \\ \dot{\boldsymbol{v}} & S(\dot{\boldsymbol{\omega}}) + S(\omega)S(\omega) & \mathbf{0}_{3\times6} \end{bmatrix} \begin{bmatrix} m \\ m\mathbf{c} \\ \mathbf{J} \end{bmatrix} \tag{4.5}$$

Then, an estimation problem linear in the inertial parameters can be set up as:

$$\boldsymbol{\Gamma}\boldsymbol{\pi} = \mathbf{b} \tag{4.6}$$

where

$$\boldsymbol{\Gamma} = \begin{bmatrix} \boldsymbol{\gamma}(X(1), \dot{X}(1), \ddot{X}(1)) \\ \vdots \\ \boldsymbol{\gamma}(X(N), \dot{X}(N), \ddot{X}(N)) \end{bmatrix} \tag{4.7}$$

Here, $\boldsymbol{\gamma}(\mathbf{X}, \dot{\mathbf{X}}, \ddot{\mathbf{X}})$ is the regressor matrix composed of measurements of the pose (position and attitude in Euler angles), $X(k)$ = $[x(k), y(k), z(k), \phi(k), \theta(k), \psi(k)]^T$ at time step k, for $k = 1, ..., N$.

$$\boldsymbol{\gamma} = \begin{bmatrix} \mathbf{0}_{3\times1} & -S(\dot{\boldsymbol{v}}) & G(\dot{\boldsymbol{\omega}}) + S(\boldsymbol{\omega})G(\boldsymbol{\omega}) \\ \dot{\boldsymbol{v}} & S(\dot{\boldsymbol{\omega}}) + S(\omega)S(\omega) & \mathbf{0}_{3\times6} \end{bmatrix} \tag{4.8}$$

The vector of the parameters to be estimated, $\boldsymbol{\pi} \in \mathbb{R}^{10}$, is

$$\boldsymbol{\pi} = \begin{bmatrix} m & m\mathbf{c} & J_{11} & J_{12} & J_{13} & J_{22} & J_{23} & J_{33} \end{bmatrix}^T \tag{4.9}$$

and

$$\mathbf{b} = \begin{bmatrix} \mathcal{T}(1) \\ \vdots \\ \mathcal{T}(N) \end{bmatrix} \tag{4.10}$$

and $\mathcal{T}(k)$ is the vector of the applied forces and torques ($\boldsymbol{\tau}, \boldsymbol{f} \in \mathbb{R}^3$)

$$\mathcal{T} = \begin{bmatrix} \boldsymbol{\tau} \\ \boldsymbol{f} \end{bmatrix} \tag{4.11}$$

If the N measurements of the pose, X, their velocities and accelerations can be assumed to be devoid of noise for practical purposes, then the matrix $\mathbf{H}$ can be said to be noise-free. The data-processing steps in section 4.3 aim to filter out noise from the measurements, such that this assumption holds. Considering that the noise on the forces and torques is Gaussian, i.e., $\mathcal{T}_i \sim N(0, \Sigma_i)$, where $i = 1, ..., 6$, the maximum likelihood solution (sec. 2.2) is then given by (4.12). The weights are the reciprocal of $\boldsymbol{\Sigma}$, the $6N \times 6N$ diagonal covariance matrix of the measured forces and torques from time instances 1 to N.

$$\hat{\boldsymbol{\pi}} = (\boldsymbol{\Gamma}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\Gamma})^{-1} \boldsymbol{\Gamma}^T \boldsymbol{\Sigma}^{-1} \mathbf{b} \tag{4.12}$$

If the measured forces and torques are considered to be independent and identically distributed (iid), the same standard deviation (4.12) can be written without the weights, as $\hat{\boldsymbol{\pi}} = (\boldsymbol{\Gamma}^T \boldsymbol{\Gamma})^{-1} \boldsymbol{\Gamma}^T \mathbf{b}$.

4.2 Generation of the excitation trajectory

As discussed in section 2.3.1, an appropriate excitation trajectory is necessary to ensure that the regressor $\boldsymbol{\Gamma}$ is full-rank and has sufficient information content for parameter estimation. To harness the continuity, periodicity of the Fourier Series and to facilite filtering of high-frequency noise, the trajectory is modeled using Fourier basis functions (the Fourier Series is discussed in detail in 2.5). The time varying references for each of the 6 states, $(x, y, z, \phi, \theta, \psi)$ from X are a sum of *n* sinusoidal harmonics in the following manner

$$\begin{aligned} X_i(t) &= a_{io} + \sum_{k=1}^{n} \frac{a_{ik}}{\omega_f k} sin(\omega_f k t) - \frac{b_{ik}}{\omega_f k} cos(\omega_f k t) \\ \dot{X}_i(t) &= \sum_{k=1}^{n} a_{ik} cos(\omega_f k t) + b_{ik} sin(\omega_f k t) \\ \ddot{X}_i(t) &= \sum_{k=1}^{n} -a_{ik} \omega_f k sin(\omega_f k t) + b_{ik} \omega_f k cos(\omega_f k t) \end{aligned} \tag{4.13}$$

where ω_f is the angular frequency, $\omega_f = 2\pi/T_f$, with T_f denoting the period of one excitation cycle. Selection of the period T_f is a design choice, influenced by factors such as acceleration of the robot. In case the robot's acceleration limits are low, a larger excitation period allows adequate exploration of the permissible robot workspace, potentially leading to an overall improvement in estimation accuracy. Morevoer, slow-paced and longer trajectories could also prevent input saturation which can adversely affect estimate accuracy as section 4.5 addresses. A low number is preferred for the selection of n, as a commanded trajectory composed of high frequency harmonics would be challenging to track as well as filter from other high-frequency noise.

The Fourier coefficients, a_{io}, a_{ik}, b_{ik} for k = 1 to n, and i from 1 to 6 are denoted by vector $\boldsymbol{\delta}$, which can be obtained by optimizing one of the two following criteria used in [74]:

1. V_1**, Minimizing the condition number** of the $\mathbf{\Gamma}$ matrix which has been normalized. With a small condition number, the estimation is less sensitive to noise from the measurements. This was observed to result in impractical trajectories in some cases, where the maximum magnitude of the states along the trajectory was of the order $10^{-3}m$. To tackle this, a regularization term is introduced to impose a penalty on the magnitude of the trajectory. In addition to the minimization of $cond(\mathbf{\Sigma}^{-0.5}\mathbf{\Gamma})$, the Root Mean Square (RMS) level of each Fourier series or the square root of its power is maximized, resulting in the following cost function

$$V_1 = cond(\mathbf{\Sigma}^{-0.5}\mathbf{\Gamma}) - \sum_{i=1}^{6} q_i \sqrt{\frac{1}{N}\sum_{\eta=1}^{N} |X_i(\eta)|^2} \tag{4.14}$$

   Where N is the number of samples present in each series X_i, from time 0 to T_f, and $\boldsymbol{q}$ is the vector of weights given to the waveforms corresponding to each of the six states $(x, y, z, \phi, \theta, \psi)$.

2. V_2**, Maximizing the log determinant of the Fisher information matrix**. The Fisher Information Matrix (FIM) criterion is a measure of the amount of information the measured variable gives about the parameter to be estimated (2.2). As the maximum likelihood estimator achieves the Cramér-Rao lower bound, maximizing the Fisher information is equal to minimizing the lower bound on the estimate covariance. Here, the log determinant of the FIM related to the MLE of the parameters (section 2.3) is maximized, known as the D-optimality criterion [70].

$$V_2 = -log\left[\det(\mathbf{\Gamma}^T\Sigma^{-1}\mathbf{\Gamma})\right] \tag{4.15}$$

Using $X(t), \dot{X}(t), \ddot{X}(t)$ from eq. (4.13) for $t = 1$ to N, the trajectory optimization problem with the optimization variables being the vector of Fourier Series coefficients, $\boldsymbol{\delta}$, is formulated as:

$$\begin{aligned}
\underset{\boldsymbol{\delta}}{\text{minimize}} \quad & V(X, \dot{X}, \ddot{X}) \\
\text{subject to:} \quad & X(1) = 0, \dot{X}(1) = 0, \ddot{X}(1) = 0 \\
& X(N) = 0, \dot{X}(N) = 0, \ddot{X}(N) = 0 \\
& X_{min} \leq X(t) \leq X_{max} \\
& \dot{X}_{min} \leq \dot{X}(t) \leq \dot{X}_{max} \\
& \ddot{X}_{min} \leq \ddot{X}(t) \leq \ddot{X}_{max}
\end{aligned}$$

such that the trajectory starts and ends with the robot at rest. Here, V represents the criterion chosen from V_1 and V_2. The robot motion is constrained to lie within feasible values of the state. Note that this step is independent of the knowledge of the inertial parameters of the robot. Nonlinear model predictive tracking control (described in sec. 2.4.2) is used to execute this trajectory.

4.3 Processing of measured data

We assume that a state estimation algorithm is available to provide robot pose estimates as it tracks multiple cycles of the excitation trajectory. Pose data corresponding to the executed trajectory is then processed before using it for inertial parameter estimation. The data processing steps leverage the periodic and band-limited nature of the excitation trajectory to improve the signal-to-noise ratio of the measured data and thus obtain accurate parameter estimates.

Noise filtering

First, the robot is made to track C cycles (C is taken as 10 in our simulations) of the excitation trajectory. As the excitation trajectory is periodic, the values of positions and attitude are averaged over multiple periods, in order to improve the signal-to-noise ratio of the data. The initial measurement data are usually disturbed by transients. The transients were observed to die out in the first cycle, so data from the second cycle onwards is used for calculating the average.

Using the Fourier transform, frequencies higher than those in the reference excitation trajectory can be filtered out as noise. However, measurement noise is not the only source of higher frequencies in the executed trajectory. Deviations from the reference caused by nonlinearities in the dynamics of the system also causes high frequency harmonics to appear in the system response. Further, actuation limits of the system make it vulnerable to saturation. Therefore, a compromise has to be reached between noise rejection and retaining useful data, even if some information is lost in the process. To tackle this problem, we use a larger number of harmonics (n_{filt}) than the number used in the formulation of the excitation trajectory, for filtering the measured data. The algorithm to determine the number of harmonics that would best model the executed trajectory while keeping out noise is described in section 4.4.

Estimating Fourier Coefficients

With the selected number of harmonics n, truncated Fourier Series are fitted to the averaged and filtered measured data. The vector of Fourier series coefficients' estimates, $\hat{\boldsymbol{\delta}}$ is found as a result of the following minimization problem:

$$\underset{\hat{\boldsymbol{\delta}}}{\text{minimize}} \quad \sum_{\eta=0}^{\mathbf{N}} \|X_{measured}(\eta) - \hat{X}(\eta)\|^2$$

Applying the result that the truncated Fourier Transform of a signal is equaivalent to finding its the least squares Fourier fit [135], the Discrete Fourier Transform is used to find the Fourier coefficients.

Finding parameter estimates

From the Fourier series fit, the values of $\dot{\hat{X}}_i$ and $\ddot{\hat{X}}_i$ are found by analytical differentiation. The matrix of regressors, $\hat{\mathbf{W}}$, can therefore be populated. Finally, the linear least squares solution, eq.(4.12), is used to obtain the parameter estimates, $\hat{\boldsymbol{\pi}}$

4.4 Selecting the number of harmonics

An ideal number of harmonics, n_{filt}, should filter out high-frequency noise from the measured data, and provide the least estimation error. Thus, the following method is used:

1. The number of harmonics are increased incrementally from n_{ref} (the number of harmonics present in the reference trajectory) to n_{end}, a chosen value such that $n_{end} > n_{ref}$. At each value of n, the data-filtering steps are repeated, and the value of the harmonic selection criteria (discussed below) is calculated.
2. The value of n at which the least residual is obtained is then selected as n_{filt}.

Two criteria to determine the low pass cut-off harmonic based on the least squares residual approach and energy-balance are presented.

- **The least squares residual approach**
  The residual error, $\mathbf{e} \in \mathbb{R}^6$, of the least squares problem from (4.6) can be defined as

  $$\mathbf{e} = \tilde{\mathbf{b}} - \mathbf{W}\hat{\boldsymbol{\pi}} \tag{4.16}$$

  For the most accurate set of parameter estimates, the torques estimated using (4.16) are expected to be closer to the measured values. Therefore, the number of harmonics which gives the least residual error is selected as n_{filt}.

- **The energy balance approach**
  An alternative method to find the number of harmonics that relies on the balance between the kinetic energy and input power was published in our work [57]. Assuming that no dissipation has occured, energy conservation dictates that the input power should be equal to the rate of kinetic energy of the system. The total kinetic energy of the system is given as

  $$T = \frac{1}{2} m \boldsymbol{v}^T \boldsymbol{v} + \frac{1}{2} \boldsymbol{\omega}^T J_C \boldsymbol{\omega} \tag{4.17}$$

  where J_C represents the inertia about the center of mass of the system. The kinetic energy rate is obtained by differentiating (4.17), to give:

  $$\dot{T} = m \dot{\boldsymbol{v}}^T \dot{\boldsymbol{v}} + \boldsymbol{\omega}^T \left(J_C \dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times J_C \boldsymbol{\omega} \right) \tag{4.18}$$

Writing the input power (scalar product of input forces and torques with velocity) in the body frame, and equating it with (4.18) gives

$$\boldsymbol{f}^T\dot{\boldsymbol{v}} + \boldsymbol{\tau}^T\boldsymbol{\omega} = m\dot{\boldsymbol{v}}^T\dot{\boldsymbol{v}} + \boldsymbol{\omega}^T\left(J_C\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times J_C\boldsymbol{\omega}\right) \tag{4.19}$$

Therefore, the components on the right give the input power, P, whereas those on the left give the rate of kinetic energy $\dot{T}$. P makes use of the input forces and torques, and measured velocities, while the calculation of $\dot{T}$ needs estimates of the inertial parameters and the measured velocities and accelerations. The most appropriate n will give the least residual between P and $\dot{T}$.

A caveat must be added here - if the robot faces input saturation while executing the reference trajectory, the effect of high frequency harmonics will be more pronounced in the obtained trajectory. So, intuitively, we can expect the method of filtering and selection of harmonics to yield less accurate estimates when a high amount of saturation is encountered while tracking the reference, as separating the high-frequency response from the high-frequency measurement noise will prove to be a challenge in this case.

4.5 Simulation results

We assume that the robot has grasped an object, and the combined system is modeled as a rigid body. Without the loss of generality, the simulation is modeled on the inertial properties of the Space CoBot free-flyer [136]. The robot mass is $6.047\ kg$. Its inertia tensor is taken to be diagonal, with components $J_{xx} = 0.0453$, $J_{yy} = 0.0417$ and $J_{zz} = 0.0519\ kg$-m^2. Ten excitation trajectories were generated, five each from optimization of criteria V_1 and V_2. These trajectories were obtained by starting the optimization from different points. The period $T_f = 10$ sec, and $n_{ref} = 3$. The robot was made to track the excitation for 10 cycles (i.e., 100 seconds) .The measurement noise present in the position and attitude measurements is taken to be zero mean Gaussian Noise with a standard deviation of $0.05\ m$ for the position and $0.03\ rad$ for the attitude measurements.

Further, the inertial parameters of the combined system after grasping of payloads were selected from the following normal distributions: A mean (μ) of $7.2\ kg$ for mass, with standard deviation (σ) of 0.5 kg. A μ of $0.0\ m$ for the centre of mass offset, with σ = $0.04\ m$. For the off-diagonal values of the inertia tensor, μ = 0.02 and σ = $0.05\ kg$-m^2 and for the diagonal values, a mean value of $0.07\ kg$-m^2, with σ = 0.015. Ten combinations of these values were used.

Estimate accuracy was compared separately for the mass, CoM offset and the inertia tensor elements. As a measure of the amount of error between the parameter estimates and the real values, the norm of the errors was used. For instance, the CoM offset error is given by

$$p_{off,error} = \sqrt{\sum_{i=1}^{3}(p_{off,i} - \hat{p}_{off,i})^2} \tag{4.20}$$

The norm inertia error was found in a similar manner, whereas for mass the absolute error was used.

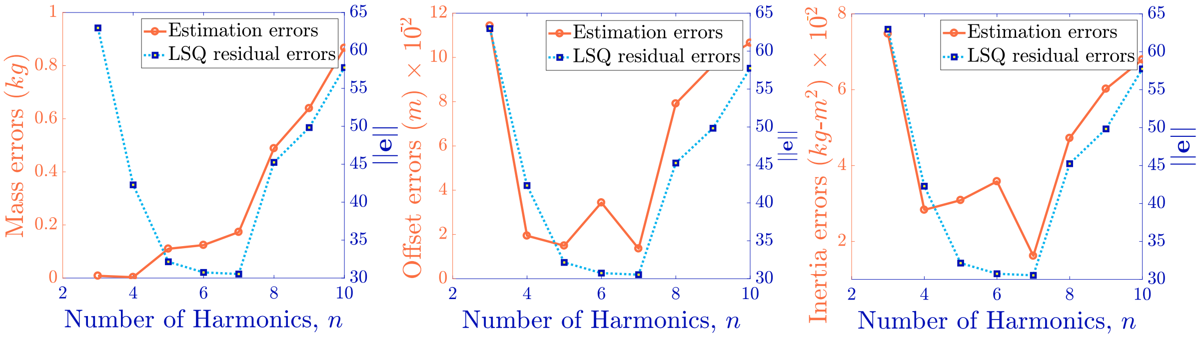


Figure 4.4: Evolution of the mass, inertia and CoM offset estimation errors, as well as the least square residual error, $||\mathbf{e}||$, as the number of harmonics retained during noise filtering are increased

4.5.1 Validating the method of harmonics selection

As mentioned previously, n_{filt} is the number of harmonics that was retained in the measured data at the time of noise-filtering. The value of n_{filt} was selected on the basis of the least residual error, $||\mathbf{e}||$, among all tested harmonics. For the simulations, $n_{ref} = 3$, and $n_{end} = 10$, i.e, the algorithm for harmonics selection tested all harmonics from 3 to 10. Results of the test for one unmodeled load are presented in fig.4.4. The following two lines are plotted as the number of harmonics is increased:

1. Evolution of the error between the parameter estimates and their actual values
2. Change in $||\mathbf{e}||$, which the harmonics selection algorithm uses to find n_{filt} based on the least squares residual

Let the value of n at which the least mass, inertia and CoM offset errors are obtained be $n_{min,m}, n_{min,i}$ and $n_{min,o}$, respectively. For the illustrated case, the estimation errors of inertia and CoM offset start to decrease and then increase, with $n_{min,i} = 4$ and $n_{min,o} = 5$. The increase can be attributed to the fact that as more number of harmonics is retained, the efficiency of filtering out high-frequency noise decreases. However, the trend of mass estimation error is different, and the least mass error is obtained at n_{ref}, with $n_{min,i} = 3$. The residual error $||\mathbf{e}||$ used for determining n_{filt} follows a trend similar to the estimation errors, with $n_{filt} = 7$. It is important to mention here that, while the least error for all parameters might not be obtained with the same n_{min}, only one value of n_{filt} can be chosen. It is up to the harmonics selection algorithm to ensure that this error remains low for all parameters, even when $n_{filt} \neq n_{min}$.

Fig. 4.5 presents the estimation errors obtained with 10 unmodeled loads when they are excited with 10 reference trajectories each. The value of n_{filt} is shown on the x-axis, and its absolute difference from $n_{min,m}, n_{min,i}$ and $n_{min,o}$ is illustrated using a heatmap. Though a handful of samples exhibit low accuracy, the figure shows that even when $|n_{filt} - n_{min,m/i/off}| \geq 4$, it does not results in higher errors for all three parameters in general, any more than when $n_{filt} = n_{min,m/i/off}$. This validates the results of the harmonics selection algorithm, in particular that if n_{filt} according the algorithm is larger thatn n_min, it does not cause high estimation errors.

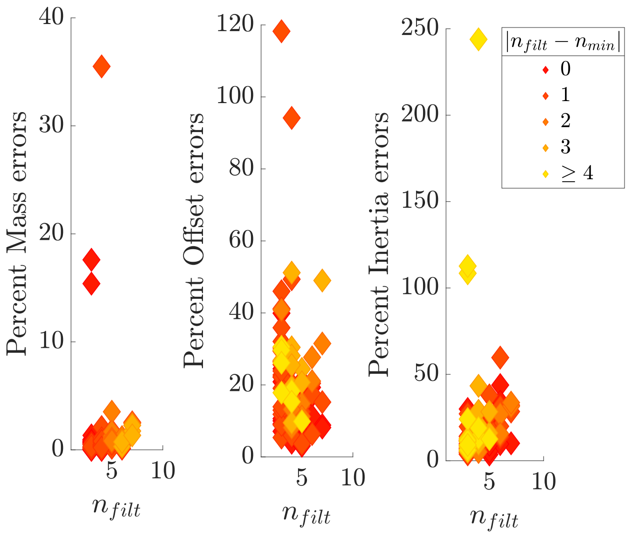


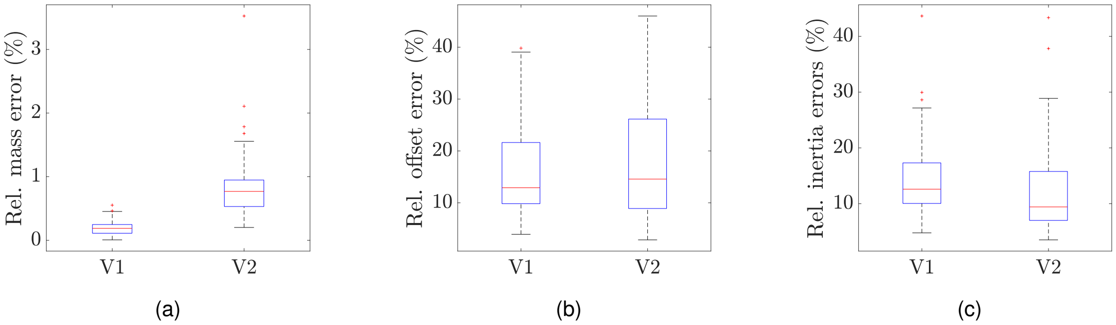
Figure 4.5: Parameter estimation results and harmonics needed for 10 loads, each estimated using 10 exciting trajectories

Recall that these samples were obtained after using either V_1 or V_2 criteria in generating the excitation trajectories. The large estimation errors could be attributed to one of the criteria leading to insufficient excitation in the parameters. Varying amounts of input saturation faced by the robot in executing the excitation trajectory could also lead to loss of accuracy in the estimates. The next sub-sections will explore these potential causes and attempt to find solutions for improving the accuracy.

4.5.2 Comparison of the excitation criteria

As input saturation at the time of tracking the excitation trajectory could be a potential cause of loss of accuracy, we eliminate the bias caused by saturation while investigating the excitation criteria. Therefore, the input saturation limits are temporarily deactivated for this study. Fig. 4.6 presents a box plot of the errors between the inertial parameter estimates obtained after optimizing either V_1 or V_2 to generate excitation, and following steps detailed in sections 4.3 and 4.4. Five excitation trajectories from each criteria were used for ten unmodeled loads each. Therefore, 50 results were generated in this manner,and figs. 4.6(a), 4.6(c) and 4.6(b) were plotted. Fig. 4.6(a) clearly points to the fact that the mass estimation errors are smaller with criteria V_1.

A paired t-test was conducted to test the alternate hypothesis that the mean of mass estimation errors obtained with criteria V_1 are higher than those obtained by criteria V_2. The same test was also carried out for CoM offset errors. Statistical significance was obtained in both cases (Mass errors test: t = -8.5162, p-value = 1.568e-11 and mean of the differences = -0.0500 kg, and CoM offset errors test: t = -1.7166, p-value = 0.0462 and mean of the differences = -0.0011 m). This means that based on the sample data, the average error of mass and center of mass, obtained using V_1 will be lower than that with V_2, at least 95% of the times. Even though the median of inertia errors with V_2 seems to be lower than V_1, it results in some outliers with high errors. Based on these results, we select criteria V_1 over V_2. The results reported from this point on only use data with excitation obtained through V_1.

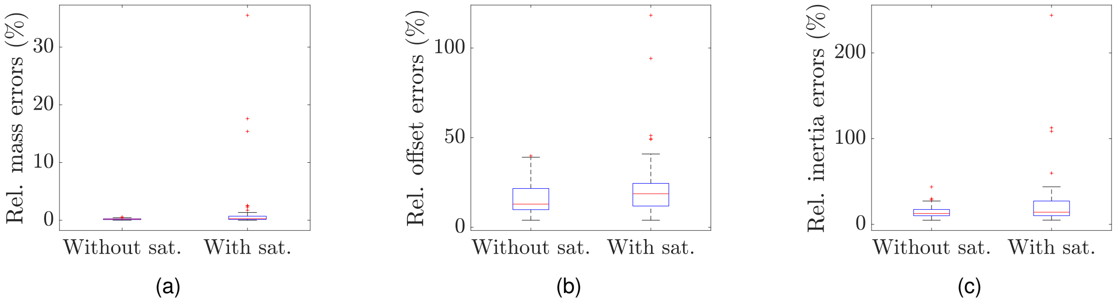


(a)

(b)

(c)

Figure 4.6: Relative estimation errors, obtained after excitation was provided with one of the two criteria, V_1 and V_2. Five excitation trajectories were generated per criteria, and each were used for the parameter estimation with 10 cases of normally distributed unmodeled loads. Saturation limits were turned off



(a)

(b)

(c)

Figure 4.7: Relative estimation errors obtained after excitation was provided with criteria V_1, with and without accounting for saturation limits, respectively. Five excitation trajectories were used for the parameter estimation with 10 cases of unmodeled loads with normally distributed paramters

4.5.3 The effect of saturation on accuracy of estimates

In a real system, the robot is likely to face saturation while tracking the excitating trajectory. Fig. 4.7 illustrates the result of saturation on the accuracy of the estimated parameters. Only those excitation trajectories that were obtained after optimizing V_1 were used. While it is clear that saturation results in more outliers in the estimated data, the resulting errors also exhibit a larger magnitude; for instance more than 200% in the case of inertias. Further investigation showed that these outliers correspond to cases where a high magnitude of saturation was encountered (one or more outputs were at their limits for $>$47.75% of the time) per cycle of the excitation trajectory.

Fig. 4.8 presents the effect of increased saturation while tracking the reference trajectories. The normalized estimation errors are plotted in increasing order of input saturation. The vector of errors is normalized separately for each of the three parameters, using the following equation:

$$d_p = \frac{d_p - min(\boldsymbol{d}_p)}{max(\boldsymbol{d}_p) - min(\boldsymbol{d}_p)} \tag{4.21}$$

Here, $\boldsymbol{d}$ stands for the vector of estimation errors. The type of errors (mass, norm of inertia and norm of CoM errors) are represented by p. Percent saturation is defined as the percentage of time for which at least one of the inputs of the robot was saturated through the course of tracking the excitation trajectory.

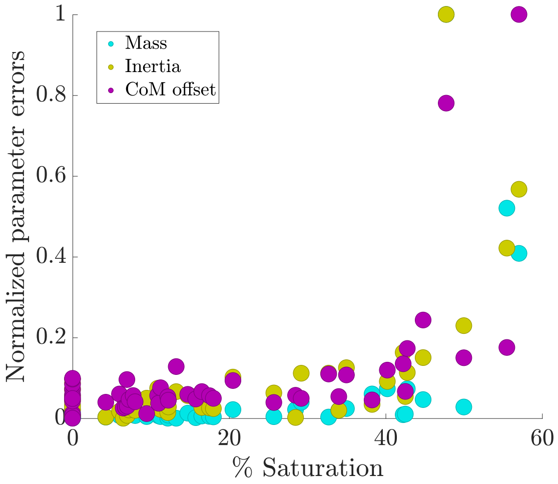


Figure 4.8: Change in the accuracy of estimates as input saturation while tracking the excitation increases. The x-axis shows the percent of time that at least one input of the robot was saturated while tracking the reference.

It is evident that, overall, an increase input saturation results in loss of estimation accuracy.

4.5.4 Improving the accuracy of the estimates

Section 4.5.3 showed that the harmonics-based noise rejection method falters at high levels of input saturation. This is attributed to an increase in the number and magnitude of high frequency harmonics in the executed trajectory. However, even without the presence of saturation, the estimation errors for CoM offset and inertia parameters in fig. 4.7 appear to be larger than mass errors. Note that the executed trajectory is expected to deviate from the reference due to the presence of system nonlinearities. This effect would compound for fast-paced trajectories which could be challenging for the robot to keep up with, resulting in more of these deviations. Consequently, each cycle of the obtained trajectory will not be exactly periodic, and spectral leakage effects will occur. As the executed trajectory is averaged over all the cycles, best results of this algorithm will be obtained when the executed trajectory follows the reference closely.

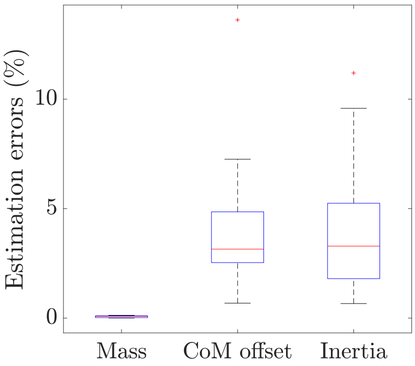


Figure 4.9: Relative percent estimation errors after using longer excitation trajectories with a period $T_f = 60$ secs and tighter acceleration limits.

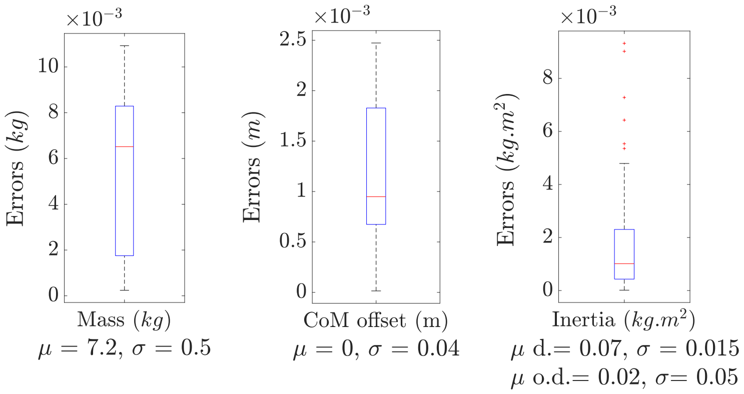


Figure 4.10: Absolute values of the relative percent estimation errors plotted in fig. 4.9. The mean and standard deviation values of the simulated data sets are given for perspective. ‘d.’ stands for diagonal, and ‘o.d’ for off-diagonal

Consdering the effect of trajectory pacing on the esitmates, results with slower excitation trajectories on a limited-actuation system were studied. Two trajectories were generated with periods T_f of 60 seconds each. They were used for estimating parameters on the set of 10 simulated loads as specified in the experimental setup. Fig. 4.9 shows the resulting relative percent errors, and fig. 4.10 plots their absolute values, for a perspective on the magnitude of these errors. Clearly, the estimation errors have now decreased. Thus, even though a longer amount of time is needed for the gathering data, using slower excitation trajectories is shown to improve estimation accuracy. Moreover, due to low-acceleration, the system is further away from reaching saturation limits than before. Therefore, determining the value of the excitation period is a trade-off between accuracy of the estimates and the time spent in obtaining them.

4.6 Discussion

The proposed method uses Fourier-basis excitation trajectories and tracks multiple cycles of it. This offers two benefits: first, from the filtering point of view, the cycles can be averaged over time to improve signal-to-noise ratio of the data and high frequency harmonics can be zeroed in order to filter out high-frequency measurement noise. Second, the Fourier series fit of filtered pose data can be differentiated analytically, thus eliminating the need for numerical differentiation during estimation. This approach was shown to work in simulation for cases with and without noise.

A potential issue arising from the application of this algorithm to a real robotic system is that even though the commanded trajectory is cyclic, this cannot be ensured for the executed trajectory. The algorithm proposes to leave out data from the first cycle in order to discard transient data. However, it cannot be ensured that the subsequent cycles will be exactly periodic. Additionally, the presence of input saturation was shown to have a detrimental effect on the accuracy of the estimates. Dynamic systems are prone to saturation, and the resulting high-frequency response makes it challenging to filter out noise from the meaningful data using this approach. The commanded trajectory can be slowed down in pace, i.e., stretched out in time by reducing acceleration, so that the saturation is avoided by keeping robot motion within the acceleration limits. On the other hand, the proposed method will not filter out low-frequency noise that persists after time-domain averaging of the tracked trajectory.

Chapter 5

Recursive inertial estimation using discrete variational mechanics

Instead of discretizing the continuous equations of motion derived from Lagrangian mechanics, discrete variational mechanics equations are derived from the discrete Lagrangian. The discrete equations of motion therefore describe the evolution of a dynamical system sympletically, respecting energy dissipation and conserving momentum. In contrast to the continuous equations of motion where the configuration space is composed of the states and their derivatives, $(q, \dot{q})$, the discrete variational equations of motion operate in the discrete configuration space of (q_k, q_{k+1}) (section 2.6). As the equations are a function of the generalized coordinates, the need for differentiating pose data is eliminated.

A novel parameter identification approach presented in [103] used discrete variational equations of the rotational dynamics to formulate a recursive inertia estimation problem. In this formulation, Euler angles were used to characterze the rotation. As the discrete dynamics can be expressed linearly with respect to the inertias, a linear identification problem can be formulated. Building upon this work, a full rigid body online inertial parameter estimation problem using discrete variational mechanics is proposed in this chapter. Lie group variational integrators based on unit dual quaternions [137] are used for this purpose. Dual quaternions are chosen here as they are unambiguous and compact representations of rigid-body pose transformation. Owing to the dual quaternion formulation, the position and orientation components do not need to be decoupled, and the efficiency and elegance of quaternions is maintained. The estimation problem is posed as a recursive Semi-Definite program, with Linear Matrix Inequality (LMI) constraints for physical consistency [138] of the estimates. Parameter estimates obtained from this method are shown to be at least one order of magnitude more accurate than those obtained using discretization methods like finite difference. To the best of our knowledge, this is the first time that discrete equations based on the principles of variational mechanics have been used for full rigid body inertial parameter estimation. This work was published in [60].

This method can be seen as an alternative to the approach proposed in the previous chapter for the free-flyer payload transportation scenario. While the first approach leverages the unique properties of the Fourier series by performing filtering and analytical differentiation, the method proposed in this

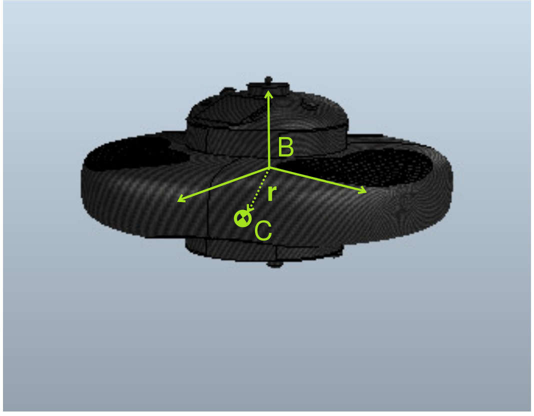


Figure 5.1: Illustration of the Space CoBot free-flyer showing origin of the body frame (B), CoM (C), and offset $(\boldsymbol{r})$.

chapter employs discrete variational mechanics to formulate the estimation problem using only robot states, thus eliminating the need for acceleration measurements. Further, the second approach can be transformed into an online and recursive problem, which is not feasible for the first approach as the data processing occurs after multiple cycles of the excitation trajectory have been tracked.

5.1 Discrete variational equations of motion using Dual Quaternions

5.1.1 Dual Quaternions

Dual quaternions (DQ) are singularity free, unambiguous, as well as the most compact $SE(3)$ rigid transform representations [139]. First proposed by Clifford [140], a dual quaternion is composed of a real and a dual part, each of which is a quaternion (quaternions are examined in A.1). A DQ $\tilde{\boldsymbol{q}}$ is defined as

$$\tilde{\boldsymbol{q}} = \bar{\boldsymbol{q}}_a + \epsilon \bar{\boldsymbol{q}}_b \tag{5.1}$$

where ϵ is a dual unit, $\epsilon^2 = 0$, and $\bar{\boldsymbol{q}}_a$ and $\bar{\boldsymbol{q}}_b$ are ordinary quaternions. Some key properties of dual quaternions are

- The conjugate of DQs is the conjugate of the constituent quaternions

$$\tilde{\boldsymbol{q}}^{\dagger} = \bar{\boldsymbol{q}}_a^{\dagger} + \epsilon \bar{\boldsymbol{q}}_b^{\dagger} \tag{5.2}$$

- DQ multiplication, denoted here by $\circ$ is given by

$$\tilde{\boldsymbol{q}}_1 \otimes \tilde{\boldsymbol{q}}_2 = \bar{\boldsymbol{q}}_{1a} \circ \bar{\boldsymbol{q}}_{1a} + \epsilon \left(\bar{\boldsymbol{q}}_{1a} \circ \bar{\boldsymbol{q}}_{2b} + \bar{\boldsymbol{q}}_{2a} \circ \bar{\boldsymbol{q}}_{1b} \right) \tag{5.3}$$

  where the operator $\circ$ represents quaternion multiplication. Consequently, the magnitude of a DQ is $\tilde{\boldsymbol{q}} \otimes \tilde{\boldsymbol{q}}^{\dagger}$

- A unit dual quaternion is defined as $\tilde{\boldsymbol{q}} \otimes \tilde{\boldsymbol{q}}^{\dagger} = \bar{\boldsymbol{I}} + \epsilon\bar{\mathbf{0}}$ which implies that $\bar{\boldsymbol{q}}_{1a}^{\dagger} \circ \bar{\boldsymbol{q}}_{1b} + \bar{\boldsymbol{q}}_{1b} \circ \bar{\boldsymbol{q}}_{1a}^{\dagger} = 0$

Unit dual quaternions can be used for representing rigid body poses (rotation and translation) in $SE(3)$. Let $\boldsymbol{l}$ be the position of a rigid body and $\bar{\boldsymbol{q}}$ represent its attitude in unit quaternions ($\boldsymbol{l} \in \mathbb{R}^3$, $\bar{\boldsymbol{q}} \in \mathbb{R}^4$ and $||\bar{\boldsymbol{q}}|| = 1$). Then $\tilde{\boldsymbol{p}}$, a unit DQ used to represent rigid body pose, can be written as [137]

$$\tilde{\boldsymbol{p}} = \bar{\boldsymbol{q}} + \epsilon\frac{1}{2}\hat{\boldsymbol{l}} \circ \bar{\boldsymbol{q}} \tag{5.4}$$

where $\hat{\boldsymbol{l}} \in \mathbb{R}^4$ signifies a pure quaternion. Using the scalar last representation,

$$\hat{\boldsymbol{l}} = \begin{bmatrix} \boldsymbol{l} \\ 0 \end{bmatrix} \tag{5.5}$$

A pose $\tilde{\boldsymbol{p}}$ can be used to transform a point vector $\mathbf{r} \in \mathbb{R}^3$ in $SE(3)$ as [137]

$$\tilde{\mathbf{r}}' = \tilde{\boldsymbol{p}} \otimes \tilde{\mathbf{r}} \tag{5.6}$$

where $\tilde{\mathbf{r}} = \bar{\boldsymbol{I}} + \epsilon\frac{1}{2}\hat{\mathbf{r}}$, $\hat{\mathbf{r}} = [\mathbf{r}^T 0]^T$. The transformed point vector can be retrieved as $\hat{\mathbf{r}}' = 2\bar{\mathbf{r}}_b \circ \bar{\mathbf{r}}_a^{\dagger}$

5.1.2 Dual Quaternion Variational Integrators

To arrive at the discrete variational equations of motion using the dual quaternion formulation, the standard approach involving the variation of the discrete action sum (outlined in section 2.6) is followed. The derivation of dual quaternion variational integrators (DQVI) detailed in [137], is summarized here.

The attitude in quaternions is related to the angular velocity of a rigid body as $\hat{\boldsymbol{\omega}}_B = 2\bar{\boldsymbol{q}}^{\dagger} \circ \dot{\bar{\boldsymbol{q}}}$. A similar relationship exists between the pure dual quaternion $\tilde{\boldsymbol{\chi}}_B$ of the velocities expressed in the body frame, and the pose dual quaternion $\tilde{\boldsymbol{p}}$.

$$\tilde{\boldsymbol{\chi}}_B = \begin{bmatrix} \hat{\boldsymbol{\omega}}_B \\ \hat{\boldsymbol{v}}_B \end{bmatrix} \tag{5.7}$$

$$\tilde{\boldsymbol{\chi}}_B = 2\tilde{\boldsymbol{p}}^{\ddagger} \otimes \dot{\tilde{\boldsymbol{p}}} \tag{5.8}$$

where the symbol $\ddagger$ denotes the dual quaternion conjugate. The Lagrangian - which, in the case without potential energy is equal to the kinetic energy - is written in the dual quaternion form as

$$L = \frac{1}{2}\left(\tilde{\boldsymbol{\chi}}_B^T \mathbf{M}_{8\times 8} \tilde{\boldsymbol{\chi}}_B\right) \tag{5.9}$$

recall that $\mathbf{M}$ represents the spatial inertia matrix (2.8),

$$\mathbf{M} = \begin{bmatrix} J & mS(\boldsymbol{c}) \\ mS^T(\boldsymbol{c}) & m\mathbf{I}_{3\times 3} \end{bmatrix} \tag{5.10}$$

where m denotes the mass, $c \in \mathbb{R}^3$ represents the center of mass offset, and J represents the robot inertia about the origin of the body frame. The spatial inertia $\mathbf{M}$ is coonverted to an 8×8 matrix by padding it with zeros

$$\mathbf{M}_{8\times 8} = \begin{bmatrix} 0 & \cdots & 0 & \cdots \\ \vdots & J & \vdots & mS(\mathbf{c}) \\ 0 & \cdots & 0 & \cdots \\ \vdots & mS^T(\mathbf{c}) & \vdots & m\mathbf{I}_{3\times 3} \end{bmatrix} \tag{5.11}$$

To find the discrete Lagrangian, $\dot{\tilde{\boldsymbol{p}}}_k$ can be approximated by using the trapezoidal rule over a fixed time step $h \approx t_{k+1} - t_k$ as:

$$\dot{\tilde{\boldsymbol{p}}}_k = \left(\tilde{\boldsymbol{p}}_{k+1} - \tilde{\boldsymbol{p}}_k\right)/h \tag{5.12}$$

Defining a quantity $\tilde{\boldsymbol{f}}_k$ as change in the dual quaternions between each time step,

$$\tilde{\boldsymbol{f}}_k = \tilde{\boldsymbol{p}}_k^{\dagger} \otimes \tilde{\boldsymbol{p}}_{k+1} \tag{5.13}$$

yields the discrete Lagrangian as

$$L_{dk} = \frac{2}{h}\left(\tilde{\boldsymbol{f}}_k^T \mathbf{M}_{8\times 8} \tilde{\boldsymbol{f}}\right) \tag{5.14}$$

External forces and torques are incorporated in the variational formulation by using a discrete approximation of the virtual work integral and applying Lagrange-D'Alembert's principle. The forces and torques are written in their pure dual quaternion form as $\tilde{\boldsymbol{\tau}}_{Bk} = \hat{\boldsymbol{\tau}}_{ak} + \epsilon\hat{\boldsymbol{\tau}}_{bk}$. Here $\boldsymbol{\tau}_{ak}$ are the applied torques and $\boldsymbol{\tau}_{bk}$ are the forces expressed in body frame.

Since dual-quaternions are homomorphic[1] to the $SE(3)$ group, six independent parameters can be used to represent $\tilde{\boldsymbol{f}}_k$. Therefore, $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi} \in \mathbb{R}^3$ are used as independent and unconstrained vector representations for the unit dual quaternion $\tilde{\boldsymbol{f}}_k = \overline{\boldsymbol{f}}_a + \epsilon\overline{\boldsymbol{f}}_b$ as in [104, 137]. Here, $\boldsymbol{\Phi}^2$ represents $\boldsymbol{\Phi}^T\boldsymbol{\Phi}$, or the L_1 vector norm, $|\boldsymbol{\Phi}|$, while the . operator represents vector inner product, $\boldsymbol{\Psi}\cdot\boldsymbol{\Phi} = \boldsymbol{\Psi}^T\boldsymbol{\Phi}$. From (5.15), it is evident that $|\boldsymbol{\Phi}| < 1$ for the parameters to be valid.

$$\overline{\boldsymbol{f}}_a = \begin{bmatrix} \sqrt{1-\boldsymbol{\Phi}^2} \\ \boldsymbol{\Phi} \end{bmatrix}, \quad \overline{\boldsymbol{f}}_b = \begin{bmatrix} -\frac{\boldsymbol{\Psi}\cdot\boldsymbol{\Phi}}{\sqrt{1-\boldsymbol{\Phi}^2}} \\ \boldsymbol{\Psi} \end{bmatrix} \tag{5.15}$$

Starting from (5.14) and taking variation of the discrete action sum, the discrete rigid body dynamic equations in dual quaternion form are [137],

$$\begin{cases} \mathcal{A}\left(\boldsymbol{\Phi}_k, \boldsymbol{\Psi}_k\right) - \boldsymbol{\alpha}_k = \mathbf{0} \\ \boldsymbol{B}\left(\boldsymbol{\Phi}_k, \boldsymbol{\Psi}_k\right) - \boldsymbol{\beta}_k = \mathbf{0} \end{cases} \tag{5.16}$$

---

[1] structure preserving

where

$$\mathcal{A}\left(\boldsymbol{\Phi}_k, \boldsymbol{\Psi}_k\right) = \left(-\frac{\boldsymbol{\Psi}_k \cdot \boldsymbol{\Phi}_k}{\sqrt{1-\boldsymbol{\Phi}_k^2}} I + S\left(\boldsymbol{\Psi}_k\right)\right)\left(\mathbf{M}_{21}\boldsymbol{\Phi}_k + \mathbf{M}_{22}\boldsymbol{\Psi}_k\right) + \left(\sqrt{1-\boldsymbol{\Phi}_k^2} I + S\left(\boldsymbol{\Phi}_k\right)\right)\left(\mathbf{M}_{11}\boldsymbol{\Phi}_k + \mathbf{M}_{12}\boldsymbol{\Psi}_k\right) \tag{5.17}$$

$$\begin{aligned}\boldsymbol{\alpha}_k = &\left(\sqrt{1-\boldsymbol{\Phi}_{k-1}^2} I - S\left(\boldsymbol{\Phi}_{k-1}\right)\right)\left(\mathbf{M}_{11}\boldsymbol{\Phi}_{k-1} + \mathbf{M}_{12}\boldsymbol{\Psi}_{k-1}\right) - \\ &\left(\frac{\boldsymbol{\Psi}_{k-1} \cdot \boldsymbol{\Phi}_{k-1}}{\sqrt{1-\boldsymbol{\Phi}_{k-1}^2}} I + S\left(\boldsymbol{\Psi}_{k-1}\right)\right)\left(\mathbf{M}_{21}\boldsymbol{\Phi}_{k-1} + \mathbf{M}_{22}\boldsymbol{\Psi}_{k-1}\right) + \frac{h^2}{2}\boldsymbol{\tau}_{ak}\end{aligned} \tag{5.18}$$

$$\boldsymbol{B}\left(\boldsymbol{\Phi}_k, \boldsymbol{\Psi}_k\right) = \left(\sqrt{1-\boldsymbol{\Phi}_k^2} I + S\left(\boldsymbol{\Phi}_k\right)\right)\left(\mathbf{M}_{21}\boldsymbol{\Phi}_k + \mathbf{M}_{22}\boldsymbol{\Psi}_k\right) \tag{5.19}$$

$$\boldsymbol{\beta}_k = \left(\sqrt{1-\boldsymbol{\Phi}_{k-1}^2} I - S\left(\boldsymbol{\Phi}_{k-1}\right)\right)\left(\mathbf{M}_{21}\boldsymbol{\Phi}_{k-1} + \mathbf{M}_{22}\boldsymbol{\Psi}_{k-1}\right) + \frac{h^2}{2}\boldsymbol{\tau}_{bk} \tag{5.20}$$

The notation $\mathbf{M}_{ij}$ for i and j = 1, 2 used in (5.17) - (5.20) represents the 3×3 matrices that form $\mathbf{M}$. For instance, $\mathbf{M}_{11} = J$, $\mathbf{M}_{12} = mS(\boldsymbol{c})$ and so on. $S()$ represents the vector written in skew-symmetric matrix form.

5.2 Formulation of the estimation problem

An estimation problem linear in the intertial parameters of interest can be formed using the dynamics given in (5.17) - (5.20). For this, we define a vector of inertial parameters $\mathbf{J}$ and operator G

$$\begin{aligned}\mathbf{J} &= \begin{bmatrix} J_{11} & J_{12} & J_{13} & J_{22} & J_{23} & J_{33} \end{bmatrix}^T \\ G(\boldsymbol{\Phi}) &= \begin{bmatrix} \Phi_1 & \Phi_2 & \Phi_3 & 0 & 0 & 0 \\ 0 & \Phi_1 & 0 & \Phi_2 & \Phi_3 & 0 \\ 0 & 0 & \Phi_1 & 0 & \Phi_2 & \Phi_3 \end{bmatrix}\end{aligned} \tag{5.21}$$

Using (5.10) and (5.21) we can write:

$$\begin{aligned}\boldsymbol{M}_{11}\boldsymbol{\Phi} &= J\boldsymbol{\Phi} = G(\boldsymbol{\Phi})\,\mathbf{J} \\ \boldsymbol{M}_{12}\boldsymbol{\Phi} &= mS(\mathbf{c})\boldsymbol{\Phi} = -mS(\boldsymbol{\Phi})\,\mathbf{c} \\ \boldsymbol{M}_{21}\boldsymbol{\Phi} &= mS^T(\mathbf{c})\boldsymbol{\Phi} = mS(\boldsymbol{\Phi})\,\mathbf{c} \\ \boldsymbol{M}_{22}\boldsymbol{\Phi} &= m\mathbf{I}_{3x3}\boldsymbol{\Phi} = m\boldsymbol{\Phi}\end{aligned} \tag{5.22}$$

Further, let

$$
\begin{aligned}
P_k &= \left(-\frac{\boldsymbol{\Psi}_k \cdot \boldsymbol{\Phi}_k}{\sqrt{1-\boldsymbol{\Phi}_k^2}} I - S\left(\boldsymbol{\Psi}_k\right) \right) \\
Q_k &= \left(-\frac{\boldsymbol{\Psi}_k \cdot \boldsymbol{\Phi}_k}{\sqrt{1-\boldsymbol{\Phi}_k^2}} I + S\left(\boldsymbol{\Psi}_k\right) \right) \\
\mathcal{M}_k &= \left(\sqrt{1-\boldsymbol{\Phi}_k^2} I - S\left(\boldsymbol{\Phi}_k\right) \right) \\
N_k &= \left(\sqrt{1-\boldsymbol{\Phi}_k^2} I + S\left(\boldsymbol{\Phi}_k\right) \right)
\end{aligned}
\tag{5.23}
$$

using (5.23) and (5.22) in (5.17) - (5.20), and writing $S\left(\boldsymbol{\Phi}_k\right)$ as $\mathfrak{S}_k$ and $G(\boldsymbol{\Phi}_k)$ as G_k, we get

$$
\begin{aligned}
\mathcal{A}\left(\boldsymbol{\Phi}_k, \boldsymbol{\Psi}_k\right) &= m Q_k \left(\mathfrak{S}_k \, \mathbf{c} + \boldsymbol{\Psi}_k\right) + N_k \left(G_k \mathbf{J} - m \mathfrak{S}_k \, \boldsymbol{c}\right) \\
\boldsymbol{\alpha}_k &= \mathcal{M}_{k-1} \left(G_{k-1} \mathbf{J} - m \mathfrak{S}_{k-1} \, \boldsymbol{c}\right) + m P_{k-1} \left(\mathfrak{S}_{k-1} \, \mathbf{c} + \boldsymbol{\Psi}_{k-1}\right) + \frac{h^2}{2} \boldsymbol{\tau}_{ak} \\
\mathcal{B}\left(\boldsymbol{\Phi}_k, \boldsymbol{\Psi}_k\right) &= m N_k \left(\mathfrak{S}_k \, \mathbf{c} + \boldsymbol{\Psi}_k\right) \\
\boldsymbol{\beta}_k &= m \mathcal{M}_{k-1} \left(\mathfrak{S}_{k-1} \, \boldsymbol{c} + \boldsymbol{\Psi}_{k-1}\right) + \frac{h^2}{2} \boldsymbol{\tau}_{bk}
\end{aligned}
\tag{5.24}
$$

After substituting (5.24) in (5.16) and rearranging, we get the following set of equations:

$$
\begin{aligned}
(N_k G_k - \mathcal{M}_{k-1} G_{k-1}) \mathbf{J} + m (Q_k \boldsymbol{\Psi}_k - P_{k-1} \boldsymbol{\Psi}_{k-1}) &+ \\
\left(\left(Q_k - N_k\right) \mathfrak{S}_k + \left(\mathcal{M}_{k-1} - P_{k-1}\right) \mathfrak{S}_{k-1}\right) \cdot m \mathbf{c} &= \frac{h^2}{2} \boldsymbol{\tau}_{ak} \\
m \left(N_k \boldsymbol{\Psi}_k - \mathcal{M}_{k-1} \boldsymbol{\Psi}_{k-1}\right) + \left(N_k \mathfrak{S}_k - \mathcal{M}_{k-1} \mathfrak{S}_{k-1}\right) \cdot m \mathbf{c} &= \frac{h^2}{2} \boldsymbol{\tau}_{bk}
\end{aligned}
\tag{5.25}
$$

Writing this in a matrix form, we obtain,

$$
\begin{bmatrix} A_k & B_k & C_k \\ \mathbf{0}_{3\times 6} & D_k & E_k \end{bmatrix}
\begin{bmatrix} \mathbf{J} \\ m\mathbf{c} \\ m \end{bmatrix}
= \frac{h^2}{2} \begin{bmatrix} \boldsymbol{\tau}_{ak} \\ \boldsymbol{\tau}_{bk} \end{bmatrix}
\tag{5.26}
$$

where

$$
\begin{aligned}
A_k &= N_k G_k - \mathcal{M}_{k-1} G_{k-1} \\
B_k &= (Q_k - N_k) \mathfrak{S}_k + (\mathcal{M}_{k-1} - P_{k-1}) \mathfrak{S}_{k-1} \\
C_k &= Q_k \boldsymbol{\Psi_k} - P_{k-1} \boldsymbol{\Psi}_{k-1} \\
D_k &= N_k \mathfrak{S}_k - \mathcal{M}_{k-1} \mathfrak{S}_{k-1} \\
E_k &= N_k \boldsymbol{\Psi}_k - \mathcal{M}_{k-1} \boldsymbol{\Psi}_{k-1}
\end{aligned}
\tag{5.27}
$$

Equation (5.26) can be written compactly as

$$
H_k \boldsymbol{\pi} = T_k \tag{5.28}
$$

where

$$\begin{aligned} H_k &= \begin{bmatrix} A_k & B_k & C_k \\ 0_{3\times 6} & D_k & E_k \end{bmatrix} \\ \boldsymbol{\pi} &= \begin{bmatrix} \mathbf{J}^T & m\mathbf{c}^T & m \end{bmatrix}^T \\ T_k &= \frac{h^2}{2} \begin{bmatrix} \boldsymbol{\tau}_{ak} \\ \boldsymbol{\tau}_{bk} \end{bmatrix} \end{aligned} \tag{5.29}$$

5.3 The estimation algorithm

If noise $(\mathbf{v})$ on the measured torques $(\tilde{T})$ is assumed to be zero mean Gaussian and independent and identically distributed (iid) , then we consider the following observation model for ((5.26))

$$\tilde{T} = H\boldsymbol{\pi} + \mathbf{v} \tag{5.30}$$

Considering that noise on the torque measurements has the same standard deviation, the minimization problem can be written as

$$\underset{\pi}{\text{minimize}} \quad \frac{1}{2}||H_{\underline{N}}\boldsymbol{\pi} - T_{\underline{N}}||_2^2 \tag{5.31}$$

where the matrix subscript $\underline{N}$ denotes that the matrix is composed of all measurements from time step 1 to N.

$$H_{\underline{N}} = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_N \end{bmatrix} \qquad T_{\underline{N}} = \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_N \end{bmatrix}$$

To avoid a rapid increase in the dimension of the problem as more measurement data arrives, we use the QR decomposition (or Gram-Schmidt). The QR decomposition of a matrix in $\mathbb{R}^{m\times k}$ yields an orthonormal[2] matrix $Q \in \mathbb{R}^{m\times m}$ and an upper triangular matrix $R \in \mathbb{R}^{m\times k}$ [66]. For a matrix of dimension $m\times k$, where $m > k$, the economy QR decomposition (denoted here as $\textbf{QR}_{\text{econ}}$) produces Q of dimension $\in \mathbb{R}^{m\times k}$ and $R \in \mathbb{R}^{k\times k}$. Since this is an overdetermined problem, the economy QR decompostion is used to formulate the recursive problem. Replacing H by QR in the objective (5.31), and since Q is orthonormal, the objective becomes $R\boldsymbol{\pi} - z$, where $Q^T T = z$.

At every time step $n = 1, \cdots, N$, the matrices of the recursive estimation problem are updated as follows.

$$Q_{n+1}, R_{n+1} = \textbf{QR}_{\text{econ}}\left(\begin{bmatrix} R_n \\ H_{n+1} \end{bmatrix}\right) \tag{5.32}$$

$$z_{n+1} = Q_{n+1}^T \begin{bmatrix} z_n \\ T_{n+1} \end{bmatrix} \tag{5.33}$$

In this manner, the information from the previous time step is encoded in the problem at the subsequent

---

[2] A matrix Q of dimension n is orthogonal when $Q^T = Q^{-1}$, or $Q^T Q = I_n$. This matrix is orthonormal when its colums are unit vectors.

time steps. The objective at each step then reduces to

$$\underset{\pi}{\text{minimize}} \quad \frac{1}{2}||R_n\boldsymbol{\pi} - z_n||_2^2 \tag{5.34}$$

Note that the dimension of R and z remains the same for every step, yet the information from new measurements is taken into account.

5.3.1 Full physical consistency constraints

Recall from (2.6) that $\mathbf{J} = \mathbf{J}_C + mS(\boldsymbol{c})S^T(\boldsymbol{c})$, where $\mathbf{J}_C$ is the inertia about the CoM of the system, and $\mathbf{J}$ is the inertia about the origin of the body frame. Ensuring positive mass and positive definite inertia, $m > 0$ and $\mathbf{J}_C \succ 0$ yields physical semi-consistency, and is not adequate to ensure that the parameters belong to a density-relizable system [141]. The inertia J_C is fully density realizable if there exists $R \in SO(3)$ such that $\mathbf{J}_C = R\mathcal{J}R^T$, the diagonal matrix $\mathcal{J} = diag(\mathcal{J}_{11}, \mathcal{J}_{22}, \mathcal{J}_{33}) \succ 0$, and the triangle inequalities on $\mathcal{J}$ are satisfied [138]:

$$\mathcal{J}_{11} + \mathcal{J}_{22} \geq \mathcal{J}_{33}, \quad \mathcal{J}_{22} + \mathcal{J}_{33} \geq \mathcal{J}_{11}, \quad and \quad \mathcal{J}_{11} + \mathcal{J}_{33} \geq \mathcal{J}_{22} \tag{5.35}$$

Therefore, the 6D spatial inertia matrix $\mathbf{M}$ is fully physically consistent if $m > 0$ and the above constraints on J_C are satisfied, such that

$$\mathbf{M} = \begin{bmatrix} R\mathcal{J}R^T + mS(\boldsymbol{c})S^T(\boldsymbol{c}) & mS(\boldsymbol{c}) \\ mS^T(\boldsymbol{c}) & mI_{3\times3} \end{bmatrix} \tag{5.36}$$

Using the results from Theorem 3 presented in [138], the full physical consistency constraints can be reduced to positive definiteness of the pseudo inertia matrix, which we denote here by $\underline{\mathbf{M}}$,

$$\underline{\mathbf{M}} \succ 0, \qquad \underline{\mathbf{M}} = \begin{bmatrix} \Sigma & m\boldsymbol{r} \\ m\boldsymbol{r}^T & m \end{bmatrix} \tag{5.37}$$

where

$$\Sigma = \frac{1}{2}Tr(\mathbf{J}_C)\mathbf{I}_{3\times3} - \mathbf{J}_C \tag{5.38}$$

Enforcing the positive definiteness of $\underline{\mathbf{M}}$ ensures the application of the individual physical consistency constraints. This 4×4 Linear Matrix Inequality (LMI) constraint offers computational benefits when solving the estimation problem recursively.

5.3.2 Conversion to a semi-definite program

Since the objective function in (5.34) is linear with LMI constraints, the least squares problem is converted to a semi-definite program (SDP) (section 1.1 of [82]). Converting the recursive optimization from

(5.34) to a recursive SDP with the addition of a slack variable s (discussed in section 2.4.3), we get

$$\begin{aligned} \underset{\pi'}{\text{minimize}} \quad & \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\pi} \\ s \end{bmatrix} \\ \text{subject to} \quad & \begin{cases} \begin{bmatrix} s & (R_N\boldsymbol{\pi} - z_N)^T \\ (R_N\boldsymbol{\pi} - z_N) & \boldsymbol{I} \end{bmatrix} \succeq 0 \\ \underline{M} \succ 0 \end{cases} \end{aligned}$$

This problem is solved at every time step with updated values of R and z. The SDP has a linear objective, and the earlier objective of (5.34) is expressed as a convex constraint. The positive definiteness constraint on the 4×4 pseudo inertia matrix $\underline{M}$ replaces the individual physical consistency constraints, benefitting the recursive computation.

5.4 Implementation

The robot model was simulated with the following inertial parameters: $m = 5\ kg$, $J(\text{i.e.}, \mathbf{J})_{xx} = 1.5$, $J_{yy} = 2$, $J_{zz} = 3$, $J_{xy} = 0.1$, $J_{xz} = 0.2$, $J_{yz} = 0.3\ kgm^2$ and r_x, r_y and r_z as 0.05, 0.1 and $0.05\ m$ respectively. Fast integrator of the type Implicit Runge-Kutta Gauss Legendre 4 (IRK-GL4) exported by the ACADO toolkit [142] was used to simulate the 6 Degree of Freedom rigid body dynamics. Time steps of 0.05s were used. For the sake of simplicity, the forces and torques are sinusoidal waves with different phase and amplitudes. However, the trajectories can be optimized for information gain to ensure that the regressor (matrix $\underline{H_N}$ from (5.31)) is full rank and well-conditioned, as in [59].

We consider that the states consisting of position, attitude and linear and angular velocities are obtained through a standard pose determination algorithm. For comparison with the proposed method, a finite difference (FD) based discretization approach was used to approximate the angular and linear accelerations. Similar to the discrete variational case, the FD equations can be arranged linear to π (this formulation is presented in sec.4.1). The parameters are then estimated by solving the SDP with CVX [143] [144] in MATLAB. On a 4 core Intel i5-7200U CPU @ 2.50GHz running Ubuntu 16.04 LTS, CVX took on average 0.2s to solve each constrained least squares problem using SDP.

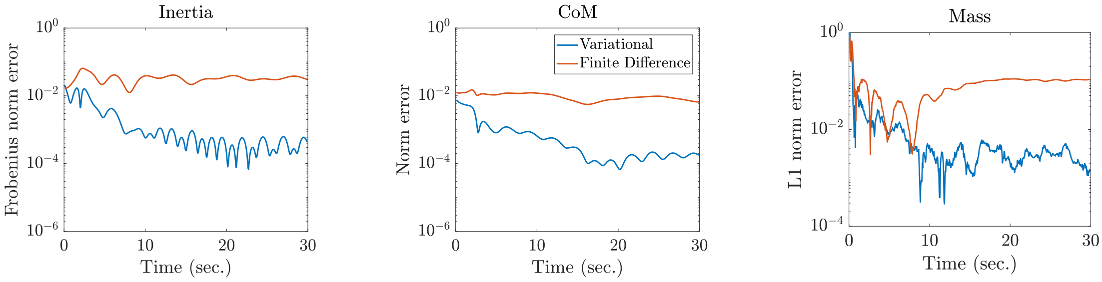


Figure 5.2: Relative estimation errors obtained through the two approaches. A logarithmic scale is used to express errors on the vertical axis.

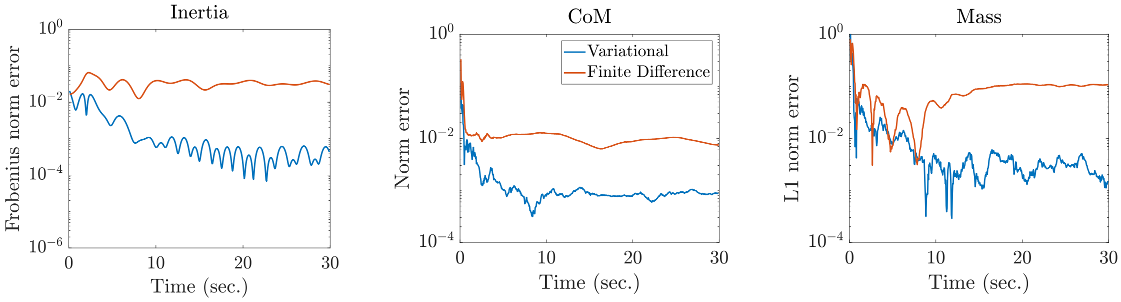


Figure 5.3: Relative estimation errors when torques and velocities were corrupted with zero mean Gaussian noise. Errors on the vertical axis are expressed using a logarithmic scale.

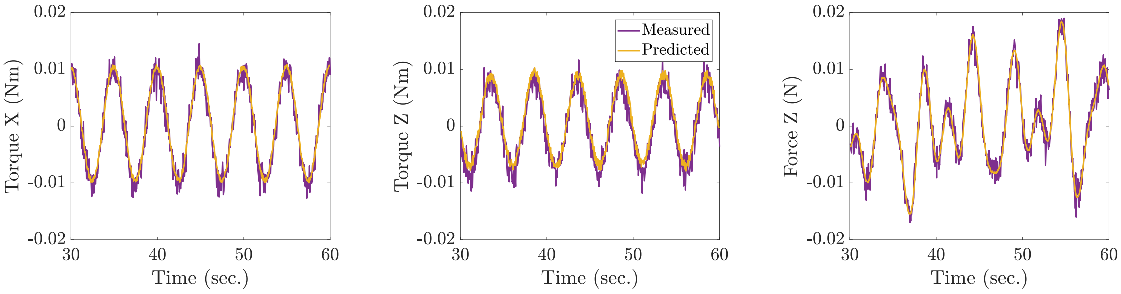


Figure 5.4: Regression errors for the next 30 seconds of simulatio following fig.5.3. Parameter estimates obtained at t = 30 were used.

5.5 Results

Fig.5.2 presents the results of parameter estimation with recursive least squares in the case of no added noise. The same data was used for estimation using the finite differences approach. The graph of estimation errors starts at similar points for both methods; however, the variational approach eventually converges to more accurate estimates. The variational approach yields at least an order of magnitude greater accuracy for each parameter than that obtained through the finite difference approach.

Further, the measured forces and torques were corrupted with zero mean Gaussian disturbances of spectral density $10^{-3}\ units/\sqrt{Hz}$. The velocities in simulation were also corrupted with process noise of $10^{-5}\ units/\sqrt{Hz}$. Fig.5.3 clearly shows that after just a few samples in the simulation, estimates yielded by the proposed method are more accurate than the finite difference discretization method. At the end of 30 seconds, the estimates obtained using the proposed method are *at least one order of magnitude* more accurate. For the next 30 seconds, the measured states and estimated parameters were used to calculate the residual errors (forces and torques predicted as $\hat{T}_{\underline{N}} = H_{\underline{N}}\hat{\boldsymbol{\pi}}$). Three of them are plotted in fig.5.4 along with their measured values. Finally, estimation results for three other sets of inertial parameters with noisy measurements are given in table 5.1. This table clearly shows that estimate accuracy benefits from using the discrete variational approach for full body inertial parameter estimation, i.e, mass, center of mass offset and inertia parameters.

	Inertial parameters										Accuracy relative to FD		
#	J_{xx}	J_{yy}	J_{zz}	J_{xy}	J_{xz}	J_{yz}	r_x	r_y	r_z	m	Inertia	CoM offset	Mass
1	2.08	3.19	4.06	-0.0487	0.0739	-0.0655	-0.08	0.06	0.15	10	2.09×10^1	1.20×10^1	2.18×10^2
2	5	5	5	0	0	0	0.1	0.2	0.1	5	1.92	3.05	2.01×10^1
3	0.0453	0.0417	0.0519	0	0	0	0	0	0	6.047	1.06	2.42	9.12

Table 5.1: Estimation accuracy due to the proposed method relative to accuracy obtained with the finite difference method. The relative accuracy is calculated as $\frac{\text{error}_{\text{FD}}}{\text{error}_{\text{V}}}$, for inertia, CoM and mass. (FD: finite difference, V: variational)

5.6 Discussion

Discrete variational integrators exhibit the key properties of symplecticity, momentum conservation and numerical efficiency. They are stable and respect energy dissipation, even in long term simulation. Further, Lie group variational integrators preserve the underlying geometric structure as the system evolves on the Lie group, without requiring constraints or reprojection. They are better suited for estimation applications over non-symplectic discretizations (e.g., finite difference) because they: 1) often yield superior results in their description of system evolution 2) are not a function of acceleration measurements, which can be noisy, difficult to obtain or need differentiation of pose data. A key benefit of using SDP is that linear constraints (such as bounds on the parameters to be estimated) can be incorporated by adding diagonal elements to the LMI. This retains convexity, and the problem can be solved (if a solution exists) efficiently by available fast solvers, as compared to nonlinear optimization.

Note that the linear least squares problem formulation in (5.30) only assumes noise on the measured torques. Therefore, the pose data is considered to be noise-free for all practical purposes. Although the VI-based method exhibited better performance than the FD approach to discretization even in the case of simulation results with a certain degree of noise in pose data, a high level of noise could easily prove detrimental to estimation accuracy irrespective of approach. Secondly, each iteration of this algorithm took 0.2 seconds, making it suitable for real-time applications. In case the measurement data stream has a higher frequency, either selective (e.g., informative) data points can be used, or the computation can be sped up by experimenting with solver options or ways of posing the problem. Finally, a potential drawback of variational integrators is scalability. They are perceived as difficult to scale for large, complex systems without specialized coordinate choices. However, works like [91] are changing that by proposing scalable VI methods for arbitrary mechanical systems in generalized coordinates.

Chapter 6

Information-aware online parameter learning

Methods explored in the previous chapters perform system identification (SysID) with the goal of using the learned parameters for optimal planning and precise control. SysID is thus performed prior to the motion task. However, it might be undesirable to suspend the primary task execution for complete sysID at every instance that the parameters change. An example of this is cargo transportation, where a sysID procedure for each cargo object could be time- and fuel- consuming. Further, as discussed in chapter 3, parameter identification is often preceded by offline calculation of excitation trajectories that make the parameters observable [21], [46], [58]. Such a pre-computed excitation trajectory might not be feasible for robots operating in dynamic environments. Additionally, classical sysID is an offline procedure where excitation measurement data is processed in batch, so online incorporation of the learned model or a real-time response to changing model parameters is not addressed.

To that end, a method called **R**eal-time information-**A**ware **T**argeted **T**rajectory planning for **L**earning via **E**stimation, or RATTLE was proposed. Belonging to the category of simultaneous learning approaches (as defined in chapter 3), RATTLE explicitly factors in information-awareness for model-learning during primary task execution. It consists of a global planner, a local planner to trigger the information-awareness as desired, and a robust model predictive controller to provide robustness guarantees against parametric and other additive uncertainties. Additionally, RATTLE's online local planner and controller model updating enable it to leverage the learned parameters, thus reducing parametric uncertainty on-the-fly for safe and precise task execution.

With the microgravity cargo transportation scenario in mind, RATTLE easily lends itself to a broad range of applications where mobile robots face parametric uncertainty. RATTLE was demonstrated in experiments on the Astrobee robotic free-flyers on board the International Space Station (ISS) (fig. 6.1), establishing its capabilities of real-time and robust parametric uncertainty reduction on resource-constrained hardware in micro-gravity environments. To the author's knowledge, this was the first time that an information-aware motion planning framework has been evaluated in space. Results have been published in [61–64], and preprint [145].

Figure 6.1: Astrobee Honey performing an informative maneuver during a RATTLE experiment on-board the ISS [Credit:NASA/ISS]

RATTLE was developed in collaboration with Keenan Albee from the Massachusetts Institute of Technology's Space Systems Lab as a part of his PhD thesis, [146]. The author's contribution to RATTLE included the information trajectory computation and the parameter estimation module, components which are described in detail in this chapter.

6.1 Problem Formulation

A robotic system with state $\mathbf{x} \in \mathbb{R}^n$, input $\mathbf{u} \in \mathbb{R}^m$, and uncertain parameters $\boldsymbol{\theta} \in \mathbb{R}^j$ is positioned at state $\mathbf{x}_0$ at $t = 0$. A region of the state space that is admissible is specified as $\mathcal{X}_{free}$, and a constraint on inputs is provided as $\mathcal{U}$. A goal region $\mathcal{X}_g$ is also specified. Let the discrete dynamics and measurement models of the system be represented as

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\theta}) + \mathbf{w}_{x,k} \tag{6.1}$$

$$\tilde{\mathbf{y}}_k = h(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\theta}) + \mathbf{w}_{y,k} \tag{6.2}$$

where the vector of the measured quantities is $\tilde{\mathbf{y}} \in \mathbb{R}^l$, $\mathbf{w}_{\mathbf{x}} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_Q)$, and $\mathbf{w}_{\mathbf{y}} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_R)$ where $\mathcal{N}$ represents Gaussian noise on the process and measurement model, respectively. Only a prior distribution on the parameters is known, $\boldsymbol{\theta}_0 \sim \mathcal{N}(\hat{\boldsymbol{\theta}}_0, \boldsymbol{\Sigma}_0)$.

The aim is to plan a trajectory minimizing the following cost function while respecting the input and state constraints, $\mathcal{U}$ and $\mathcal{X}_{free}$,

$$J = g(\mathbf{x}_N) + \sum_{k=0}^{N-1} l(\mathbf{x}_k, \mathbf{u}_k) \tag{6.3}$$

Here, $g(\mathbf{x}_N)$ is a terminal cost and $l(\mathbf{x}_k, \mathbf{u}_k)$ is a running cost, computed using the current knowledge of the system model. Since knowledge of $\boldsymbol{\theta}$ can be improved through parameter estimation, it is possible to obtain an enhanced dynamics model that is a closer representation of the real system.

Note that this problem is formulated using the discretized continuous equations of motion in place of discrete variational mechanics. In addition to the estimation approach using variational mechanics presented in chapter 5, the solution of optimal control problems based on discrete mechanics has been proposed [147], and explored in multiple contexts [148, 149]. However, for reasons of simplicity and ease of validation of RATTLE's information-aware parameter learning concept, the integration of discrete mechanics in the problem formulation is left for future work.

6.2 Approach

At its core, RATTLE is an information-aware motion-planning framework, with the goal of gaining and incorporating the most recent parameter estimates on-the-fly. Emphasis on the information content in the trajectory can be adjusted, for instance, depending on the amount of parametric uncertainty in the system. Recently learned parameters are incorporated using online-updateable planner and controller models. Therefore, RATTLE bridges the gap between pure system identification approaches and model-based planning and control that calls for accurate parameter knowledge. This is achieved by means of the following four components:

- A high-level (global) planner
- A mid-level (local) information-aware planner
- A low-level robust model predictive controller
- An online parameter estimator

A schematic of the planning components in the framework is given in fig. 6.2. The high-level global planner, the information-aware mid-level planner and the low-level controller all work in different time scales. The high-level planner provides a long-horizon dynamically-feasible collision-free global plan. The mid-level planner uses updated information about the system and the environment to plan locally, introducing information-richness based on the weighting parameter $\Gamma \in \mathbb{R}^j$ for j unknown parameters. These plans are executed in a closed-loop using a low-level controller with adjustable robustness guarantees based on the parametric uncertainty levels. This is combined with a parameter estimator that yields latest estimates of desired inertial parameters in real-time. The confidence level of the estimates, i.e. their variance, are used to determine the value of Γ.

6.2.1 High-Level (Global) Planner: Kinodynamic RRT

This framework uses sampling-based planning, more specifically, Kinodynamic-RRT, a variant of the rapidly exploring random tree (RRT) algorithm for global planning. Kinodynamic-RRT generates waypoints and is suitable for long-horizon planning in cases where optimization-based planning becomes intractable [150, 151]. Ellipsoidal-on-ellipsoidal obstacle representation is used for efficient and convex continuous collision checking [152]. The result of this initial long-horizon planning is a path, $\mathcal{P}_g$, of $\mathbf{x}_{0:N_g} \in \mathcal{X}_{free}$, where each node obeys $\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k, \theta)$ and any additional enforced constraints. Dynamics propagation is typically accomplished using a set of representative motion primitives. For

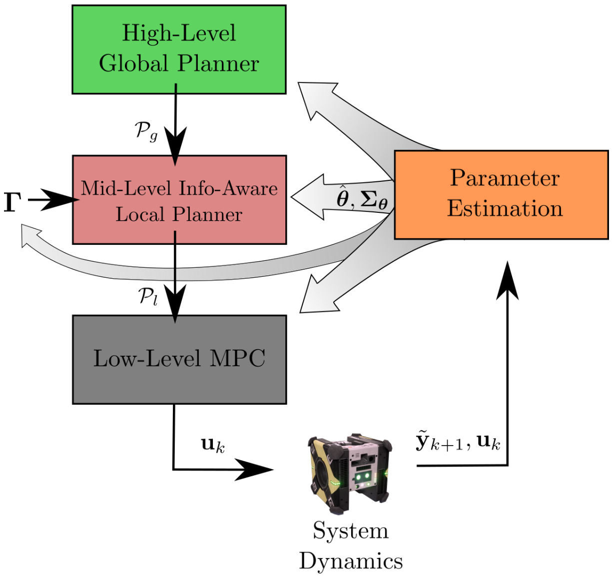


Figure 6.2: Components of the RATTLE motion-planning framework. A global planner that provides high-level plans, a local planner that provides mid-level, informative plans by autonomously adjusting the information weighting factor, Γ, and a low-level model predictive controller. Using the control inputs and state measurements, the model-based parameter estimator outputs the latest estimates, which are leveraged to update the controller and planner models, with the aim of improving tracking accuracy and optimality. Diagram from [64]

computation speed, RATTLE's Kinodynamic-RRT considers translational dynamic feasibility, and dynamics propagation is accomplished using motion primitives. These adaptations enable real-time global path computation while incorporating the most recent model and environment information.

6.2.2 Mid-Level (Local) Planner: Information-Aware Receding Horizon Trajectory Planner

The mid-level planner performs receding-horizon, information-aware planning. Starting off with the high-level, global plan $\mathcal{P}_g$ given by the Kinodynamic-RRT, the planner plans trajectories between waypoints using updated information about the robot's parameters, $\boldsymbol{\theta}_k$. Notably, this planner has the ability to optimize a cost function that promotes excitation by maximizing a weighted trace of the Fisher Information Matrix (FIM), thus facilitating the estimation of dynamic parameters *alongside* state error and input minimization.

Assuming that the parameters remain constant throughout the trajectory i.e., $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k$, and due to the Gaussian nature of the measurement noise and linear measurement model, over time $k = 0$ to $k = N$, the FIM can be found as:

$$\mathbf{F} = \sum_{k=0}^{N} \mathbf{H}_k^T \boldsymbol{\Sigma}^{-1} \mathbf{H}_k \tag{6.4}$$

where

$$\mathbf{H}_k = \frac{\partial h(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} + \frac{\partial h(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\theta})}{\partial \mathbf{x}} \cdot \frac{\partial \mathbf{x}(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \tag{6.5}$$

Let

$$\boldsymbol{\phi}_k = \frac{\partial \mathbf{x}(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \tag{6.6}$$

Eq. (6.6) can also be written as (note that subscript k is dropped for brevity):

$$\boldsymbol{\phi} = \begin{bmatrix} \frac{\partial x_1}{\partial \theta_1} & \cdots & \frac{\partial x_1}{\partial \theta_j} \\ \vdots & \ddots & \vdots \\ \frac{\partial x_n}{\partial \theta_1} & \cdots & \frac{\partial x_n}{\partial \theta_j} \end{bmatrix} \tag{6.7}$$

Then, the value of $\boldsymbol{\phi}$ can be obtained by the solution of the following ordinary differential equation, with initial conditions $\boldsymbol{\phi}(0) = \{0\} \in \mathbb{R}^{n \times j}$ [153], [49], [61].

$$\dot{\boldsymbol{\phi}} = \frac{\partial f(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\theta})}{\partial \mathbf{x}} \boldsymbol{\phi} + \frac{\partial f(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \tag{6.8}$$

As is common in trajectory optimization problems, dynamics equation (6.3) is discretized. For each local plan period, the optimization problem solved by the mid-level planner over the horizon of length L time steps, starting at time instant t, is

$$\begin{aligned} \underset{\mathbf{x},\mathbf{u}}{\text{minimize}} \quad & \sum_{k=0}^{L-1} \mathbf{x}_{t+k}^T \mathbf{Q} \mathbf{x}_{t+k} + \mathbf{u}_{t+k}^T \mathbf{R} \mathbf{u}_{t+k} + \Gamma^T \, diag\left(\mathbf{F}^{-1}\right) \\ \text{subject to} \quad & \mathbf{x}_{t+k+1} = f_n(\mathbf{x}_{t+k}, \mathbf{u}_{t+k}, \boldsymbol{\theta}), \; k = 0, .., L-1 \\ & \mathbf{x}_{t+k} \in \mathcal{X}_{free}, k = 0, .., L, \\ & \mathbf{u}_{t+k} \in \mathcal{U}, k = 0, .., L-1, \end{aligned} \tag{6.9}$$

where N is the length of the horizon and $\mathbf{Q} \succ 0$ and $\mathbf{R} \succ 0$ are positive definite weighting matrices and f_n denotes the discretized process model from eq.(6.1). The relative weighting term $\Gamma \in \mathbb{R}^j$ is used to adjust the emphasis on the information content in the trajectory for each parameter. The operation $diag$ extracts the diagonal elements of the inverse FIM.

Variance-based information weighting

Feedback from the current model knowledge in the form of estimate covariance is used to determine the trade-off between maximizing excitation and minimizing state error. Each element γ_i of Γ is a decaying exponential, with the ratio between a minimum desired parameter variance ($\sigma_{n,i}$) and the current variance ($\sigma_{k,i}$, where $\sigma_{k,i}$ denotes the ith diagonal element of Σ_k) as the decay rate for each local plan period. In other words, if the estimate covariance is higher than the minimum desired value, the information weighting decays slower. This decay is sped up the closer to the threshold, and information weighting is shut off completely beyond a tolerence level, making the desired trajectory solely goal-oriented once a desired confidence level in the estimates is achieved. This approach is shown in Algorithm 1 [1].

Algorithm 1 The covariance-based information weighting procedure. α tunes tolerance above a minimum desired parameter variance, $\sigma_{n,i}$, and β tunes the rate of weighting decay. $\mathbf{\Gamma}$ is assigned based on the tuning and the latest estimate covariance update, $\mathbf{\Sigma}_{k+1}$.

```
procedure CovarWeight($\mathbf{\Sigma_k}, \mathbf{\Gamma_0}, \alpha, \beta$)
    $\mathbf{\Sigma}_{k+1} \leftarrow$ UpdateCovar($\mathbf{\Sigma}_k$)
    for $\forall i$ do
        if $\sigma_{k,i} \leq \alpha\sigma_{n,i}$ then
            $\gamma_{k,i} \leftarrow 0$
        else
            $\gamma_{k,i} \leftarrow \gamma_{0,i} e^{-\frac{\beta\sigma_{n,i}}{\sigma_{k,i}}}$
```

6.2.3 Low level controller: Robust Model Predictive Control

A Low level robust MPC is used for executing the local plans. The controller guarantees that under a bounded, additive disturbance, $\mathrm{w}_x \in \mathbb{W}$, the system will stay in a minimum Robust Positively invariant set (mRPI) centered at the nominal trajectory plan. More specifically, RATTLE uses linear tube MPC for translation motion. For every horizon, an updated model and computation of the mRPI is carried out. The robust MPC provides guarantees against collision during the transient phase of the parameter estimation by including the uncertainty from the parameter covariance levels in w. A separate tuned Proportional-Derivative (PD) controller is used for attitude control [146].

6.2.4 Parameter estimator: The Extended Kalman Filter

RATTLE uses an Extended Kalman Filter (EKF) for real-time sequential parameter estimation. The EKF (detailed further in Appendix A.2.2) is a non-linear extension of the Kalman Filter, obtained by first-order Taylor linearization of the error dynamics about the current estimate. For parameter estimation using the EKF, the parameters are assumed to be gradually varying. They are subject to a random walk caused by white Gaussian noise, $\mathbf{w}_{\boldsymbol{\theta}} \sim \mathcal{N}(0, \Sigma_\theta)$ [154], which helps to continue improving the estimate. The discretized process model is given by

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \mathbf{w}_{\boldsymbol{\theta},k} \tag{6.10}$$

This results in the process Jacobian being an identity matrix. The discretized measurement model is

$$\tilde{\mathbf{r}}_k = \mathcal{H}(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\theta_k}) + \mathbf{w}_{r,k} \tag{6.11}$$

where $\mathbf{w}_{\mathbf{r}} \sim (0, \Sigma_{\mathbf{r}})$ and $\mathcal{H}()$ denotes the measurement model. The measurement Jacobian is given by $\mathbf{H} = \frac{\partial\mathcal{H}(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\theta}_k)}{\partial\boldsymbol{\theta}}|_{\boldsymbol{\theta}_{k|k-1}}$.

---

[1] The generation of excitation trajectories addressed in sec.4.2 discussed two optimization criteria for ensuring full-rank and numerical stability of the regressor: minimizing the condition number and maximizing the log likelihood of the determinant of the information matrix (E-optimality). For the simulation dataset, 4.5.2 showed the condition number criterion to result in a higher parameter accuracy. However, to study the impact of maximum information content on estimation accuracy, an excitation criterion based on Fisher Information Matrix was employed in this approach. Further, to tune the relative weighting for different parameters of interest, A-optimality, i.e. minimizing the trace of the inverse FIM is used instead of E-optimality.

6.3 The Astrobee testing platform

NASA's Astrobee robots are cube shaped free-flyers deployed on board the ISS to research and test autonomy algorithms (further details in appendix B). In particular, Astrobee is being used to develop autonomous caretaking abilities including inspection, monitoring and logistics such as cargo unpacking and transportation [10, 155]. Therefore, testing RATTLE on Astrobee enabled a realistic evaluation of RATTLE's abilities. Before RATTLE could be deployed in micro-gravity, it was tested and refined on the Astrobee simulation environment and the Astrobee ground testing facility.

Figure 6.3: Astronaut David Saint-Jacques preparing Astrobee for an experiment aboard the International Space Station's Kibo module. [Credits: NASA/Johnson Space Center]

Astrobee is equipped with three Snapdragon ARM processors which communicate using an Ethernet Network [156, 157]. The Low Level Processor (LLP) controls the robot propulsion and the high-rate control loop, while the Mid Level Processor (MLP) carries out resource-intensive vision and planning algorithms. The LLP and MLP both run Linux, while the High Level Processor (HLP) runs Android. The HLP therefore manages interaction devices such as the camera and microphone. It also provides the encapsulation capabilities of Android Packages for Guest Science. The robot code is written in C++. Commands can be sent to the robot using the Ground Data System (GDS) interface from the control station on the ground or the ISS. These commands, issued using the Data Distribution Service (DDS) standard, are transformed to ROS by Astrobee's DDS bridge [156].

The **Astrobee flight Software** (FSW)[2] uses Robot Operating System (ROS) [158] as middleware for message-passing, with multiple nodes running on Astorbee's LLP and MLP. The Astrobee Software Simulator (visualization shown in fig. 6.4) replaces the FSW's hardware drivers with Gazebo plugins, providing the same interfaces to the flight software as the Astrobee hardware. It consists of high-fidelity models of Astrobee subsystems, including propulsion and navigation [156, 157]. Individual components of RATTLE and their collective performance was evaluated using the Astrobee Software Simulator.

Mobility and Guidance, Navigation and Control are Astrobee's key subsystems relevant to the deployment of RATTLE on hardware. A brief description of these subsystems and the updates that were

---

[2] https://github.com/nasa/astrobee

made to integrate RATTLE (fig. 6.5) are summarised in the following:

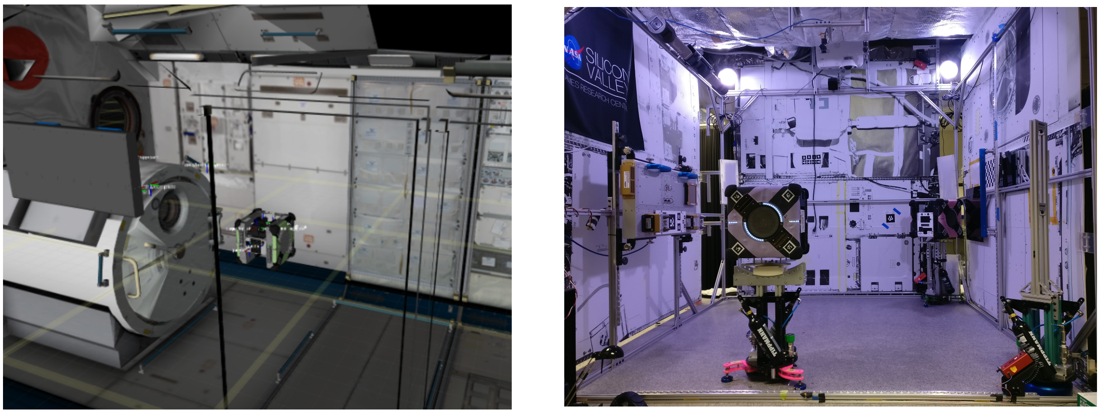


Figure 6.4: Left: Astrobee inside the Japanese laboratory Module on the ISS in the Astrobee flight software simulation environment. Right: Astrobee free-flyer BSharp on its air-bearing on the granite table at Ames research center, CA, USA.

- Astrobee's **Mobility**[3] subsystem is in charge of ensuring safe motion of the robot [156]. It performs obstacle avoidance, accounts for keep-in zones, and considers kinematic constraints. Mobility consists of the *Choreographer* which manages the mobility subsystem, the *Mapper*, which maintains a map of the environment and its clutter, against which the choreographer validates its trajectories, and *Planner* which contains two planner nodes: Trapezoidal planner, which is a straight-line planner used primarily in docking and perching or to generate a trajectory between a sequence of poses, and Quadratic program planner which generates smooth, complex plans and avoid obstacles, with segments represented as polynomials [159].
- The **GNC**[4] susbsystem is responsible for trajectory tracking and control. It consists of an *Multi-State Constraint Kalman Filter* (MSCKF referred to as EKF in the FSW) [160] which takes in IMU and visual measurements from the localization subsytem to estimate the robot's state, *Control* (CTL) which uses a Simulink auto-generated C code wrapped into ROS nodelets [156] to determine the forces and torques to track the plans sent by the mobility subsystem, and the *Force Allocation Module* (FAM), which converts the forces and torques to commands for the *Propulsion Module Controller* (PMC). The PMC commands the impeller speeds and nozzle opening angles. Note that Astrobee generates thrust by drawing in air through two central impellers and expelling it via 12 exhaust nozzles (see Appendix. B).
- The **Localization** (LOC) subsystem consists of a graph-based visual inertial localizer, AstroLoc [161]. Inputs to the localization algorithm include optical flow feature tracking, IMU acceleration and angular velocity data, and one of the following three types of features: sparse map features, AR tags (particularly for docking) and handrail measurements [160]. Acknowledging Astrobee's heavy reliance on Visual Interial Odometry (VIO) for navigation due to map landmarks being unavailable, AstroLoc offers robustness against chirality issues in VIO and improvements in IMU bias estimation

---

[3] https://nasa.github.io/astrobee/html/mobility.html
[4] https://nasa.github.io/astrobee/html/gnc.html

to reduce large localization drifts. In the latest versions of the FSW, the EKF node has been removed from the GNC subsystem, with AstroLoc performing localization.

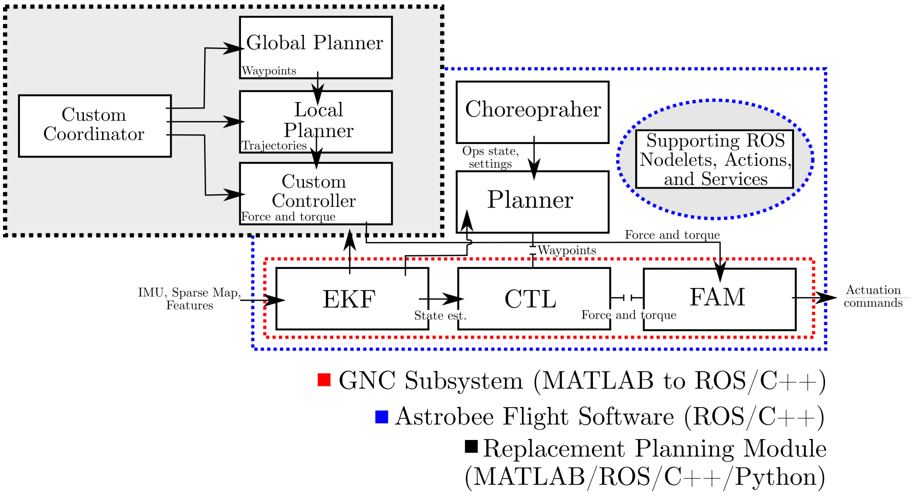


Figure 6.5: Astrobee autonomy stack and additions made to deploy RATTLE [162]. Note that the MATLAB models were converted to C++ in the Astrobee FSW v0.16.7 released in December 2022.

The deployment of RATTLE on Astrobee required the introduction of multiple custom modules into the flight software, as illustrated in 6.5. The mobility subsystem was replaced by a custom planning module, with a coordinator managing interaction between the individual components - the global and local planners, and the model predictive controller. The planning module used pose estimates from the EKF, and the control outputs (forces and torques) were sent to the FAM, from where they were converted to PMC commands. These updates were documented in a Brief Guide to Astrobee's Flight Software [162] and made publicly available to the broader Astrobee research community.

The **Astrobee ground testing facility** at NASA Ames Research Center's Granite Lab provides the opportunity to deploy and test code on robot hardware on the ground. The robots are placed on air-bearing, floating freely on a $1.5 \times 1.5m$ granite table surrounded by a mock-up of the interior of the space station (fig. 6.4) [163]. The robot motion is limited to x, y and yaw directions, providing opportunities for testing fine-tuned movement control, manipulation and localization [157]. An important step between simulation and microgravity testing, multiple ground-test opportunities enabled the evaluation of RATTLE's performance on the resource-constrained Astrobee hardware. Ground-testing also brought to light issues including computation delays that led to fine-tuning various RATTLE parameters such as the planner and controller periods and gains, thus preparing RATTLE for real-time execution in microgravity.

On the **International Space Station**, the Astrobees are docked and move around the Japanese Experiment Module (JEM), with roughly $1.5 \times 6.4 \times 1.7m$ testing volume [162]. This unique setup enables the evaluation of autonomy algorithms on Astrobee at its full holonomic 6DoF motion capability in a micro-gravity environment. To the best of the author's knowledge, RATTLE was the information-aware motion-planning algorithm to be demonstrated on Astrobee, and one of the first to make significant updates to the Astrobee software stack.

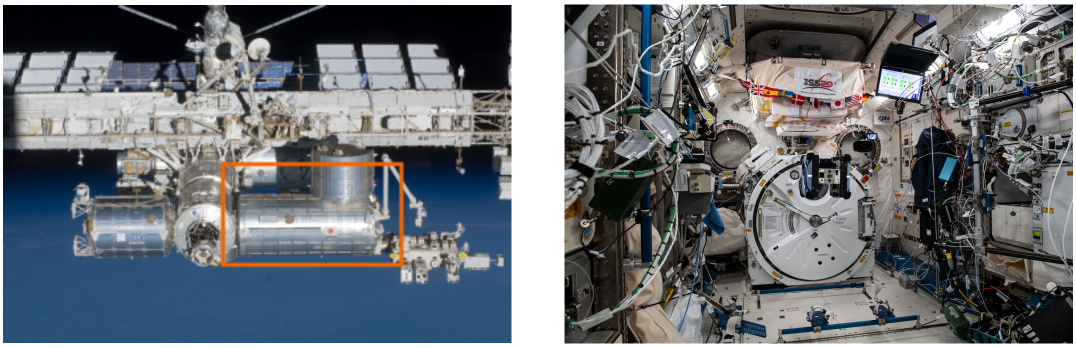


Figure 6.6: The Kibo/Japanese experiment Module (JEM) that houses Astrobee on the ISS (L, highlighted), and Astrobee Honey in the JEM (R). Honey has since been downmassed due to an SD card issue, but two others, Bumble and Queen are active on station. [Credit: NASA]

6.4 Results

This section presents results from RATTLE's implementation using the Astrobee simulation, granite and microgravity testing facility. Uncertain inertial properties were assumed at the beginning of each test, with the uncertain model used by the local planner and the controller. The objective of each test was for the robot to safely evade obstacles and arrive at the goal point with an improved knowledge of its inertial parameters. To evaluate and compare the effectiveness of RATTLE, the tests were performed with varying initial and environmental conditions: with and without information gain, with and without attached payload, and with different inital parameter values and obstacle arrangements.

Platform	Simulation/Hardware	DoF
Astrobee simulator [Gazebo]	Simulation	3, 6
Granite testing facility [NASA Ames]	Hardware	3
Microgravity testing facility [ISS]	Hardware	6

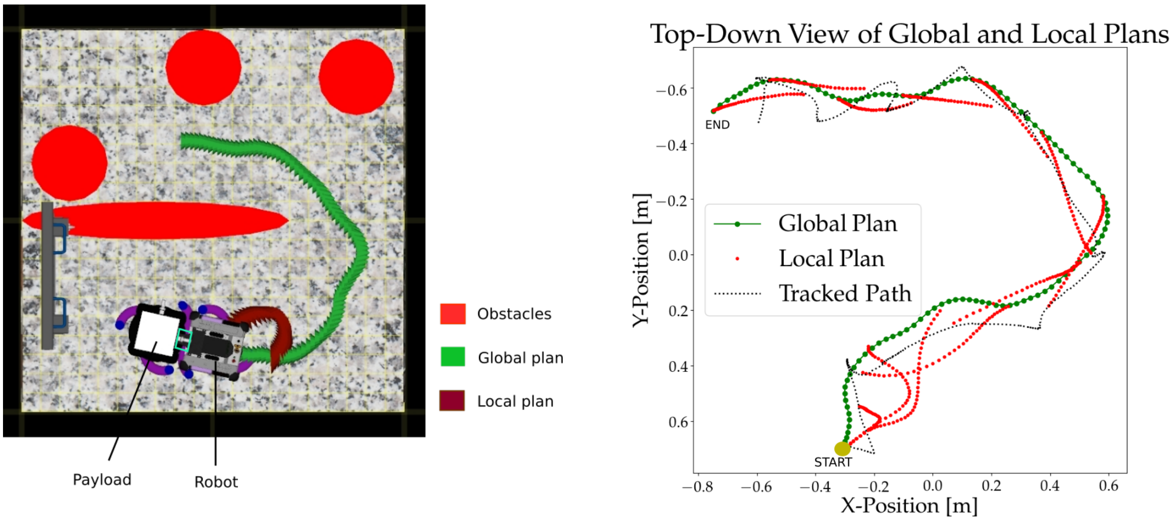
Table 6.1: A Summary of the Astrobee experimental platforms

6.4.1 Implementation details

Fast, online computation of the planning and control components, as well as coordination between modules for timely relaying of the local plans, control commands and parameter estimates, were essential for RATTLE's hardware operation. The use of translational dynamics only and motion primitives kept the global planner replan times low, taking between 0.5 and 2 secs on hardware. This also allowed the possibility of online replanning. For the receding horizon local planner, the optimization problem was implemented using the ACADO toolkit's C++ bindings [142], including keep-out-zone and simulated obstacle constraints and Astrobee's thrust limits. CasADi [164] was used for implementing the robust tube MPC. In addition, Bullet Physics' C++ collision checker [165] and Autograd [166] were used for collision detection and automatic differentiation respectively. The local plan period was 12 secs, with a speed of between 2 and 7 secs for local plan computation, while the MPC operated at 5 Hz, taking between 0.05 and 2 secs for each computation. The Extended Kalman Filter as described in sec. 6.2.4) estimates the mass and principal moments of inertia of the system using the commanded actuation and measurement histories. To keep up with the incoming data stream of the localization and control output ROS topics (about 62.5Hz, see section 6.5.1), the EKF processes the data as soon as it arrives. The dynamic models used by the local controller and planner are update-able, incorporating the most recent inertial parameter knowledge at the time of the next planner and controller horizon. For testing the improvement in estimation accuracy due to the uncertainty-aware framework, the robot was initially considered to have uncertain values of mass and moments of inertia. Before the simulation and hardware results presented in this section, the RATTLE algorithm was evaluated and improved based on the results of tests carried out on its earlier iterations such as [61].

6.4.2 High-fidelity simulation results

As sec. 6.3 mentions, RATTLE was tested extensively on the Astrobee Software Simulator before its deployment on hardware. Fig. 6.7 illustrates the performance of the various RATTLE components in a proof of concept experiment using the 3DoF granite table simulation environment. Astrobee navigates among obstacles (shown in red) to reach the goal as shown in fig. 6.7(a). The information-aware local plan adds excitation between the current robot pose and the next global plan waypoint. Fig. 6.7(b) illustrates a sample run of RATTLE, showing the local plan segments with decreasing information content towards the end of the trajectory.



(a) (b)

Figure 6.7: Implementation of RATTLE in the Astrobee simulation environment. (a)Top view of the 3DoF granite lab simulation setup with the robot shown grasping a payload and the blue rectangle highlighting the point of attachment (for the real world reference, see Fig. 6.4(b) and 6.11), representing an environment cluttered with obstacles inflated for Astrobee radius (b) Example of the executed trajectory in simulation. Note the reduction of excitation in local plans towards the end of the trajectory.

RATTLE components in action in the 6DoF simulation environment are shown in fig.6.9. The objective is to navigate through multiple simulated obstacles to reach the goal point. The robot starts off with incorrect inertial parameters and high parametric uncertainty, resembling an inertial reconfiguration

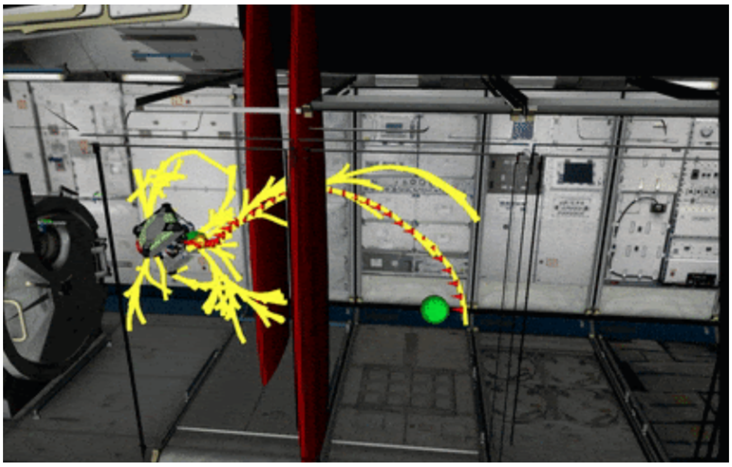
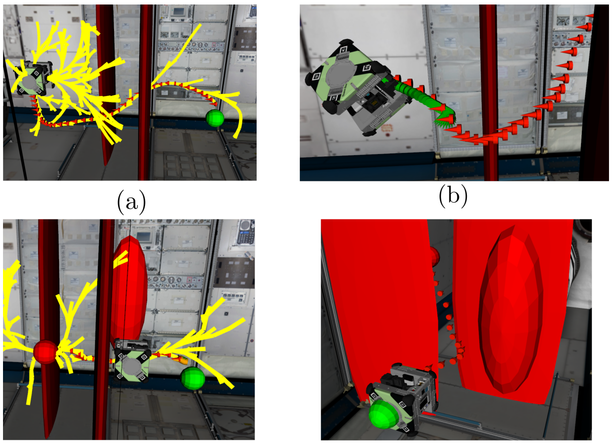


Figure 6.8: The Astrobee 6DoF simulation environment shows the robot inside the Kibo module. The planes in red are obstacles added to the environment as the robot goes from the start point to the goal point, shown in green. The yellow branches illustrate the expansion of the RRT tree.

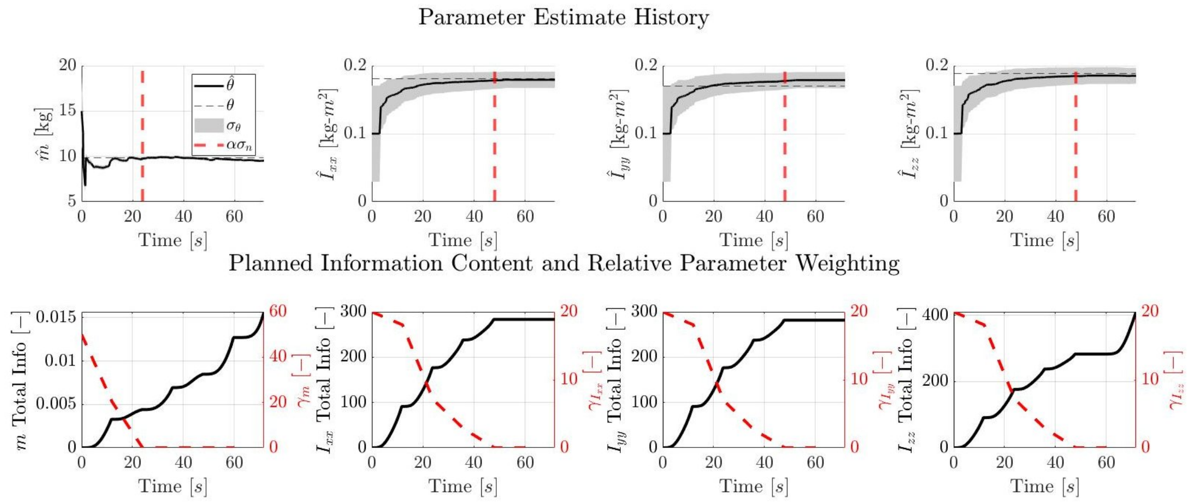
scenario. The kinodynamic RRT plans an initial global plan, the tree shown in yellow. Next, based on the information weighting Γ, the local plan incorporates information content between the global plan segments. An instance of global replanning is also illustrated in (c), where an 'astronaut' obstacle appears near the global plan, presenting a risk of collision. The new global plan is then tracked to reach the goal.



(a) (b) (c) (d)

Figure 6.9: Components of The RATTLE framework in action: (a) the kino-RRT trajectory from start to the goal point, (b) the information-aware local plan segment shown in green, (c) a global replan occurs due to the insertion of an obstacle and (d) the robot makes it to the goal point after avoiding the new obstacle.

The inertial parameters of interest in this simulation experiment are mass and the principal moments of inertia. Fig. 6.10 shows the estimate evolution and information content per parameter. The series of figures on the bottom shows the cumulative information content (in black) and the relative information weighting (in red) for each parameter. Note that the mass information gain keeps increasing regardless of the explicit information content shut off, indicating that mass information content is obtained for "free". A similar effect is seen for I_{zz}, where the global plan contained a yaw rotation to reach the goal point. The net effect of explicit information weighting is to increase information content in the plans that


Parameter Estimate History

Planned Information Content and Relative Parameter Weighting

Figure 6.10: Online parameter estimates with σ bound covariance are shown (top). The dashed vertical line indicates information weighting shutoff, triggered by estimate confidence reaching specified levels. The figures below show the information content and the relative information weighting for each parameter. The information content increases when the parameter weighting is active.

otherwise would not have sufficient information gain. The effect of information weighting is reflected in the parameter estimates, shown in the top row of fig. 6.10. The dotted vertical line illustrates the estimate covariance-based information weighting shutoff, which corresponds to the plateau in the parameter estimate values, indicating that a specified noise-floor is achieved.

6.4.3 Granite Lab experimentation results

The ground-testing set up with payload attached is shown in fig.6.11, and the results are shown in fig.6.12. Results show that in spite of incorporating excitation, the inertial parameter estimates do not converge to their known values during experiments, in particular for the case with payload. After examining the measured data, input saturation and unmodeled dynamics were thought to be the cause for this behaviour. A detailed discussion of how these factors and some others were addressed to obtain a considerable amount of improvement in estimate accuracy is discussed in this section.

Hardware	Astrobee+ Arm+Carriage	Payload+ Carriage	Combined System (I_{zz} about CM)
m $[kg]$	19.0	11.8	30.8
I_{zz} $[kg-m^2]$	0.25	0.015	0.94
c_x $[m]$	0.0	0.0	0.0
c_y $[m]$	0.0	$-.305$	-0.12

Table 6.2: Hardware ground truth values. Note that hardware values are approximations, accounting for gas level, arm extension, and number of batteries used.

6.4.4 Microgravity experiment results

The goal of RATTLE's microgravity experiments was to demonstrate real-time hardware operation and online parameter uncertainty reduction. The virtual obstacles shown in fig. 6.9 were also considered

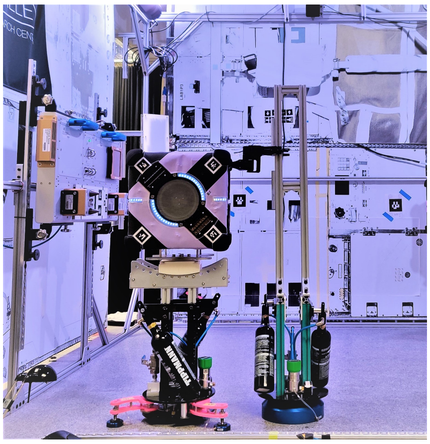


Figure 6.11: Astrobee on its air-bearing and grappled payload.

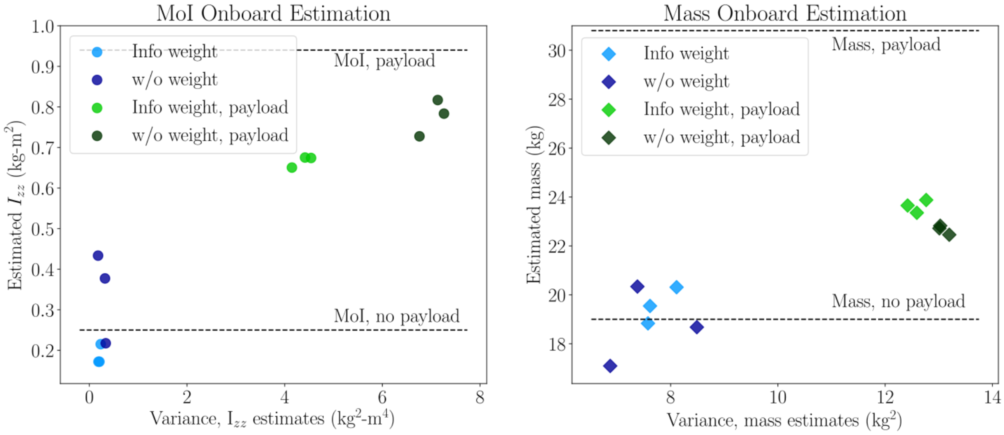


Figure 6.12: Initial parameter estimation results for ground testing on the Astrobee hardware. Plots show the effect of using incentivized information gain on the estimation of mass and inertia (MoI, I_{zz}). Test cases with our method and with a normal receding horizon planner are shown, each with and without the payload shown in fig.6.11 attached.

	Without Payload	With Payload
I_{zz} Covariance [% Change]	-25.01%	-38.05%
m Covariance [% Change]	2.47%	-3.71%

Table 6.3: Parameter estimate variance reduction of information-aware plans relative to non-informative plans for hardware testing. Decreases indicate greater precision of estimated model parameters. Both “without payload” and “with payload” cases are shown (average of three runs at the final timestep of motion).

to exist in the ISS testing volume during the on-orbit demonstration. Fig. 6.13 illustrates the information content for an informative trajectory and a non-informative trajectory, where zero explicit information weighting was used. Fig. 6.15 shows the parameter estimation results for these two runs. Similar mass estimation accuracy and information content are obtained for both runs, while the moments of inertia estimates benefit from increased information content, resulting in estimation progress till convergence. An increase in information content is observed in I_{zz} in both cases due to the global plan. Even though

RATTLE's planning and control modules ran completely online on Astrobee hardware, parameter estimates obtained on-orbit were impacted by the factors described in sec. 6.5. Therefore, the results presented here are obtained offline by replaying data recorded in microgravity. Post generation of these results, some other issues affecting the estimation were addressed, and the final estimation results are presented in sec. 6.5.

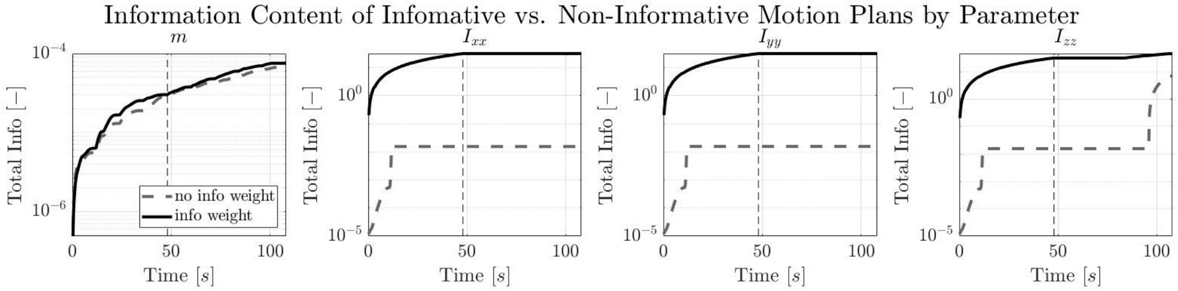


Figure 6.13: Cumulative information content for the informative and non-informative runs on-orbit. The mass information content is comparable, while the information weighting adds further excitation for the principal moments of inertias. Note that I_{zz} gets an additional information-boost at the end due to a yaw-rotation from the global plan.

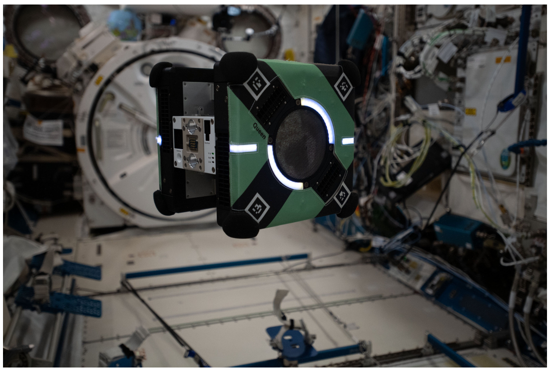


Figure 6.14: Astrobee freeflyer Honey on the International Space Station in August 2021. Honey has since been downmassed due to an SD card issue. [Credit: NASA/ISS]

m $[kg]$	I_{xx} $[kg-m^2]$	I_{yy} $[kg-m^2]$	I_{yy} $[kg-m^2]$
9.5838	0.1534	0.1427	0.1623

Table 6.4: Astrobee's known inertial parameters, obtained through calibration performed by NASA Ames.

6.5 Parameter Estimation convergence in the presence of noisy data

6.5.1 Time stamp ordering

The parameter estimation algorithm received the robot states and commanded control inputs by subscribing to the relevant ROS topics, denoted here as CTL for the control inputs, and EKF for the local-

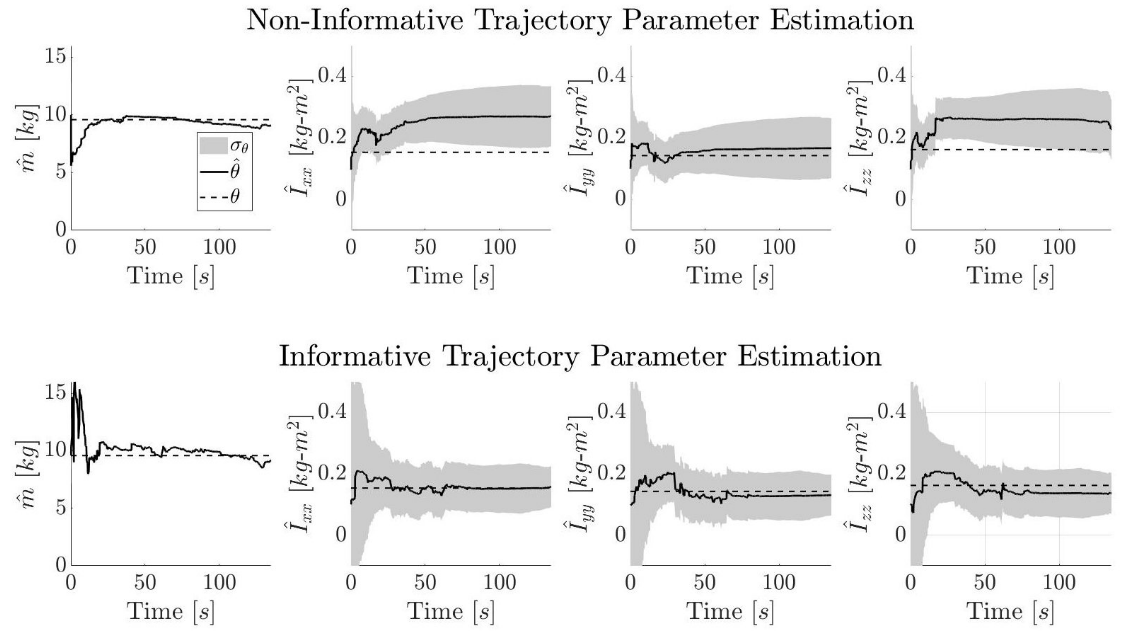


Figure 6.15: Inertial parameter estimates for the informative and non-informative runs. Explicit information weighting facilitates the converge of inertia estimates for the informative trajectory, which are more accurate and precise as compared to the non-informative trajectory. Note that comparable mass accuracy is obtained for both runs.

ization EKF data. The expected publication frequency of both these topics is 62.5 Hz. However, as fig. 6.16 illustrates, latency occurs on the regular on hardware; the initial decrease in frequency thought to arise from launching RATTLE nodes and global plan computation. After that, the CTL frequency picks up and maintains a steady rate, but the localization output frequency varies throughout the experiment. Moreover, the delayed signals often arrive with timestamps tenths of seconds earlier than when they are published. A time-stamp ordering algorithm was implemented to solve this issue. EKF and CTL data were separately collected till batches of 30 were formed (this value was tuned such that no messages would be lost in the process), and then sent to the *TimeStampOrder* function as shown in Algorithm 2, before being processed sequentially by the parameter estimator.

Algorithm 2 The time-stamp ordering algorithm used to synchronize control inputs (CTL) localization data (EKF) before use by the parameter estimator. The algorithm receives batches of CTL and EKF data, combines them in one array. This array is then sorted based on time stamps, and compared with a previously stored and sorted batch. The sorted signal is returned to be processed sequentially by the estimation algorithm.

Input: Two vectors of CTL and EKF data of 30 elements each

Output: A vector of 30 elements containing CTL and EKF data sequentially arranged based on their timestamps

```
procedure TIMESTAMPORDER($s_{ekf}, s_{ctl}$)
    $s_{sorted,k} \leftarrow$ CombineAndSort($s_{ekf}, s_{ctl}$)
    if $t_{sorted,k}[1] < t_{sorted,k-1}[end]$ then
        $s_{sorted,k\&k-1} \leftarrow$ CombineAndSort($s_{sorted,k}, s_{sorted,k-1}$)
        $s_{sorted,k-1} \leftarrow s_{sorted,k\&k-1}[0:30]$
        $s_{sorted,k} \leftarrow s_{sorted,k\&k-1}[30:end]$
    $s_{ordered} \leftarrow s_{sorted,k-1}$
    $s_{sorted,k-1} \leftarrow s_{sorted,k}$
```

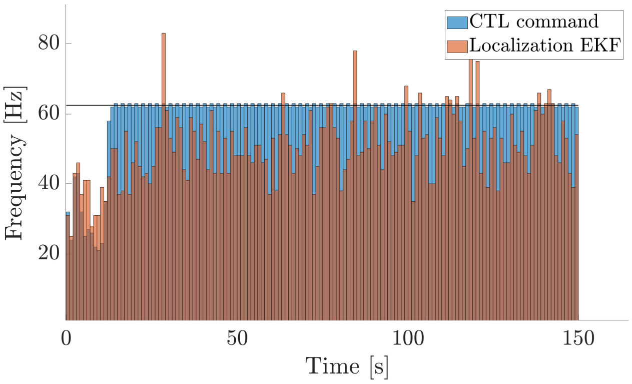


Figure 6.16: A histogram illustrating the frequency of the commanded control signals (CTL) and the localization estimates (EKF) on Astrobee hardware during a sample microgravity experiment. The bins correspond to the number of signals sent out per second (Hz). Note that the expected frequency of both signals is 62.5 Hz, as shown by the black line, but computation latency results in signals being dropped or delayed.

6.5.2 Measured data processing

To troubleshoot the estimation algorithm further, the projected linear accelerations, i.e., those found by dividing the commanded forces by the known true mass of Astrobee, were compared to the measured accelerations. The commanded forces and the measured accelerations are both in the body frame. Fig. 6.17 (L) shows the disparity between measured and projected accelerations on the X axis, which would ideally be expected to correspond. A similar scaling effect was observed for the accelerations on the Y and Z axis, but to a different extent. In order to account for this effect, the residual accelerations were modeled as a linear system.

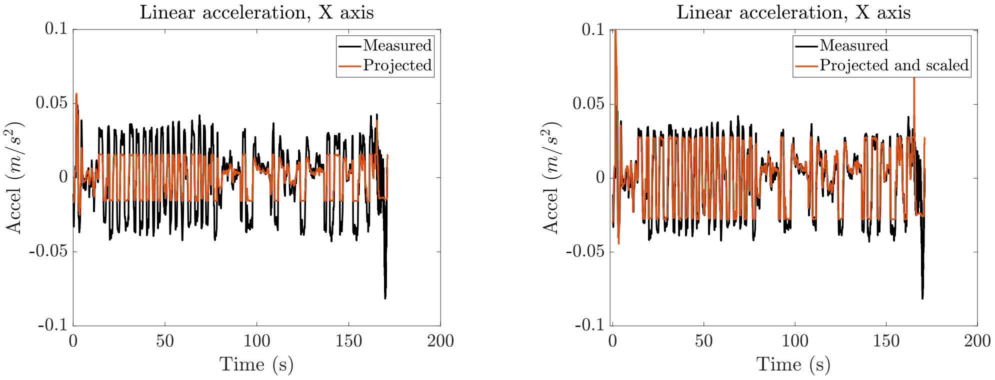


Figure 6.17: Accelerations published by Astrobee's localization algorithm after removing the bias from the IMU, and projected accelerations (obtained by dividing applied force by the true value of Astrobee's mass). This scaling effect was observed only on the linear accelerations. On the right, the same plot after finding the scaling factor.

Table 6.5: Scaling factors per axis for projected linear accelerations to match those given by Astrobee's localization algorithm. While the reason for these stray linear acceleration effects is being investigated, the parameter estimation algorithm treats this as a known linear model.

θ'_x	θ'_y	θ'_z
1.774	1.0201	1.2088

Linear least squares was used to model the "mass", m'_i per axis, where $i \in \{x, y, z\}$. The resulting scaling factors per axis are given in table 6.5. Measured linear accelerations recorded over multiple on-orbit experiments were stacked per axis, resulting in vectors $\mathbf{a}_i$, where $i \in \{x, y, z\}$. Similarly, the force vectors over multiple experiments are collected as $\mathbf{f}_i$. The known true mass of Astrobee is denoted as m, $m = 9.58kg$. Let $\theta = 1/m$ and $\theta' = 1/m'$.

$$\theta_i'^* = argmin||\mathbf{f}_i(\theta + \theta_i') - \mathbf{a}_i||_2^2 \tag{6.12}$$

$$\theta_i'^* = (\mathbf{f}_i^T \mathbf{f}_i)^{-1} \mathbf{f}_i^T a_i - \theta \tag{6.13}$$

Additionally, analysis of the measured and projected accelerations also reveals a latency between them, as shown by the dashed line in fig. 6.18. This is likely due to Astrobee's localization algorithm which filters and removes bias from IMU measurements, as well as uses graph-based localization to obtain state estimates, publishing all robot states together to a ROS topic. Using cross correlation between the two signals, the delay was found to be $0.16s$. The projected liner accelerations after accounting for the delay and the scaling are shown in fig. 6.18.

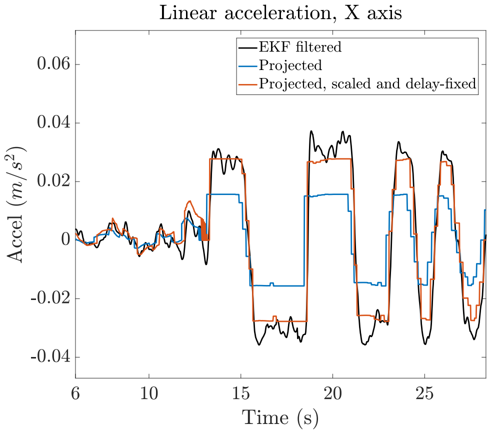


Figure 6.18: Results of the processing steps on Astrobee's measured accelerations - The projected linear accelerations without processing are shown by the blue line, while the projected accelerations post processing, i.e, after accounting for scaling and the delay of the localization algorithm are shown in the orange lines.

6.5.3 Post-saturation forces and torques

The inertial parameter estimation algorithm uses as inputs the forces and torques and the measurement data. As actuation input saturation is expected in dynamical systems, the applied or resulting actuation might not always match the commanded actuation. Therefore, the knowledge of the post-saturation input forces and torques is crucial for accurate parameter estimates.

As described in sec. 6.3, Astrobee's FAM sends out the impeller speed and allowable nozzle opening angles to the Propulsion Module Controller. As the saturation is imposed on the flap rotation that controls the opening and *not* on the commanded forces and torques, an inverse mixing operation is conducted to compute the actuation post-saturation. Steps used by the FAM in the forward-mixing operation (commanded thrust to servo actuation for nozzle opening), and those implemented during the development of this thesis for retrieving the post-saturation forces and torques from nozzle angles (called the inverse-mixing operation) are detailed in this section.

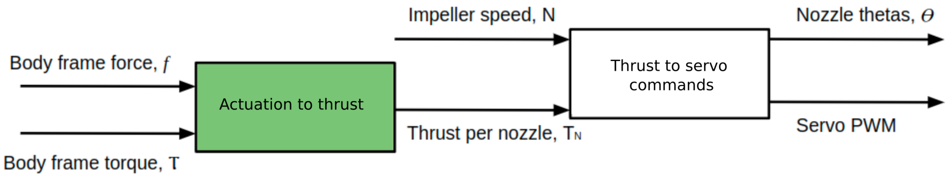


Figure 6.19: A simplified block diagram illustrating the series of operations carried out by the FAM. Block *Actuation to thrust* converts the forces and torques in body frame to the thrust per nozzle given in (6.14), while block *Thrust to servo commands* deals with the underlying physics to yield the nozzle opening angles and the servo PWM signals required to generate the commanded thrust. The operations in block A are known and can be inverted, while inverting Block B, which uses a lookup table, needed further analysis.

The **forward force allocation** operation takes place as follows. A description of the symbols used is given in table 6.6 in the order of appearance.

- In block A, the commanded forces and torques in robot body frame, f and $\tau \in \mathbb{R}^3$ are converted to thrust per nozzle $\underline{T_N} \in \mathbb{R}^{12}$. Note that T_N may have a negative thrust value, as indicated by the underline.

  $$\underline{T_N} = \begin{bmatrix} D & (R \times D) \end{bmatrix}^{\dagger} \begin{bmatrix} f \\ \tau \end{bmatrix} \tag{6.14}$$

  where $\dagger$ denotes the Moore-Penrose pseudoinverse, and $(... \times ...)$ denotes the horizontal stacking of the matrix after computing the cross product of corresponding columns of R and D. Note that D is 3×12, where the columns contain the direction of each of the 12 nozzles. For instance, $\begin{bmatrix} 0 & 0 & -1 \end{bmatrix}^T$ indicates that the nozzle is pointing towards the negative z direction. Similarly, R, the result of subtraction of two 3×12 matrices, is also 3×12. The minimum value below zero of T_N is added to it to prevent negative thrust from being commanded. That is, if $T_{N_i} < 0$, $T_N = \underline{T_N} + \min(|\underline{T_N}|)$.

- Block B accounts for the impeller and nozzle air flow dynamics to convert the commanded thrust per nozzle to nozzle opening angles. Fig.6.20 shows the airflow into the plenum through the impeller, and its precise exhaust through the nozzles, creating thrust. The impeller speed N is assigned based on the flight mode chosen at the time of operation. There are four possible preset values, from 0 to 293.12 RPM.

  For each set of six nozzles connected to a single impeller, (i = 1 to 6 and 7 to 12), the following operations are carried out:

Symbol	Known constant	Description
D_i	x	Direction of each nozzle
R_i	x	Position of each nozzle - position of center of gravity
N	x	Impeller speed
d	x	Impeller diameter
h	x	Impeller height
ρ	x	Air density
C_{d_i}	x	Nozzle Discharge coefficient, determined through testing
C_{dp}		Pressure output coefficient
A_i		Commanded area per nozzle
S_i		Commanded nozzle opening
W	x	Nozzle width
H	x	Nozzle intake height
L	x	Flap length
θ_i		Commanded opening angle

Table 6.6: Description of the symbols used in explanation of the FAM, indicating the known parameters. Subscript i denotes that the quantity is defined or calculated for each nozzle, $i \in [1, ..., 12]$. The geometric parameters are indicated in figs. 6.20 and 6.21.

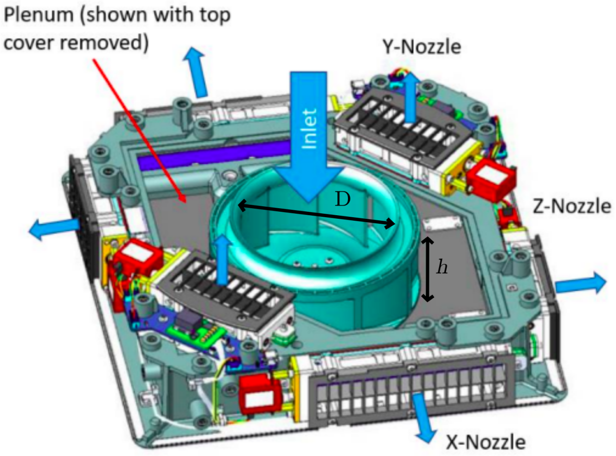


Figure 6.20: Astrobee propulsion module (plenum and 6 nozzles) showing airflow. Thrust is produced by varying the nozzle opening angle for a constant impeller speed. Image credit: [167]

1. The plenum rise in pressure, ΔP, is calculated from the corresponding total thrust, T_N, using the following sub-steps:
   (a) The total thrust per nozzle (connected to one impeller), normed by the discharge coefficient and divided by impeller speed square, $\frac{\sum_i \frac{T_{N_i}}{C_{d_i}}}{N^2}$, is mapped to C_{dp} using a lookup table (more on this in section 6.5.3).
   (b) C_{dp} is then used to calculate the pressure rise, ΔP, in the plenum created by impeller action

$$\Delta P = \rho C_{dp} N^2 d^2 \tag{6.15}$$

2. Recall that Astrobee generates precise thrust by opening the nozzles following the rise in pressure. Therefore, for each nozzle connected to the impeller, the commanded opening area to generate the required thrust is found as

$$A_i = \frac{T_{N_i}}{2\Delta P C_{d_i}^2} \tag{6.16}$$

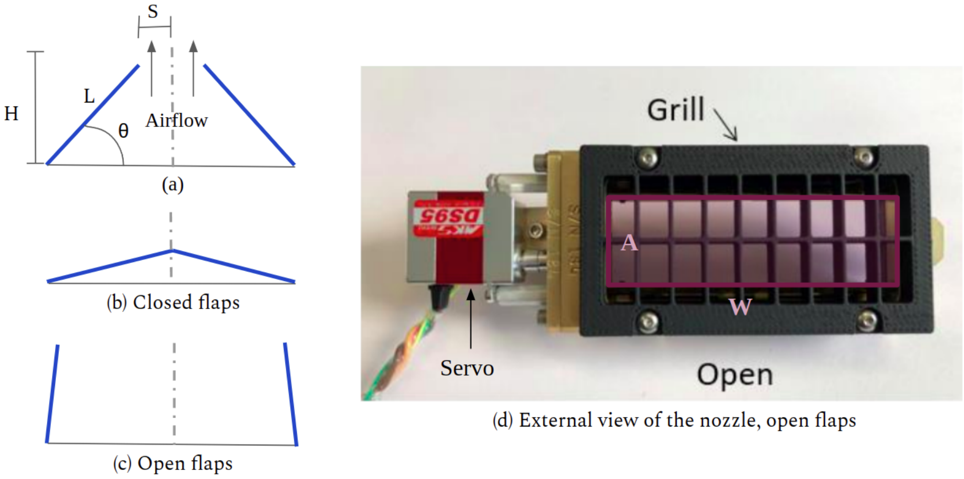
3. This is followed by the commanded nozzle opening per flap. As both flaps of the nozzle will be opened the same amount, the commanded nozzle opening is calculated for one flap by dividing the area by 2 (fig. 6.21 illustrates the nozzle opening mechanism).

$$S_i = \frac{A_i}{2W} \tag{6.17}$$

4. Finally, the commanded nozzle opening angle is calculated

$$cos(\vartheta_i) = \frac{H - S_i}{L} \tag{6.18}$$

The maximum allowed nozzle opening is about $79.91°$, and the lowest is $15.68°$. After applying the saturation function, $\vartheta_i = max\{\vartheta_{min}, min\{\vartheta_i, \vartheta_{max}\}\}$, the commanded angles are mapped on a scale of 0 to 255 and sent off to be converted to raw servo commands.



(a)

(b) Closed flaps

(c) Open flaps

(d) External view of the nozzle, open flaps

Figure 6.21: Airflow through the flaps, shown in blue. Physical parameters of the flaps (a), the flaps closed (b) and open all the way through (c), and (d) the exterior of the nozzle showing nozzle width, as well as the commanded opening area with the flaps completely open. Figures (a)-(c) represent a side view of the flaps, as seen from the direction of the servo. Figure (d) adapted from [167].

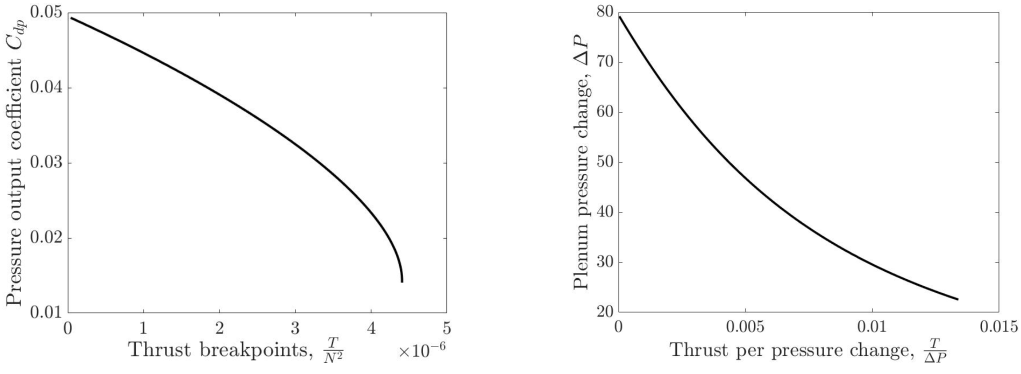
The lookup table

Since the saturation check takes place after the nozzle angles are calculated, finding the post-saturation forces and torques requires inverting the FAM series of operations. The principal challenge here is posed by eq. 6.16 - both T_{N_i} and ΔP are unknown, and T_{N_i} is the quantity needed to find f and τ by inverting block A. Further, ΔP for each impeller depends on C_{dp}, which is the result of a look up depending on $\sum_i \frac{T_{N_i}}{C_{d_i}}$, denoted as T for brevity. This lookup table is generated from the impeller delta pressure to flow rate curve[5].

To summarize the lookup generation, for each impeller, a range of flow rate coefficients, C_q, and the output C_p, which is a polynomial in Cq, is generated. Using C_q and C_p, the total thrust divided by N^2 (square of the impeller speed) is found using the following relation which follows from fluid mechanics:

$$\frac{T}{N^2} = \rho C_q d^3 h \sqrt{2Cp} - 2\rho d^2 A_0 C_p \tag{6.19}$$

The first component in the equation calculates the thrust, while the second accounts for the leakage; A_0 is the zero thrust area. The lookup table is formed by taking select thrust breakpoints, and the corresponding pressure output Cp coefficient (referred to as C_{dp}). A plot of the lookup table is given in fig.6.22(a).



(a) (b)

Figure 6.22: Visualization of total thrust divided by impeller speed $\frac{T}{N^2}$ and the corresponding Cdp values (L), and the lookup table from $\frac{T}{\Delta P}$ to ΔP used for the inverse operation.

Finding the lookup from $\frac{T}{\Delta P}$ to ΔP

As mentioned earlier, the main challenge in inverting the FAM is (6.16), where T_{N_i} and ΔP are both unknown. This is resolved in the inverse FAM by using the lookup table from $\frac{T}{N^2}$ to C_{dp} to obtain the lookup from $\frac{T}{\Delta P}$ to ΔP. This is done using the following steps:

- From the relation in (6.15), ΔP can be computed from Cp as $\Delta P = \rho C_{dp} N^2 d^2$. Using this in (6.19)

---

[5] https://github.com/nasa/astrobee/blob/master/gnc/pmc/src/fam.cc

gives,

$$T = C_q N d^2 h \sqrt{2\rho \Delta P} - 2A_0 \Delta P \tag{6.20}$$

Therefore, the input points of the lookup table $\frac{T}{N^2}$ are converted to T by mutliplying with N^2, and the output points Cp are converted to ΔP by using (6.15). This gives us a lookup from T to ΔP.

- Since the relation between the total thrust and the corresponding C_{dp} values is monotonic (fig. 6.22(a)), the thrust breakpoints T are divided by ΔP to obtain the lookup from $\frac{T}{\Delta P}$ to ΔP, shown in fig. 6.22(b)

Inverting the FAM

For performing the reverse mixing operation, the commanded post-saturation nozzle angles θ_i are first converted to commanded area per nozzle, A_i using eqs. 6.17 and 6.18. Then, using eq. 6.16, the following quantity is found.

$$\frac{\sum_{i=1}^{6} \frac{T_{Ni}}{C_{di}}}{\Delta P} = \sum_{i=1}^{6} 2A_i C_{di} \tag{6.21}$$

and similarly for the second impeller, with nozzles $i = 7...12$. Next, the new lookup table is used to find ΔP from $\frac{T}{\Delta P}$, followed by calculation of the thrust per nozzle from eq. 6.16 as given below, and inversion of block A.

$$T_{Ni} = 2A_i \Delta P C_{di}^2 \tag{6.22}$$

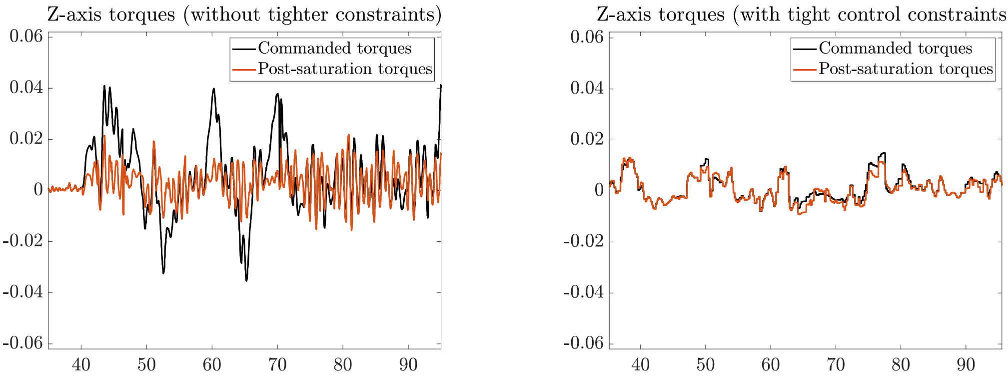


Figure 6.23: Post-saturation (applied) Z- axis torques found after inverting the FAM operation for information-aware trajectories executed in microgravity. The saturation effect was first discovered in a proof-of-concept test session (L), while using tight control constraints to avoid localization loss mitigated some of the saturation effects during full RATTLE demonstration session (R).

The post-saturation force and torque values are valuable to obtain accurate parameter estimates during the test and improvement phase of RATTLE. However, the conservative force and torque bounds (given in table 6.7) used to increase robustness against localization loss (discussed in 6.6.1) majorly reduced the occurrences of saturation and discrepancies between the commanded and applied wrenches (fig. 6.23). Therefore, supplying tight actuation bounds via a controller like the MPC could mitigate some of the saturation effects, and based on the application, even eliminate the need for doing the FAM inversion step.

Table 6.7: Limits used for the control force and torque during the microgravity experiments. Conservative values are used by the controller in order to avoid loss of visual features caused by rapid acceleration.

Limits	Quantity	Bounds
Planner force (N)	f_x, f_y, f_z	$[-0.1, 0.1]$
Planner torque (Nm)	τ_x, τ_y, τ_z	$[-0.05, 0.05]$
Controller force (N)	f_x, f_y, f_z	$[-0.452, 0.452], [-0.216, 0.216], [-0.257, 0.257]$
Controller torque (Nm)	τ_x, τ_y, τ_z	$[-0.045, 0.045], [-0.022, 0.022], [-0.026, 0.026]$

6.5.4 Updates to the estimation algorithm

After accounting for the factors described above, a **sequential least squares** algorithm was used to test the estimation accuracy on data gathered from RATTLE's microgravity tests. Linear and angular accelerations, $a_i, \alpha_i, where i \in \{x, y, z\}$, were used as measurements, with $\alpha_i = \dot{\omega}_i$, where ω represents angular velocities measured by the IMU. Note that Astrobee's localization algorithm uses low-pass filters on the acceleration and angular velocity measurements. The acceleration measurements are arranged as $\tilde{\mathbf{a}} = [a_x, a_y, a_z, \alpha_x, \alpha_y, \alpha_z]^T$. The input forces and torques are denoted as f_i and τ_i respectively. Quantities that are inverse of the mass, m and the principal moments of inertia I_i are estimated, formulating the problem as

$$\tilde{\mathbf{a}} = H\Theta + v, \qquad v \sim \mathcal{N}(0, R) \tag{6.23}$$

where v is the measurement noise

$$\Theta = \begin{bmatrix} \frac{1}{m} & \frac{1}{I_{xx}} & \frac{1}{I_{yy}} & \frac{1}{I_{zz}} \end{bmatrix}^T \tag{6.24}$$

and

$$H = \begin{bmatrix} f_x & 0 & 0 & 0 \\ f_y & 0 & 0 & 0 \\ f_z & 0 & 0 & 0 \\ 0 & \tau_x & 0 & 0 \\ 0 & 0 & \tau_y & 0 \\ 0 & 0 & 0 & \tau_z \end{bmatrix} \tag{6.25}$$

The covariance recursive form of the least squares algorithm [168] is used to sequentially solve (6.23), given as:

$$\hat{\Theta}_{k+1} = \hat{\Theta}_k + K_{k+1}(\tilde{\mathbf{a}} - H_{k+1}\hat{\Theta}_k) \tag{6.26}$$

where

$$K_{k+1} = PH_{k+1}^{T}\left[H_{k+1}P_{k}H_{k+1}^{T} + R\right]^{-1} \tag{6.27}$$

$$P_{k+1} = \left[I - K_{k+1}H_{k+1}\right]P_{k} \tag{6.28}$$

where P denotes the estimate covariance, and W is the measurement covariance. (6.26) is the Kalman update equation, with K being the Kalman gain. Note that the $\omega \times I\omega$ terms were neglected in this case, as their impact measured only up to the order of 10^{-4}. However, the linear formulation of eq. 6.23 can be extended to include those terms, as well as estimate other dynamic parameters such as center of mass offsets.

Outlier rejection

Owing to its minimization of the L_2 norm of the residuals, the least squares estimation is particularly sensitive to outliers, which can skew the fit. In the case of the RATTLE micro-gravity experiments, sudden jumps in Astrobee's localization were a key reason for the loss of accuracy in the estimation procedure. Fig. 6.24 shows one such localization outlier as seen in the angular acceleration. Note that the angular accelerations were computed by numerical differentiation of the angular velocity data, resulting in amplification of the measurement noise.

The Chi squares test [169] is a commonly used metric for checking the health of the Kalman filter. Following the Gaussian error assumptions made during the filter derivation, this test can help determine the filter output's validity. In the recursive least squares (or Kalman filter), (6.26), the innovation, or the difference between the observed and the predicted measurements are given as

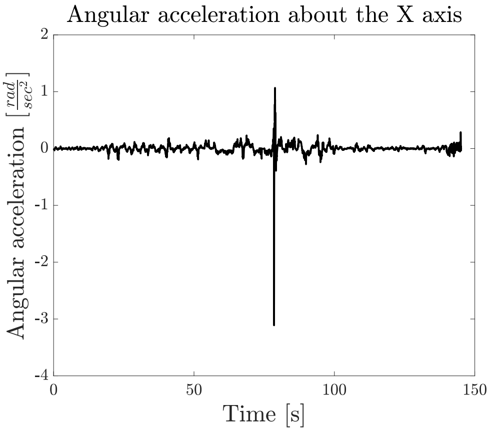


Figure 6.24: Angular acceleration showing localization outliers during a RATTLE micro-gravity run. Apart from the jumps seen in the plot at 78 seconds, some other localization difts can be seen. Fast motions that cause a loss of visual features, as well as external disturbance forces are thought to cause these effects.

$$\gamma_k = \tilde{\mathbf{a}}_k - H_k\hat{\Theta}_{k-1} \tag{6.29}$$

and the innovation covariance, $E\left[\gamma_k, \gamma_k^T\right]$, is

$$S_k = H_k P_{k-1} H_k^T + R \tag{6.30}$$

If the assumption of randomly distributed Gaussian measurement noise from (6.23) holds, then the normalized sum of innovations should belong to a Chi-squared distribution with l degrees of freedom, where l is the number of independently sampled normal distributions. In this case, l is equal to the number of measurements, $l = 6$, as they are considered independent and identically distributed.

$$\gamma_k^T S_k^{-1} \gamma_k \in \chi_l^2 \tag{6.31}$$

Alternatively, this is equivalent to the Mahalanobis distance between the expected measurement based on the previous estimate and the actual measurement [170]. Based on the critical value of the χ_l^2 distribution at the desired confidence level, a probabilistic threshold can thus be set for $\gamma_k^T S_k^{-1} \gamma_k$ to determine measurement reliability. In this case, the value corresponding to the probability level of 0.99 is chosen as the threshold ν, i.e., $P(\gamma_k^T S_k^{-1} \gamma_k \leq \nu) = 0.99$. If the result of the test was higher than a threshold $\gamma_k^T S_k^{-1} \gamma_k > \nu$, then the measurement was rejected and the update steps 6.26 and 6.27 were not performed.

6.5.5 Final estimation results

The recursive least squares estimation algorithm was used on replayed microgravity data, with a total of 11 ROS bags of experimental data, and the estimation results are presented in figs. 6.25 and 6.26. As described in 6.4, the experiments began with a high-uncertainty distribution of the inertial parameters, intending to improve the parameter estimates as the robot tracks a path to the final position. The tests are divided into three categories:

1. without information gain, where the robot tracks a trajectory to the goal point without any explicit information weighting in the local plan
2. with information gain, where the RATTLE algorithm is used, and the local planner explicity incorporates informative motions in the plan en route to the goal
3. information gain with parameter updates, which is category (2) with the addition of parameter updates, i.e., the local planner and controller models are updated based on the most recent estimates.

At testing time, however, the parameter estimates were not correctly incorporated in the planning and control models. This caused the robot to drift off course and at times resulted in the loss of localization. The reason for this behaviour, most a bug in the code that caused resetting of the parameters to their initial uncertain values, was yet to be ascertained at the time of writing. Due to this unexpected behaviour, tests with parameter updates have been grouped under a separate category.

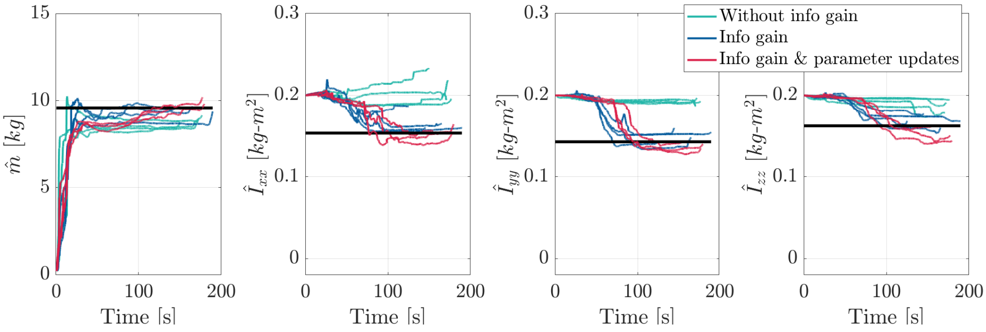


Figure 6.25: Evolution of the parameter estimates for mass and principal moments of inertias. These results were obtained by using data gathered from microgravity experiments.

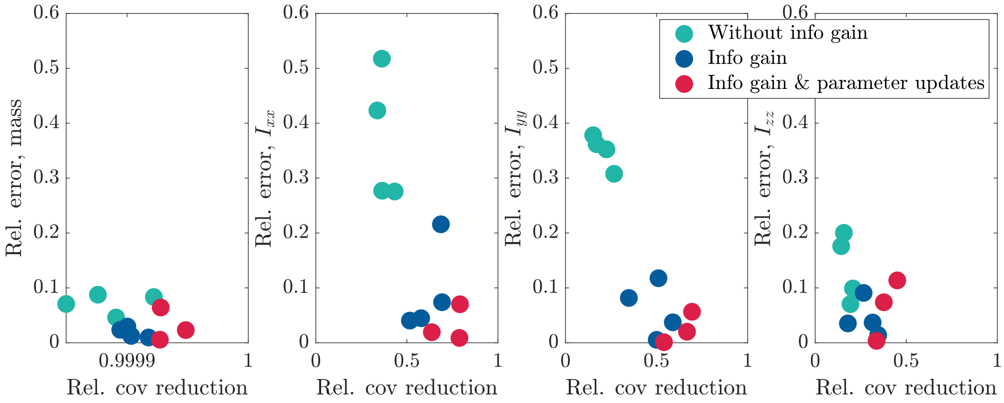


Figure 6.26: Estimate error and covariance reduction for mass and principal moments of inertia parameter estimation. These results were obtained by using data gathered from microgravity experiments.

The results from fig. 6.25 show that even the trajectories without the information gain were able to get mass estimates close to their true values. This was expected, as the robot performs linear motion throughout the trajectory, resulting in mass parameter excitation without explicit emphasis. Fig. 6.26 shows a high estimate covariance reduction for the mass parameter for all three cases, with a marginally greater increase in accuracy for the info gain case. For inertia parameters I_{xx} and I_{yy}, both figures show a clear decrease in error and in final estimate covariance for the category with information awareness. While the non-information aware case (test category 1) does not provide enough excitation for the estimates to be learned, learning continues for the information-aware case, leading to estimate convergence. Tests in all categories required a final yaw rotation for the robot to arrive to the goal state. This is visible in the sequential estimation plot for I_{zz}, where the final rotation provides bonus excitation for parameter estimation, specially for the non info case. This is partly the reason why although information gain does not seem to be as effective for I_{zz} as the other two moments of inertias. Further investigation is needed to determine other reasons for this behaviour. In sum, RATTLE's information-aware motion led to more accurate estimates and a higher covariance reduction when compared to the trajectories without info gain. Hardware experimental results show that this effect was more pronounced on the moments of inertias, more specifically, on the I_{xx} and I_{yy} parameters.

6.6 Discussion

6.6.1 Localization

Fig. 6.27 shows the difference in localization between the informative and non-informative trajectories for which the results are shown in fig. 6.15. The information-aware trajectory has worse tracking than the ”regular” trajectory which solely performed state error minimization without information seeking maneuvers. This result seems counter-intuitive, given that the objective of RATTLE was to facilitate parameter estimation for improved tracking. AstroLoc [161], Astrobee’s localization system at the time of writing this thesis, performs visual-inertial graph-based optimization and the estimation of IMU biases. Occlusions due to objects such as cargo bags, lighting conditions, and the inability to find landmarks can cause a heavy reliance on visual-inertial odometry. AstroLoc is more robust to large drifts and lost robot events than Astrobee’s previous augmented-state EKF localizer due to features like covariance sanity checks, providing fallbacks in case of cheirality errors for 3D map features and visual odometry factors. As Astrobee’s localization is under active development, special care in the motion constraints was taken (see table 6.7) to avoid jumps in localization, which would be a major challenge for the controller to handle. However, the rotation induced by the information-aware local plans still resulted in localization drifts due to feature losses and mismatch, which the figure shows. Therefore, Astrobee’s ability to track efficiently was adversely affected in the information-aware tests. Enforcing tighter motion constraints and evaluating their effect on the localization and computation speed of the planner and controller components should be evaluated in order to demonstrate the full potential of RATTLE.

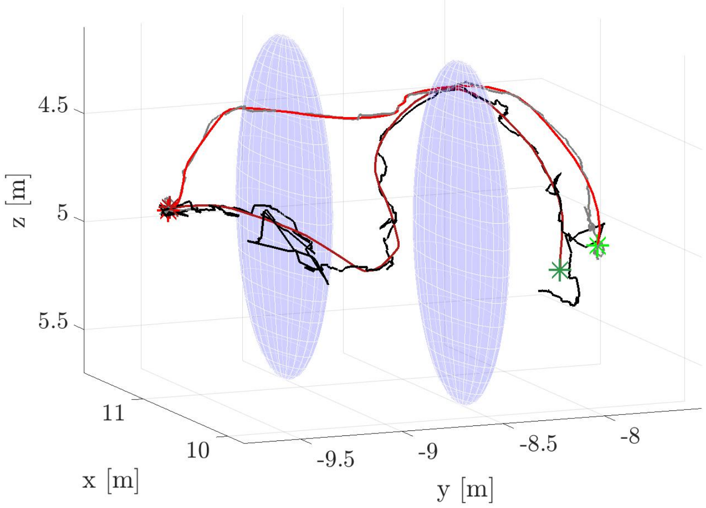


Figure 6.27: Planned and executed trajectories in microgravity. The red lines show the global plan, while the black/gray show the executed trajectories. Start point is denoted by red and end point by green. The ovals denote two simulated obstacles. The dark shades correpond to the information aware/RATTLE trajectories, while the others denote the regular trajectories without information weighting.

6.6.2 Software contributions

Multiple software contributions of this thesis resulted from integrating and testing the RATTLE framework on the Astrobee test platform, which is still maturing in its autonomy capabilities.

- The steps and challenges involved in integration of RATTLE with the Astrobee FSW and eventual deployment on Astrobee hardware have been outlined in A Brief Guide To Astrobee [162]
- The author and collaborators intend to release the RATTLE source code to the public in the near future
- Issues affecting the accuracy of the parameter estimates such as force and torque saturation and acceleration scaling have been shared with NASA, as well as other members of the Astrobee community. For instance, the *inverse_fam* module, programmed by the author to calculate post-saturation forces and torques, was used in micro-gravity demonstrations of an outlier rejection EKF approach for robust localization, developed by groups at the German Aerospace Center (DLR) and the Massachusetts Institute of Technology (MIT)
- Joint work performed leading up to the microgravity test sessions resulted in code packages publicly released by collaborators for researchers new to Astrobee such as Astrobee ROS Guest Science Demo and ASAP Astrobee Science Application Package

6.6.3 Future work

The RATTLE framework's objective is to perform a robust and adaptive online reduction of parametric uncertainty via information-aware trajectories and to leverage the latest parameter knowledge for improved tracking in safety- and precision- critical applications. The RATTLE framework was integrated with NASA's Astrobee free-flyer software stack and deployed on the International Space Station on Astrobee's resource-constrained hardware. RATTLE's information-awareness and covariance-based excitation shut-off led to improved estimation accuracy in real-time compared to a non-informative trajectory. However, RATTLE's effect on tracking performance could not be demonstrated as the informative rotational motions resulted in localization drifts on Astrobee. Additionally, to implement RATTLE on hardware, simplifications such as linearity assumptions for the tube MPC, translation-only global planning, and estimation of four inertial parameters were used. To extend RATTLE to more complex and uncertain systems, re-evaluation of these assumptions is necessary. Further, propagation of parameteric uncertainty through the system dynamics in place of considering it as an additive disturbance to generate uncertainty bounds could be of interest.

Moreover, speed vs. planning horizon trade-offs for planning and control with a higher number of states and parameters of interest will be needed. Future work thus involves studying the effect of parameter updates on tracking performance and extending RATTLE to dynamic systems with a larger state space, e.g., free-flyer dynamics. In the case of Astrobee, some scaling and saturation effects initially prevented the accurate estimation of parameters. Therefore, incorporating this information in the planner and studying RATTLE's performance in the face of non-observable parameters and external, unmodeled effects like drift are some interesting future directions.

Acknowledgement The development of the RATTLE approach was carried out in collaboration with Keenan Albee, Ph.D. student at the Massachusetts Institute of Technology's Space System's Lab (SSL), as were publications [61–64] and [162]. Keenan Albee developed the RRT* global planning module and the robust tube MPC, and the author implemented the parameter estimators, and the planning of information-rich trajectories (fig. 6.2 illustrates the components of RATTLE). The mid-level planer as well as the consolidation of code for ground and micro-gravity testing was a result of joint effort, building off of Keenan Albee's experience with Astrobee. The author also developed excitation trajectories and a maximum likelihood parameter estimator for a research article on Uncertainty aware motion planning for on-orbit assembly [171], published in collaboration with Bryce Doerr, a postdoctoral associate at the Massachusetts Institute of Tecnhnology's SSL and Astrodynamics, Space Robotics, and Controls Lab (ARCLab), and Keenan Albee. Additionally, this thesis benefited from discussions on Model Predictive Control and C++ Astrobee integration with Pedro Roque, Ph.D. student at the KTH institute of Technology's division of Decision and Control Systems (DCS). Inputs from Prof. David Miller and Prof. Richard Linares were important for the success of this work. The author thanks them for this fruitful collaboration. This collaboration was enabled by seed projects under the MIT Portugal program, of which the author was a part of from 2018-2022. The author gratefully acknowledges these sponsors.

Chapter 7

Conclusions

This thesis addressed the problem of ensuring safe and precise trajectory execution in an environment rife with uncertainty, which is a crucial On-Orbit Servicing (OOS) challenge. Interaction with payloads and other spacecraft, fuel use, and reconfiguration changes the system's inertial properties, necessitating re-calibration. If used in model-based planning and control, inaccurate dynamics interfere with safe and precise task operation and often hinder optimal planning and execution. Therefore, this thesis focused on reducing inertial uncertainty in free-flyer operations, explicitly considering a free-flyer cargo transportation scenario.

If estimating the uncertain parameters is of interest, upfront system identification can be performed and the accurate estimates incorporated at the time of planning and control. For this, most free-flyer parameter identification approaches in literature perform offline batch estimation, i.e., gathering data by executing excitation trajectories before using it for estimation. In contrast, if the goal is successful task execution without expending resources on identification, robust or direct adaptive control approaches can be used to reject disturbances caused due to uncertainty. With the primary objective of ensuring stability and reducing tracking error, disturbance rejection approaches do not focus on parameter learning. Instead, they often assume bounded uncertainties, and can exhibit conservative behaviour as robustness is calculated by accounting for the worst-case performance.

Two inertial parameter identification methods were studied in this thesis to minimize the deleterious effects of noisy acceleration measurements on inertial estimation accuracy. While the first approach leverages the continuous nature and periodicity of the Fourier Series for analytical differentiation and improved filtration of measurement data in batch, the second approach benefits from the symplectic nature of discrete variational integration to design a sequential rigid body estimation problem by representing the states as dual quaternions without the need for acceleration measurements.

Also relevant to this thesis is a class of approaches called simultaneous learning which finds the middle ground between system identification and disturbance rejection. The goal is to perform learning on the go, gathering information to inform planning or control behavior. Active learning is widely used in simultaneous learning approaches, where informative motion plans are executed to promote estimate observability and accuracy. Some methods under simultaneous active learning solve an approximation of the stochastic optimization problem, finding a control policy that is inherently information seeking. While they account for stochasticity in the dynamics and can optimally trade-off exploration (information-seeking motion) and exploitation (using gathered information for improved navigation), they are computationally expensive and scalability to complex dynamical systems remains a challenge. On the other hand, active learning approaches, which explicitly induce information-seeking or covariance-reducing behavior, have been used for pure system identification or landmark visibility during localization. Online active learning has been used for inertia property estimation in [172], but online incorporation of the estimates has not been addressed.

A real-time and scalable method amenable to resource-constrained hardware is of interest for the on-orbit precise cargo transportation scenario. This thesis proposed RATTLE, an information-aware motion planning framework for parameter learning. RATTLE comes with a global planner, an information-aware local planner which balances state error to the goal and information maximization, a robust model predictive controller to calculate control commands and provide robustness against transients, and an inertial parameter estimator. As a result of incorporating the latest estimates in the planner and controller models, RATTLE is able to leverage the recent model knowledge for precise tracking.

Limited accessibility to micro-gravity research platforms being a barrier, only a handful of parameter identification approaches have been tested on-orbit or using micro-gravity data, relying primarily on simulation results. RATTLE was tested on NASA's Astrobee free-flyer on board the ISS, demonstrating its adaptive, robust, real-time capabilities on 6DOF hardware systems with limited computational resources. To the author's knowledge, this is the first time that an information-aware planning framework was demonstrated in space and one of the first experiments to make significant changes to Astrobee's flight stack. By balancing exploration and exploitation in a safe, scalable, and real-time manner, RATTLE is a step towards addressing the simultaneous learning problem for free-flying space robots and mobile robotics in general.

7.1 Summary of Contributions

A summary of the contributions resulting from the work devloped in this thesis is as follows:

- To address the issue of obtaining noise-free acceleration measurements in inertial parameter estimation based on the equations of motion, a batch inertial parameter estimation approach adapted from system identification of ground robots [75] was proposed. The resulting approach, via filteration of high-frequency harmonics and analytical differentiation, avoids noise-inducing integration or differentiation on measurements and yields accurate estimates in simulation experiments with noise.

- An alternative rigid-body inertial estimation approach to the issue of noisy acceleration measurement that avoids the differentiation of the measured data entirely was proposed. Extending work done on rotational inertia estimation using discrete variational mechanics in [103], a sequential complete rigid-body inertial parameter estimation approach based on dual quaternion variational integration was developed. simulation results showed at least one order of magnitude increase in estimate accuracy compared to the finite difference discretization.

- To perform robust parametric uncertainty reduction *during* the execution of other control tasks, RATTLE[1], a novel motion planning algorithm for creating information-aware plans with online parameter learning, was proposed. RATTLE was demonstrated for a microgravity inertial uncertainty scenario on NASA's Astrobee, a 6DOF holonomic freeflyer, considering rigid body Newton-Euler dynamics with thirteen states and four inertial unknowns (mass and moments of inertia). Open source contributions include a guide on integrating guest science GNC stack with Astrobee's flight software and the soon-to-be public RATTLE source code.

7.2 Future Work

The parameter estimation approach aided by information-aware trajectory design to reduce inertial uncertainty adapted in this thesis leads to multiple avenues for future work. This section presents recommendations for extensions to the developed algorithms and directions for future research based on the findings of this thesis.

The proposed estimation methods assume rigid-body dynamics - a reasonable assumption for the Astrobee free-flyer with its arm stowed. However, flexibilities and sloshing effects are common occurrences in spacecraft, for instance, in free-flying manipulator systems. As works such as [107] and [44] have shown, these dynamics, when represented in the canonical equations of motion can be posed as an estimation problem linear in the parameters of interest given in (2.19). Therefore, extending the methods proposed in this thesis to accommodate models with complex dynamics and a more extensive set of unknown parameters is an exciting avenue for future work. Alternatively, a growing field of model learning that combines deep learning-based black box approaches with physical priors for better generalization and data-efficiency could be of interest here. One such end-to-end learning approach for separable mechanical systems using priors derived from discrete variational mechanics was proposed in [173]. Further, the model-based estimation approach often assumes known system dynamics without accounting for other sources of uncertainty. Employing model-free techniques to supplement the parametric estimation, for instance, Gaussian Process Regression to estimate the residual uncertainty [35], could help in this regard.

Explicit active learning or the generation of reference trajectories that facilitate estimation by explicitly producing information-rich motion, is a central element in this thesis. RATTLE leverages the flexibility of this approach to control the information content, tuning the weighting on parameters of interest and ceasing informative motion altogether as needed. As the complexity of the dynamics increases, and with it the number of parameters, it might become intractable to tune the information content for each parameter. Further, calculating the information metric, for instance, the Fisher Information Matrix (FIM) for highly coupled parameters, would become computationally intensive, posing a challenge to real-time

---

[1] **R**eal-time information-**A**ware **T**argeted **T**rajectory planning for **L**earning via **E**stimation

computation. Therefore, future research could focus on the scalability of information-aware learning, in particular, by identifying and emphasizing the parameters relevant to the task at hand.

Data from the microgravity experiments of the RATTLE algorithm revealed the need to process the measurement data prior to its use for parameter estimation. It would thus be interesting to perform an online validation of the complete estimation pipeline, including the data processing steps like outlier rejection and accounting for the acceleration scaling. Further, evaluating the improvement of online tracking performance after incorporating recent parameter estimates would demonstrate RATTLE's full potential. Simplifying assumptions that enabled the real-time implementation of RATTLE included: the use of translational dynamics for the global planner, considering a linear model for the robust tube MPC, and neglecting the center of mass offset to bring down the estimate set to four parameters (mass and principal moments of inertia). Evaluating these assumptions while considering the available computational resources would be critical when applying RATTLE to complex, non-linear dynamical systems.

Bibliography

[1] Y. Gao and S. Chien. Review on space robotics: Toward top-level science through space exploration. *Science Robotics*, 2(7):5074, 2017.

[2] K. Yoshida. Achievements in space robotics. *IEEE Robotics & Automation Magazine*, 16(4):20–28, 2009.

[3] L. Pedersen, D. Kortenkamp, D. Wettergreen, I. Nourbakhsh, and D. Korsmeyer. A survey of space robotics. In *ISAIRAS*, 2003.

[4] R. C. Moeller, L. Jandura, K. Rosette, M. Robinson, J. Samuels, M. Silverman, K. Brown, E. Duffy, A. Yazzie, E. Jens, et al. The sampling and caching subsystem (scs) for the scientific exploration of jezero crater by the mars 2020 perseverance rover. *Space Science Reviews*, 217(1):1–43, 2021.

[5] P. Callen. Robotic transfer and interfaces for external ISS payloads. In *3rd Annual ISS Research and Development Conference*, 2014.

[6] A. M. Long, M. G. Richards, and D. E. Hastings. On-orbit servicing: a new value proposition for satellite design and operation. *Journal of Spacecraft and Rockets*, 44(4):964–976, 2007.

[7] A. Ellery, J. Kreisel, and B. Sommer. The case for robotic on-orbit servicing of spacecraft: Spacecraft reliability is a myth. *Acta Astronautica*, 63(5-6):632–648, 2008.

[8] J. H. Saleh, E. Lamassoure, and D. E. Hastings. Space systems flexibility provided by on-orbit servicing: Part 1. *Journal of Spacecraft and Rockets*, 39(4):551–560, 2002.

[9] C. M. Reynerson. Spacecraft modular architecture design for on-orbit servicing. In *2000 IEEE Aerospace Conference. Proceedings (Cat. No. 00TH8484)*, volume 4, pages 227–238. IEEE, 2000.

[10] M. Bualat, J. Barlow, T. Fong, C. Provencher, and T. Smith. Astrobee: Developing a free-flying robot for the international space station. In *AIAA SPACE 2015 Conference and Exposition*, page 4643, 2015.

[11] T. Williams and S. Tanygin. On-orbit engineering tests of the aercam sprint robotic camera vehicle. *Spaceflight mechanics 1998*, pages 1001–1020, 1998.

[12] D. Miller, A. Saenz-Otero, J. Wertz, A. Chen, G. Berkowski, C. Brodel, S. Carlson, D. Carpenter, S. Chen, S. Cheng, et al. Spheres: a testbed for long duration satellite formation flying in micro-gravity conditions. In *Proceedings of the AAS/AIAA space flight mechanics meeting*, volume 105, pages 167–179. Clearwater, Florida, January, 2000.

[13] M. Deans, J. Badger, and T. Smith. Integrated system for autonomous and adaptive caretaking (ISAAC). `https://ntrs.nasa.gov/api/citations/20190029054/downloads/20190029054.pdf`, 2019.

[14] S. Mitani, M. Goto, R. Konomura, Y. Shoji, K. Hagiwara, S. Shigeto, and N. Tanishima. Int-ball: Crew-supportive autonomous mobile camera robot on ISS/jem. In *2019 IEEE Aerospace Conference*, pages 1–15. IEEE, 2019.

[15] E. Hüllermeier and W. Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110(3):457–506, 2021.

[16] Z. Vafa and S. Dubowsky. The kinematics and dynamics of space manipulators: The virtual manipulator approach. *The International Journal of Robotics Research*, 9(4):3–21, 1990.

[17] L. Meirovitch and S. Lim. Maneuvering and control of flexible space robots. *Journal of guidance, control, and dynamics*, 17(3):520–528, 1994.

[18] M. Reyhanoglu and J. R. Hervas. Nonlinear dynamics and control of space vehicles with multiple fuel slosh modes. *Control Engineering Practice*, 20(9):912–918, 2012.

[19] S. A. A. Moosavian and E. Papadopoulos. Free-flying robots in space: an overview of dynamics modeling, planning and control. *Robotica*, 25(5):537, 2007.

[20] K. Yoshida, D. Nenchev, G. Ishigami, and Y. Tsumaki. Space robotics. In *The International Handbook of Space Technology*, pages 541–573. Springer, 2014.

[21] J.-H. Shin and J.-J. Lee. Dynamic control with adaptive identification for free-flying space robots in joint space. *Robotica*, 12(6):541–551, 1994.

[22] O. Ma, H. Dang, and K. Pham. On-orbit identification of inertia properties of spacecraft using a robotic arm. *Journal of guidance, control, and dynamics*, 31(6):1761–1771, 2008.

[23] E. Wilsona, D. W. Suttera, and R. W. Mahb. MCRLS for on-line spacecraft mass-and thruster-property identification. In *Proceedings of the International Association of Science and Technology for Development (IASTED) international conference on intelligent systems and control*, 2004.

[24] W. Rackl, R. Lampariello, and A. Albu-Schäffer. Parameter identification methods for free-floating space robots with direct torque sensing. *IFAC Proceedings Volumes*, 46(19):464–469, 2013.

[25] J. A. Keim, A. B. Acikmese, and J. F. Shields. Spacecraft inertia estimation via constrained least squares. In *2006 IEEE Aerospace Conference*, pages 6–pp. IEEE, 2006.

[26] K. Yoshida and S. Abiko. Inertia parameter identification for a free-flying space robot. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, page 4568, 2002.

[27] Y.-P. Chen and S.-C. Lo. Sliding-mode controller design for spacecraft attitude tracking maneuvers. *IEEE transactions on aerospace and electronic systems*, 29(4):1328–1333, 1993.

[28] J.-U. Park, K.-H. Choi, and S. Lee. Orbital rendezvous using two-step sliding mode control. *Aerospace science and technology*, 3(4):239–245, 1999.

[29] R. Ghosh Dastidar. On the advantages and limitations of sliding mode control for spacecraft. In *AIAA SPACE 2010 Conference & Exposition*, page 8777, 2010.

[30] B. T. Lopez. *Adaptive robust model predictive control for nonlinear systems*. PhD thesis, Massachusetts Institute of Technology, 2019.

[31] C. Buckner and R. Lampariello. Tube-based model predictive control for the approach maneuver of a spacecraft to a free-tumbling target satellite. In *2018 Annual American Control Conference (ACC)*, pages 5690–5697. IEEE, 2018.

[32] J.-F. Shi, S. Ulrich, and A. Allen. Spacecraft adaptive attitude control with application to space station free-flyer robotic capture. In *AIAA Guidance, Navigation, and Control Conference*, page 1780, 2015.

[33] S. Ulrich, A. Saenz-Otero, and I. Barkana. Passivity-based adaptive control of robotic spacecraft for proximity operations under uncertainties. *Journal of Guidance, Control, and Dynamics*, 39(6): 1444–1453, 2016.

[34] Y. Xu, H. Shum, T. Wade, and J. Lee. Parameterization and adaptive control of space robot systems. *IEEE Trans. on Aerospace and Electronics Systems*, 30(2):435–451, 1994.

[35] J. Kabzan, L. Hewing, A. Liniger, and M. N. Zeilinger. Learning-based model predictive control for autonomous racing. *IEEE Robotics and Automation Letters*, 4(4):3363–3370, 2019.

[36] L. Hewing, J. Kabzan, and M. N. Zeilinger. Cautious model predictive control using gaussian process regression. *IEEE Transactions on Control Systems Technology*, 2019.

[37] A. Aswani, H. Gonzalez, S. S. Sastry, and C. Tomlin. Provably safe and robust learning-based model predictive control. *Automatica*, 49(5):1216–1226, 2013.

[38] G. Chowdhary, M. Mühlegg, J. P. How, and F. Holzapfel. Concurrent learning adaptive model predictive control. In *Advances in Aerospace Guidance, Navigation and Control*, pages 29–47. Springer, 2013.

[39] G. Lee, S. S. Srinivasa, and M. T. Mason. GP-ILQG: Data-driven robust optimal control for uncertain nonlinear dynamical systems. *arXiv preprint arXiv:1705.05344*, 2017.

[40] M. Lopes, F. Melo, and L. Montesano. Active learning for reward estimation in inverse reinforcement learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 31–46. Springer, 2009.

[41] D. J. Webb, K. L. Crandall, and J. van den Berg. Online parameter estimation via real-time replanning of continuous gaussian pomdps. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5998–6005. IEEE, 2014.

[42] Y. Murotsu, K. Senda, M. Ozaki, and S. Tsujio. Parameter identification of unknown object handled by free-flying space robot. *Journal of guidance, control, and dynamics*, 17(3):488–494, 1994.

[43] R. Lampariello and G. Hirzinger. Freeflying robots-inertial parameters identification and control strategies. *Advanced Space Technologies for Robotics and Automation (ASTRA), 2000*, 5(7), 2000.

[44] W. Rackl and R. Lampariello. Parameter identification of free-floating robots with flexible appendages and fuel sloshing. In *Proceedings of 2014 International Conference on Modelling, Identification & Control*, pages 129–134. IEEE, 2014.

[45] W. Xu, Z. Hu, Y. Zhang, Z. Wang, and X. Wu. A practical and effective method for identifying the complete inertia parameters of space robots. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5435–5440. IEEE, 2015.

[46] O.-O. Christidi-Loumpasefski, K. Nanos, and E. Papadopoulos. On parameter estimation of space manipulator systems using the angular momentum conservation. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5453–5458. IEEE, 2017.

[47] B. Naveen, S. V. Shah, and A. K. Misra. Momentum model-based minimal parameter identification of a space robot. *Journal of Guidance, Control, and Dynamics*, 42(3):508–523, 2019.

[48] C. J. Ostafew, A. P. Schoellig, T. D. Barfoot, and J. Collier. Learning-based nonlinear model predictive control to improve vision-based mobile robot path tracking. *Journal of Field Robotics*, 33(1):133–152, 2016.

[49] A. D. Wilson, J. A. Schultz, A. R. Ansari, and T. D. Murphey. Real-time trajectory synthesis for information maximization using sequential action control and least-squares estimation. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4935–4940. IEEE, 2015.

[50] P. Slade, P. Culbertson, Z. Sunberg, and M. Kochenderfer. Simultaneous active parameter estimation and control using sampling-based bayesian reinforcement learning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 804–810. IEEE, 2017.

[51] T. C. Nguyen-Huynh and I. Sharf. Adaptive reactionless motion and parameter identification in postcapture of space debris. *Journal of Guidance, Control, and Dynamics*, 36(2):404–414, 2013.

[52] A. T. Espinoza and D. Roascio. Concurrent adaptive control and parameter estimation through composite adaptation using model reference adaptive control/Kalman Filter methods. *1st Annual IEEE Conference on Control Technology and Applications, CCTA 2017*, 2017-January:662–667, 2017. doi: 10.1109/CCTA.2017.8062538.

[53] M. S. De Queiroz, V. Kapila, and Q. Yan. Adaptive nonlinear control of multiple spacecraft formation flying. *Journal of Guidance, Control, and Dynamics*, 23(3):385–390, 2000.

[54] H. Wang and Y. Xie. Prediction error based adaptive jacobian tracking for free-floating space manipulators. *IEEE Transactions on Aerospace and Electronic Systems*, 48(4):3207–3221, 2012.

[55] J.-J. E. Slotine and W. Li. Composite adaptive control of robot manipulators. *Automatica*, 25(4): 509–519, 1989.

[56] O.-O. Christidi-Loumpasefski, G. Rekleitis, and E. Papadopoulos. Concurrent parameter identification and control for free-floating robotic systems during on-orbit servicing. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6014–6020. IEEE, 2020.

[57] M. Ekal and R. Ventura. An energy balance based method for parameter identification of a free-flying robot grasping an unknown object. In *2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 110–116. IEEE, 2018.

[58] M. Ekal and R. Ventura. On inertial parameter estimation of a free-flying robot grasping an unknown object. In *2018 5th International Conference on Control, Decision and Information Technologies (CoDIT)*, pages 815–821. IEEE, 2018.

[59] M. Ekal and R. Ventura. On the accuracy of inertial parameter estimation of a free-flying robot while grasping an object. *Journal of Intelligent & Robotic Systems*, 98(1):153–163, 2020.

[60] M. Ekal and R. Ventura. A dual quaternion-based discrete variational approach for accurate and online inertial parameter estimation in free-flying robots. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6021–6027. IEEE, 2020.

[61] K. Albee, M. Ekal, R. Ventura, and R. Linares. Combining parameter identification and trajectory optimization: Real-time planning for information gain. In *15th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA) 2019*, 2019.

[62] M. Ekal, K. Albee, B. Coltin, R. Ventura, R. Linares, and D. Miller. Online information-aware motion planning with inertial parameter learning for robotic free-flyers, 2021.

[63] M. Ekal, K. Albee, B. Coltin, R. Ventura, R. Linares, and D. W. Miller. Model predictive control and integration with the autonomy stack of the astrobee free-flyer. Poster presented at i-SAIRAS 2020, https://www.hou.usra.edu/meetings/isairas2020/eposter/4073.pdf, 2020.

[64] K. Albee, M. Ekal, B. Coltin, R. Ventura, R. Linares, and D. W. Miller. The rattle motion planning algorithm for robust online parametric model improvement with on-orbit validation. *IEEE Robotics and Automation Letters*, 7(4):10946–10953, 2022.

[65] R. Featherstone. *Rigid body dynamics algorithms*. Springer, 2014.

[66] G. Strang, G. Strang, G. Strang, and G. Strang. *Introduction to linear algebra*, volume 3. Wellesley-Cambridge Press Wellesley, MA, 1993.

[67] B. Siciliano and O. Khatib. *Springer handbook of robotics*. Springer, 2016.

[68] E. G. Papadopoulos. *On the dynamics and control of space manipulators*. PhD thesis, Massachusetts Institute of Technology, 1990.

[69] M. Macdonald and V. Badescu. *The international handbook of space technology*. Springer, 2014.

[70] K. Yoshida. Space robot dynamics and control: to orbit, from orbit, and future. In *Robotics Research*, pages 449–456. Springer, 2000.

[71] J. L. Crassidis and J. L. Junkins. *Optimal estimation of dynamic systems*. Chapman and Hall/CRC, 2004.

[72] P. Tichavsky, C. H. Muravchik, and A. Nehorai. Posterior cramér-rao bounds for discrete-time nonlinear filtering. *IEEE Transactions on signal processing*, 46(5):1386–1396, 1998.

[73] Identification of the barycentric parameters of robot manipulators from external measurements. *Automatica*, 28(5):1011–1016, 1992.

[74] J. Swevers, C. Ganseman, D. B. Tukel, J. De Schutter, and H. Van Brussel. Optimal robot excitation and identification. *IEEE transactions on robotics and automation*, 13(5):730–740, 1997.

[75] J. Swevers, C. Ganseman, J. De Schutter, and H. Van Brussel. Experimental robot identification using optimised periodic trajectories. *Mechanical Systems and Signal Processing*, 10(5):561–577, 1996.

[76] W. Rackl, R. Lampariello, and G. Hirzinger. Robot excitation trajectories for dynamic parameter estimation using optimized b-splines. In *2012 IEEE international conference on robotics and automation*, pages 2042–2047. IEEE, 2012.

[77] B. Armstrong. On finding exciting trajectories for identification experiments involving systems with nonlinear dynamics. *The International journal of robotics research*, 8(6):28–48, 1989.

[78] J. T. Betts. Survey of numerical methods for trajectory optimization. *Journal of guidance, control, and dynamics*, 21(2):193–207, 1998.

[79] J. T. Betts. *Practical methods for optimal control and estimation using nonlinear programming*. Society for Industrial and Applied Mathematics, 2010.

[80] R. Tedrake. Underactuated robotics: Learning, planning, and control for efficient and agile machines course notes for mit 6.832. *Working draft edition*, 3, 2009.

[81] J. B. Rawlings, D. Q. Mayne, and M. Diehl. *Model predictive control: theory, computation, and design*, volume 2. Nob Hill Publishing Madison, WI, 2017.

[82] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM review*, 38(1):49–95, 1996.

[83] G. P. Tolstov. *Fourier series*. Courier Corporation, 2012.

[84] H. F. Davis. *Fourier series and orthogonal functions*. Courier Corporation, 1989.

[85] D. Sundararajan. The discrete fourier transform. In *Signals and Systems*, pages 125–160. Springer, 2023.

[86] L. Bluestein. A linear filtering approach to the computation of discrete fourier transform. *IEEE Transactions on Audio and Electroacoustics*, 18(4):451–455, 1970.

[87] W. T. Cochran, J. W. Cooley, D. L. Favin, H. D. Helms, R. A. Kaenel, W. W. Lang, G. C. Maling, D. E. Nelson, C. M. Rader, and P. D. Welch. What is the fast fourier transform? *Proceedings of the IEEE*, 55(10):1664–1674, 1967.

[88] J. E. Marsden and M. West. Discrete mechanics and variational integrators. *Acta Numerica*, 10 (1):357–514, 2001.

[89] T. M. Caldwell, D. Coleman, and N. Correll. Optimal parameter identification for discrete mechanical systems with application to flexible object manipulation. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 898–905. IEEE, 2014.

[90] J. Schultz, K. Flaßkamp, and T. D. Murphey. Variational integrators for structure-preserving filtering. *Journal of Computational and Nonlinear Dynamics*, 12(2), 2017.

[91] E. R. Johnson and T. D. Murphey. Scalable variational integrators for constrained mechanical systems in generalized coordinates. *IEEE Transactions on Robotics*, 25(6):1249–1261, 2009.

[92] H. Nijmeijer and A. J. Van der Schaft. *Nonlinear dynamical control systems*, volume 175. Springer, 1990.

[93] E. Eade. Lie groups for 2D and 3D transformations. *https://ethaneade.org/lie.pdf*, 2013.

[94] T. Lee, M. Leok, and N. H. McClamroch. Lie group variational integrators for the full body problem in orbital mechanics. *Celestial Mechanics and Dynamical Astronomy*, 98(2):121–144, 2007.

[95] N. Nordkvist and A. K. Sanyal. A lie group variational integrator for rigid body motion in se (3) with applications to underwater vehicle dynamics. In *49th IEEE Conference on Decision and Control (CDC)*, pages 5414–5419. IEEE, 2010.

[96] I. I. Hussein, M. Leok, A. K. Sanyal, and A. M. Bloch. A discrete variational integrator for optimal control problems on so (3). In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 6636–6641. IEEE, 2006.

[97] T. P. Setterfield, D. W. Miller, A. Saenz-Otero, E. Frazzoli, and J. J. Leonard. Inertial properties estimation of a passive on-orbit object using polhode analysis. *Journal of Guidance, Control, and Dynamics*, 41(10):2214–2231, 2018.

[98] B. E. Tweddle, A. Saenz-Otero, J. J. Leonard, and D. W. Miller. Factor graph modeling of rigid-body dynamics for localization, mapping, and parameter estimation of a spinning object in space. *Journal of Field Robotics*, 32(6):897–933, 2015.

[99] C. G. Atkeson, C. H. An, and J. M. Hollerbach. Estimation of inertial parameters of manipulator loads and links. *The International Journal of Robotics Research*, 5(3):101–119, 1986.

[100] J. Swevers, W. Verdonck, and J. De Schutter. Dynamic model identification for industrial robots. *IEEE Control Systems Magazine*, 27(5):58–71, 2007.

[101] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar. Design, modeling, estimation and control for aerial grasping and manipulation. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2668–2673. IEEE, 2011.

[102] M. L. Psiaki. Estimation of a spacecraft's attitude dynamics parameters by using flight data. *Journal of Guidance, Control, and Dynamics*, 28(4):594–603, 2005.

[103] Z. R. Manchester and M. A. Peck. Recursive inertia estimation with semidefinite programming. In *AIAA Guidance, Navigation, and Control Conference*, page 1902, 2017.

[104] Z. R. Manchester and M. A. Peck. Quaternion variational integrators for spacecraft dynamics. *Journal of Guidance, Control, and Dynamics*, 39(1):69–76, 2016.

[105] S. Leyendecker, S. Ober-Blöbaum, J. E. Marsden, and M. Ortiz. Discrete mechanics and optimal control for constrained systems. *Optimal Control Applications and Methods*, 31(6):505–528, 2010.

[106] K. Nanos and E. Papadopoulos. On parameter estimation of space manipulator systems with flexible joints using the energy balance. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3570–3576. IEEE, 2019.

[107] O.-O. Christidi-Loumpasefski, K. Nanos, and E. Papadopoulos. On parameter estimation of flexible space manipulator systems. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1843–1848. IEEE, 2020.

[108] Y. Liu, Y. Fu, W. He, and Q. Hui. Modeling and observer-based vibration control of a flexible spacecraft with external disturbances. *IEEE Transactions on Industrial Electronics*, 66(11):8648–8658, 2018.

[109] L. W. Taylor Jr. Parameter estimation for distributed parameter models of complex, flexible structures. *IFAC Proceedings Volumes*, 24(3):1155–1160, 1991.

[110] C. Edwards and S. Spurgeon. *Sliding mode control: theory and applications*. CRC Press, 1998.

[111] A. Ferrara and G. P. Incremona. Design of an integral suboptimal second-order sliding mode controller for the robust motion control of robot manipulators. *IEEE Transactions on Control Systems Technology*, 23(6):2316–2325, 2015.

[112] D. Q. Mayne, M. M. Seron, and S. Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2):219–224, 2005.

[113] B. T. Lopez, J. P. Howl, and J.-J. E. Slotine. Dynamic tube mpc for nonlinear systems. In *2019 American Control Conference (ACC)*, pages 1655–1662. IEEE, 2019.

[114] A. Majumdar and R. Tedrake. Funnel libraries for real-time robust feedback motion planning. *The International Journal of Robotics Research*, 36(8):947–982, 2017.

[115] P. A. Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. California Institute of Technology, 2000.

[116] S. Singh, A. Majumdar, J.-J. Slotine, and M. Pavone. Robust online motion planning via contraction theory and convex optimization. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5883–5890. IEEE, 2017.

[117] J.-J. E. Slotine and W. Li. *Applied Nonlinear Control*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1991. ISBN 0130408905.

[118] E. Lavretsky and K. A. Wise. Robust adaptive control. In *Robust and adaptive control*, pages 317–353. Springer, 2013.

[119] K. J. Åström and B. Wittenmark. *Adaptive control*. Courier Corporation, 2013.

[120] S. Abiko, R. Lampariello, and G. Hirzinger. Impedance control for a free-floating robot in the grasping of a tumbling target with parameter uncertainty. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1020–1025. IEEE, 2006.

[121] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[122] F. L. Lewis and D. Vrabie. Reinforcement learning and adaptive dynamic programming for feedback control. *IEEE circuits and systems magazine*, 9(3):32–50, 2009.

[123] D. Izzo, M. Märtens, and B. Pan. A survey on artificial intelligence trends in spacecraft guidance dynamics and control. *Astrodynamics*, 3(4):287–299, 2019.

[124] Y.-H. Wu, Z.-C. Yu, C.-Y. Li, M.-J. He, B. Hua, and Z.-M. Chen. Reinforcement learning in dual-arm trajectory planning for a free-floating space robot. *Aerospace Science and Technology*, 98: 105657, 2020.

[125] C. J. Ostafew, A. P. Schoellig, and T. D. Barfoot. Robust constrained learning-based nmpc enabling reliable mobile robot path tracking. *The International Journal of Robotics Research*, 35(13):1547–1563, 2016.

[126] O. Sigaud, C. Salaün, and V. Padois. On-line regression algorithms for learning mechanical models of robots: a survey. *Robotics and Autonomous Systems*, 59(12):1115–1129, 2011.

[127] P. Abbeel, M. Quigley, and A. Y. Ng. Using inaccurate models in reinforcement learning. In *Proceedings of the 23rd international conference on Machine learning*, pages 1–8, 2006.

[128] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[129] A. Feldbaum. Dual control theory I. *Automation Remote Control*, 21:874–880, 1960.

[130] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger. Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems*, 3:269–296, 2020.

[131] S. Patil, G. Kahn, M. Laskey, J. Schulman, K. Goldberg, and P. Abbeel. Scaling up gaussian belief space planning through covariance-free trajectory optimization and automatic differentiation. In *Algorithmic foundations of robotics XI*, pages 515–533. Springer, 2015.

[132] D. Nguyen-Tuong, J. Peters, and M. Seeger. Local gaussian process regression for real time online model learning and control. In *Proceedings of the 21st International Conference on Neural Information Processing Systems*, pages 1193–1200, 2008.

[133] N. Jansen, B. Könighofer, S. Junges, A. Serban, and R. Bloem. Safe Reinforcement Learning Using Probabilistic Shields (Invited Paper). In *31st International Conference on Concurrency Theory (CONCUR 2020)*, volume 171 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 3:1–3:16. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020.

[134] A. Albu-Schaffer, W. Bertleff, B. Rebele, B. Schafer, K. Landzettel, and G. Hirzinger. ROKVISS-robotics component verification on ISS current experimental results on parameter identification. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006.*, pages 3879–3885. IEEE, 2006.

[135] A. V. Oppenheim and R. W. Schafer. Digital signal processing(book). *Research supported by the Massachusetts Institute of Technology, Bell Telephone Laboratories, and Guggenheim Foundation. Englewood Cliffs, N. J., Prentice-Hall, Inc., 1975. 598 p*, 1975.

[136] P. Roque and R. Ventura. Space cobot: Modular design of an holonomic aerial robot for indoor microgravity environments. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4383–4390. IEEE, 2016.

[137] J. Xu and K. H. Halse. Dual quaternion variational integrator for rigid body dynamic simulation. *arXiv preprint arXiv:1611.00616*, 2016.

[138] P. M. Wensing, S. Kim, and J.-J. E. Slotine. Linear matrix inequalities for physically consistent inertial parameter identification: A statistical perspective on the mass distribution. *IEEE Robotics and Automation Letters*, 3(1):60–67, 2017.

[139] B. Kenwright. A beginners guide to dual-quaternions: what they are, how they work, and how to use them for 3D character hierarchies. *International Conference on Computer Graphics, Visualization and Computer Vision*, 20:1–13, 2012.

[140] M. Clifford. Preliminary sketch of biquaternions. *Proceedings of the London Mathematical Society*, 1(1):381–395, 1871.

[141] S. Traversaro, S. Brossette, A. Escande, and F. Nori. Identification of fully physical consistent inertial parameters using optimization on manifolds. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5446–5451. IEEE, 2016.

[142] B. Houska, H. Ferreau, and M. Diehl. ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization. *Optimal Control Applications and Methods*, 32(3):298–312, 2011.

[143] M. C. Grant and S. P. Boyd. Graph implementations for nonsmooth convex programs. In *Recent advances in learning and control*, pages 95–110. Springer, 2008.

[144] M. C. Grant and S. P. Boyd. Cvx: Matlab software for disciplined convex programming, version 2.0 beta. `http://cvxr.com/cvx`, 2003.

[145] B. Doerr, K. Albee, M. Ekal, R. Ventura, and R. Linares. The reswarm microgravity flight experiments: Planning, control, and model estimation for on-orbit close proximity operations. *arXiv preprint arXiv:2301.01319*, 2023.

[146] K. Albee. *The ReSWARM Microgravity Flight Experiments: Planning, Control, and Model Estimation for On-Orbit Close Proximity Operations*. PhD thesis, Massachusetts Institute of Technology, 2022.

[147] O. Junge, J. E. Marsden, and S. Ober-Blöbaum. Discrete mechanics and optimal control. *IFAC Proceedings Volumes*, 38(1):538–543, 2005.

[148] D. Y. Lee, R. Gupta, U. V. Kalabić, S. Di Cairano, A. M. Bloch, J. W. Cutler, and I. V. Kolmanovsky. Geometric mechanics based nonlinear model predictive spacecraft attitude control with reaction wheels. *Journal of Guidance, Control, and Dynamics*, 40(2):309–319, 2017.

[149] K. Xu, J. Timmermann, and A. Trächtler. Nonlinear model predictive control with discrete mechanics and optimal control. In *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 1755–1761. IEEE, 2017.

[150] S. M. LaValle and J. J. Kuffner Jr. Randomized kinodynamic planning. *The international journal of robotics research*, 20(5):378–400, 2001.

[151] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*, 30(7):846–894, 2011.

[152] Y.-K. Choi, J.-W. Chang, W. Wang, M.-S. Kim, and G. Elber. Continuous collision detection for ellipsoids. *IEEE Transactions on visualization and Computer Graphics*, 15(2):311–325, 2008.

[153] J. H. Taylor. The cramer-rao estimation error lower bound computation for deterministic nonlinear systems. In *1978 IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes*, pages 1178–1181, Jan 1978. doi: 10.1109/CDC.1978.268121.

[154] G. Marafioti, S. Tebbani, D. Beauvois, G. Becerra, A. Isambert, and M. Hovd. Unscented kalman filter state and parameter estimation in a photobioreactor for microalgae production. *IFAC Proceedings Volumes*, 42(11):804–809, 2009.

[155] M. G. Bualat, T. Smith, E. E. Smith, T. Fong, and D. Wheeler. Astrobee: A new tool for iss operations. In *2018 SpaceOps Conference*, page 2517, 2018.

[156] L. Fluckiger, K. Browne, B. Coltin, J. Fusco, T. Morse, and A. Symington. Astrobee robot software: A modern software system for space. In *iSAIRAS (International Symposium on Artificial Intelligence, Robotics and Automation in Space)*, number ARC-E-DAA-TN55483, 2018.

[157] A. M. Vargas, R. G. Ruiz, P. Wofford, V. Kumar, B. Van Ross, A. Katterhagen, J. Barlow, L. Flückiger, J. Benavides, T. Smith, et al. Astrobee: Current status and future use as an international research platform. In *International Astronautical Congress (IAC)*, 2018.

[158] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, et al. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[159] M. Watterson, T. Smith, and V. Kumar. Smooth trajectory generation on se(3) for a free flying space robot. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5459–5466, 2016. doi: 10.1109/IROS.2016.7759803.

[160] B. Coltin, J. Fusco, Z. Moratto, O. Alexandrov, and R. Nakamura. Localization from visual landmarks on a free-flying robot. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4377–4382. IEEE, 2016.

[161] R. Soussan, V. Kumar, B. Coltin, and T. Smith. Astroloc: An efficient and robust localizer for a free-flying robot. In *IEEE International Conference on Robotics and Automation*, pages 4106–4112, Philadelphia, PA, USA, May 2022. IEEE. ISBN 978-1-72819-681-7. doi: 10.1109/ICRA46639.2022.9811919.

[162] K. Albee, M. Ekal, and C. Oestreich. A Brief Guide to Astrobee's Flight Software. https://github.com/albee/a-brief-guide-to-astrobee, 2020.

[163] P. Kim, B. Coltin, O. Alexandrov, and H. J. Kim. Robust visual localization in changing lighting conditions. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 5447–5452. IEEE, 2017.

[164] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl. Casadi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019.

[165] E. Coumans. Bullet physics simulation. In *ACM SIGGRAPH 2015 Courses*, New York, NY, USA, 2015. ACM. ISBN 9781450336345. doi: 10.1145/2776880.2792704. URL `https://doi.org/10.1145/2776880.2792704`.

[166] D. Maclaurin, D. Duvenaud, and R. P. Adams. Autograd: effortless gradients in numpy. In *ICML 2015 AutoML Workshop*, volume 238, page 5, 2015.

[167] E. Daley. Astrobee free-flyer nozzle mechanism summary. In *The 45th Aerospace Mechanism Symposium*, number ARC-E-DAA-TN73478, 2020.

[168] J. L. Junkins and H. Schaub. *Analytical mechanics of space systems*. American Institute of Aeronautics and Astronautics, 2009.

[169] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[170] F. M. Mirzaei and S. I. Roumeliotis. A kalman filter-based algorithm for imu-camera calibration: Observability analysis and performance evaluation. *IEEE transactions on robotics*, 24(5):1143–1156, 2008.

[171] B. Doerr, K. Albee, M. Ekal, R. Linares, and R. Ventura. Safe and uncertainty-aware robotic motion planning techniques for agile on-orbit assembly. In *31st AAS/AIAA Space Flight Mechanics Meeting*, 2021.

[172] A. D. Wilson and T. D. Murphey. Maximizing fisher information using discrete mechanics and projection-based trajectory optimization. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2403–2409. IEEE, 2015.

[173] S. Santos, M. Ekal, and R. Ventura. Symplectic momentum neural networks-using discrete variational mechanics as a prior in deep learning. In *Learning for Dynamics and Control Conference*, pages 584–595. PMLR, 2022.

[174] M. Hall. *The theory of groups*. Courier Dover Publications, 2018.

[175] W. R. Hamilton. *Elements of quaternions*. Longmans, Green, & Company, 1866.

[176] M. D. Shuster et al. A survey of attitude representations. *Navigation*, 8(9):439–517, 1993.

[177] N. Trawny and S. I. Roumeliotis. Indirect kalman filter for 3D attitude estimation. *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep*, 2:2005, 2005.

[178] F. L. Markley and J. L. Crassidis. *Fundamentals of spacecraft attitude determination and control*. Springer, 2014.

[179] C. Gehring, D. Bellicoso, M. Bloesch, H. Sommer, P. Fankhauser, M. Hutter, and R. Siegwart. Kindr library – kinematics and dynamics for robotics. `https://opensource.docs.anymal.com/doxygen/kindr/master/cheatsheet_latest.pdf`, 2021.

[180] S. Verhagen and P. J. Teunissen. Least-squares estimation and kalman filtering. In *Springer Handbook of Global Navigation Satellite Systems*, pages 639–660. Springer, 2017.

[181] R. E. Kalman. A new approach to linear filtering and prediction problems, 1960.

[182] S. Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002.

[183] S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[184] E. A. Wan and R. Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*, pages 153–158. Ieee, 2000.

[185] E. M. Kong, A. Saenz-Otero, S. Nolet, D. S. Berkovitz, D. W. Miller, and S. W. Sell. Spheres as formation flight algorithm development and validation testbed: Current progress and beyond. In *Proceedings from the 2nd International Symposium on Formation Flying Missions and Technologies*, 2004.

[186] A. S. Otero, A. Chen, D. W. Miller, and M. Hilstad. Spheres: Development of an ISS laboratory for formation flight and docking research. In *Proceedings, IEEE Aerospace Conference*, volume 1, pages 1–1. IEEE, 2002.

[187] G. Lapilli, D. R. Kirk, H. Gutierrez, P. Schallhorn, B. Marsell, J. Roth, and J. Moder. Results of microgravity fluid dynamics captured with the spheres-slosh experiment. In *International Astronautical Congress*, number IAC-15-A2. 6.2, 2015.

[188] A. Saenz-Otero, J. Katz, and D. Miller. Spheres demonstrations of satellite formations aboard the ISS. In *32nd Annual AAS Guidance and Control Conference*, pages 2009–011. Citeseer, 2009.

[189] D. Fourie, B. Tweddle, S. Ulrich, and A. Saenz Otero. Vision-based relative navigation and control for autonomous spacecraft inspection of an unknown object. In *AIAA guidance, navigation, and control (GNC) conference*, page 4759, 2013.

[190] K. Van Hecke, G. C. de Croon, D. Hennes, T. P. Setterfield, A. Saenz-Otero, and D. Izzo. Self-supervised learning as an enabling technology for future space exploration robots: ISS experiments on monocular distance learning. *Acta Astronautica*, 140:1–9, 2017.

[191] H. J. Kramer. ISS utilization: SPHERES (Synchronized Position Hold Engage Re-orient Experiment Satellites). `https://directory.eoportal.org/web/eoportal/satellite-missions/i/iss-spheres`, 2012.

[192] T. Smith, J. Barlow, M. Bualat, T. Fong, C. Provencher, H. Sanchez, and E. Smith. Astrobee: A new platform for free-flying robotics on the international space station. 2016.

[193] T. G. Chen, A. Cauligi, S. A. Suresh, M. Pavone, and M. R. Cutkosky. Testing gecko-inspired adhesives with astrobee aboard the international space station: Readying the technology for space. *IEEE Robotics & Automation Magazine*, 29(3):24–33, 2022.

[194] C. Oestreich, A. T. Espinoza, J. Todd, K. Albee, and R. Linares. On-orbit inspection of an unknown, tumbling target using nasa's astrobee robotic free-flyers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2039–2047, 2021.

Appendix A

Preliminaries

This chapter covers the preliminary concepts relevant to the thesis such as rigid body pose representations and central estimation concepts like least squares and the Kalman filter.

A.1 Continuous rigid body kinematics and dynamics

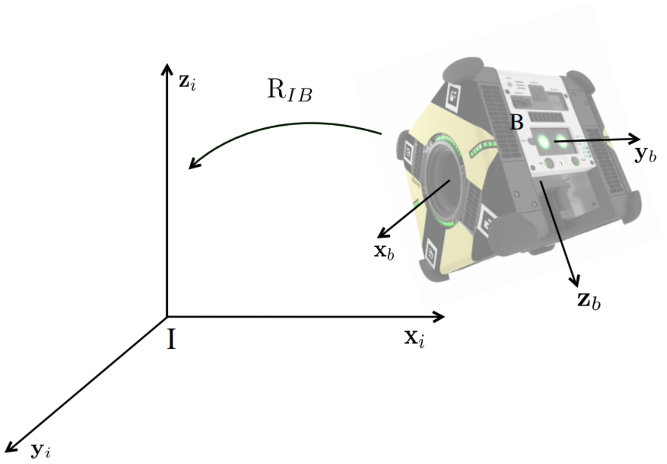


Figure A.1: An illustration of reference frames [Astrobee image credit: NASA]

Reference frames and object pose

A coordinate reference frame [67] is composed of an origin and a triad of right-handed mutually orthogonal basis vectors (for instance, frame I with origin I and vectors $\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i$ from fig.A.1), all of which are fixed with respect to a particular body. Rigid body pose (position and orientation) can be expressed as the rotation and translation between frames. A *body* frame is considered attached to the moving rigid-body while the stationary frame of reference is generally an *inertial* frame. For example, in fig.A.1, the pose and motion of the Astrobee's body frame B will be expressed with respect to the inertial frame I.

A minimum of six parameters are needed ot locate a body in Euclidean space, three for position and orientation each [67]. Position is expressed as a vector in $\mathbb{R}^3$, given by the position of the origin of the robot frame (e.g., $\mathbf{B}$) relative to that of another frame (e.g., $\mathbf{I}$). Several representations of orientation exist, each with their own special features and drawbacks.

Rotation matrix

The direction cosine matrix (DCM) or rotation matrix can be used to represent the orientation of one frame relative to another. In case of the previous example, the orientation of the robot (orientation of B relative to I) is given by [67, 168]

$$\mathbf{R}_{IB} = \begin{bmatrix} \mathbf{x}_i \cdot \mathbf{x}_b & \mathbf{y}_i \cdot \mathbf{x}_b & \mathbf{z}_i \cdot \mathbf{x}_b \\ \mathbf{x}_i \cdot \mathbf{y}_b & \mathbf{y}_i \cdot \mathbf{y}_b & \mathbf{z}_i \cdot \mathbf{y}_b \\ \mathbf{x}_i \cdot \mathbf{z}_b & \mathbf{y}_i \cdot \mathbf{z}_b & \mathbf{z}_i \cdot \mathbf{z}_b \end{bmatrix} \tag{A.1}$$

where the $\cdot$ (dot) operator denotes inner product, $\mathbf{x}_b, \mathbf{y}_b, \mathbf{z}_b$ are the mutually orthogonal basis vectors of the body frame B. As the basis vectors are unit vectors, each element of $\mathbf{R}$ reduces to cosine of the angle between the corresponding two vectors, thus the name DCM. Rotation matrices are orthogonal, i.e, its columns are of unit length and mutually perpendicular (orthonormal), with determinant 1. Given that they are a nine-element representation used for expressing the three parameters of rotation, the conditions of orthonormality provide six constraints between its elements. Orthogonality also implies that the inverse of a rotation matrix is equal to its transpose

$$\mathbf{R}_{BI} = \mathbf{R}_{IB}^{-1} = \mathbf{R}_{IB}^{T} \tag{A.2}$$

where the superscripts $^{-1}$ and ^{T} denote the matrix inverse and transpose, respectively. Further, $\mathbf{R}_{IB}^{T}\mathbf{R}_{IB} = \mathbb{I}_{3\times3}$, where $\mathbb{I}_{3\times3}$ is the 3×3 identity matrix. Rotations can be composed by matrix multiplication of DCMs. For instance, let $\mathbf{R}_{BC}$ denote the orientation of Astrobee's navigation camera frame, $\mathbf{C}$, with respect to its body frame. Then,

$$\mathbf{R}_{IC} = \mathbf{R}_{IB}\mathbf{R}_{BC} \tag{A.3}$$

In addition to expressing orientation, DCMs are used to transform a vector with components expressed in one frame into another. For instance, if $\mathbf{r}_b$ is an arbitrary vector expressed in $\mathbf{B}$

$$\mathbf{r}_i = \mathbf{R}_{IB}\mathbf{r}_b \tag{A.4}$$

where $\mathbf{r}_i$ signifies the components of $\mathbf{r}$ expressed in frame I.

Rotation matrices are singularity free representations of attitude and uniquely represent an orientation. The set of rotations, i.e, all orthogonal matrices with determinant 1, under the operation of matrix multiplication forms a Lie group[1] called as $SO(3)$ or *special orthogonal group*, or the rotation group in general [67]. Quaternions (examined below) and Euler angles are examples of alternate parametrization of rotations from $SO(3)$.

Quaternions

Quaternions are a redundant and singularity free attitude representation suitable for describing large rotations [168]. First described by Hamilton [175], quaternions are defined as

$$\bar{\boldsymbol{q}} = q_w + q_x\boldsymbol{i} + q_y\boldsymbol{j} + q_z\boldsymbol{k} \tag{A.5}$$

where $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ are hypercomplex numbers, such that $\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = -1$, while q_w, q_x, q_y and q_z are real numbers. It is to be noted here that two conventions of representing quaternions exist, Hamilton and JPL (discussed in [176] and [177], respectively). This thesis uses the former, where $\boldsymbol{ij} = \boldsymbol{k},\ -\boldsymbol{jk} = \boldsymbol{i},\ -\boldsymbol{ki} = \boldsymbol{j}$. Using vector form, quaternions can be represented as

$$\bar{\boldsymbol{q}} = \begin{bmatrix} \boldsymbol{q} \\ q_w \end{bmatrix}, \quad \boldsymbol{q} = [q_x, q_y, q_z]^T \tag{A.6}$$

Note that the scalar-last ordering is used here, which is a choice independent of the convention used. Rotation quaternions or unit quaternions - which will be simply referred to as quaternions from hereon - are of unit length

$$||\bar{\boldsymbol{q}}|| = \sqrt{\bar{\boldsymbol{q}}^T\bar{\boldsymbol{q}}} = 1 \tag{A.7}$$

and fulfill the relation

$$\bar{\boldsymbol{q}} = \begin{bmatrix} \mathbf{e}\sin(\vartheta/2) \\ \cos(\vartheta/2) \end{bmatrix} \tag{A.8}$$

where $\mathbf{e}$ is a unit vector that specifies the rotation axis fixed in both the initial and final orientation, while ϑ is the angle of rotation. Both $\mathbf{e}$ and ϑ form the Euler Axis-angle representation [178]. A few important properties of quaternions are

- Quaternions $\bar{\boldsymbol{q}}$ and $-\bar{\boldsymbol{q}}$ produce the same result, albeit with different directions of rotation to get to the final configuration
- Quaternion multiplication, or the composition of two rotations described by $\bar{\boldsymbol{q}}_1$ and $\bar{\boldsymbol{q}}_2$, denoted here by the symbol $\circ$, is found as,

  $$\bar{\boldsymbol{q}}_1 \circ \bar{\boldsymbol{q}}_2 = \mathbf{Q}(\bar{\boldsymbol{q}}_1)\bar{\boldsymbol{q}}_2 \tag{A.9}$$

  where

  $$\mathbf{Q}(\bar{\boldsymbol{q}}) = \begin{bmatrix} \mathbf{I}_{3\times3}q_w + S(\boldsymbol{q}) & \boldsymbol{q} \\ -\boldsymbol{q}^T & q_w \end{bmatrix} \tag{A.10}$$

---

[1] In mathematics, a group is a non-empty set of elements and a binary operation, whose members satisfy the four conditions of closure, identity, inverse and associativity [174]

where $S()$ denotes the skew-symmeric matrix and $\mathbf{I}_{3\times 3}$ denotes the 3×3 identity matrix.

- The quaternion conjugate is defined as

$$\bar{\boldsymbol{q}}^{\dagger} = [-\boldsymbol{q}, q_w]^T, \quad \bar{\boldsymbol{q}} \circ \bar{\boldsymbol{q}}^{\dagger} = \bar{\boldsymbol{I}} = [0,0,0,1]^T \tag{A.11}$$

- After a vector $\mathbf{r}_B \in \mathbb{R}^3$ is rotated by a quaternion $\bar{\boldsymbol{q}}_{IB}$, the transformed vector is given by

$$\mathbf{r}_I = \bar{\boldsymbol{q}}_{IB} \circ \hat{\mathbf{r}}_B \circ \bar{\boldsymbol{q}}_{IB}^{\dagger} \tag{A.12}$$

where the hat symbol denotes a pure quaternion, $\hat{\mathbf{r}} = [\mathbf{r}^T 0]^T$. The conversion between a quaternion $\boldsymbol{q}_{IB}$ to the corresponding rotation matrix R_{IB} follows from eq.(A.12), and is given as[2]:

$$R_{IB} = \begin{bmatrix} q_w^2 + q_x^2 - q_y^2 - q_z^2 & 2q_xq_y - 2q_zq_w & 2q_wq_y + q_xq_z \\ 2q_xq_y + 2q_zq_w & q_w^2 - q_x^2 + q_y^2 - q_z^2 & 2q_yq_z - 2q_wq_x \\ 2q_xq_z - 2q_wq_y & 2q_yq_z + 2q_wq_x & q_w^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix} \tag{A.13}$$

or compactly written as $R_{IB} = (2q_w^2 - 1)\mathbf{I}_{3\times 3} + 2q_w\boldsymbol{q} + 2\boldsymbol{q}\boldsymbol{q}^T$

A note on the special Euclidean group

The complete rigid body pose is thus given by position in group $\mathbb{R}^3$ and attitude from $\mathrm{SO}(3)$. The product space of $\mathbb{R}^3$ and $\mathrm{SO}(3)$ forms a group known as the special Euclidean group, $\mathrm{SE}(3)$ [67]:

$$\mathrm{SE}(3) = \{(\mathbf{p}, \mathbf{R}) : \mathbf{p} \in \mathbb{R}^3, \mathbf{R} \in \mathrm{SO}(3)\} \tag{A.14}$$

Members of this group are a set of 4×4 homogeneous transformation matrices,

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{A.15}$$

Similar to rotation matrices, the transformations can be composed by multiplication of individual transformations. For instance, $\mathbf{T}_{IC} = \mathbf{T}_{IB}\mathbf{T}_{BC}$. Further the inverse of a homogenous transformation matrix performs the reverse transformation opertation, i.e, $\mathbf{T}_{IB}^{-1} = \mathbf{T}_{BI}$. Dual quaternions, discussed in section 5.1.1, can also be used to represent poses in $\mathrm{SE}(3)$.

---

[2] A handy summary of key rotation representations and their properties can be found in this cheatsheet [179]

A.2 Estimation algorithms

A.2.1 The Least Squares Approach for estimation

Linear least squares estimation [71, 180] can be used to solve problems of the form

$$\tilde{\mathbf{y}} = \mathbf{H}\boldsymbol{\theta} + \mathbf{v} \tag{A.16}$$

where,

$\tilde{\mathbf{y}} \in \mathbb{R}^m$: measurement vector

$\boldsymbol{\theta} \in \mathbb{R}^n$: vector of parameters to be estimated

$\mathbf{v} \in \mathbb{R}^m$: measurement noise

$\mathbf{H} \in \mathbb{R}^{m \times n}$: regressor that maps the parameters to be estimated to the observations

Consider the residual error is given by $\mathbf{e} = \tilde{\mathbf{y}} - \hat{\mathbf{y}}$, where $\hat{\mathbf{y}} = \mathbf{H}\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\theta}}$ denotes the estimated value of $\boldsymbol{\theta}$. Then, according to Gauss' principle of least squares, the optimum choice for the parameter $\hat{\boldsymbol{\theta}}$ is one that minimizes the cost function

$$J = \frac{1}{2}\mathbf{e}^T W \mathbf{e} \tag{A.17}$$

or

$$J = \frac{1}{2}\left(\tilde{\mathbf{y}} - \mathbf{H}\boldsymbol{\theta}\right)^T W \left(\tilde{\mathbf{y}} - \mathbf{H}\boldsymbol{\theta}\right) \tag{A.18}$$

where $W \in \mathbb{R}^{m \times m}$ is a symmetric error weighting matrix. The necessary condition for the global minimum of the quadratic cost (A.18) is

$$\nabla_{\hat{\boldsymbol{\theta}}} J \equiv \begin{bmatrix} \frac{\partial J}{\partial \hat{\boldsymbol{\theta}}_1} \\ \vdots \\ \frac{\partial J}{\partial \hat{\boldsymbol{\theta}}_n} \end{bmatrix} = \mathbf{H}^T W \mathbf{H}\hat{\boldsymbol{\theta}} - \mathbf{H}^T W \tilde{\mathbf{y}} = \mathbf{0} \tag{A.19}$$

with the sufficient condition being [168]

$$\nabla^2_{\hat{\boldsymbol{\theta}}} J \equiv \frac{\partial^2 J}{\partial \hat{\boldsymbol{\theta}} \partial \hat{\boldsymbol{\theta}}^T} = \left(\mathbf{H}^T W \mathbf{H}\right) \succ 0 \tag{A.20}$$

An analytical solution to the linear least squares problem results from (A.19), which, provided that $\mathbf{H}$ is full-rank and well-conditioned, is given as:

$$\hat{\boldsymbol{\theta}} = \left(\mathbf{H}^T W \mathbf{H}\right)^{-1} \mathbf{H}^T W \tilde{\mathbf{y}} \tag{A.21}$$

In case the noise has a zero mean Gaussian distribution with a known covariance R, $\tilde{\mathbf{y}} \sim N(0, R)$, the weighting $W = R^{-1}$ can be used, and the solution corresponds to the minimum variance estimate [168].

The least squares approximation is a powerful tool in estimation, with variations such as weighted least squares, recursive least squares, and total least squares in case of errors in both the measurements and the regressor [71]. This method has also been extended to non-linear cases of the form $\mathbf{y} = h(\boldsymbol{\theta})$, where the system is linearized about a current estimate, and algorithms such as Levenberg-Marquardt can be employed. In case of nonlinear problem formulations with constraints on $\boldsymbol{\theta}$, constrained optimization is widely used to minimize the objective J.

Recursive least squares

In many real-world scenarios, data must be processed sequentially rather than all at once, in a batch fashion. To simplify the derivation of the recursive least squares approach, two subsets of measurements are considered [168]:

$$\tilde{\mathbf{y}}_1 = \mathbf{H}_1\boldsymbol{\theta} + \mathbf{v}_1 \tag{A.22}$$

$$\tilde{\mathbf{y}}_2 = \mathbf{H}_2\boldsymbol{\theta} + \mathbf{v}_2 \tag{A.23}$$

where,

$\tilde{\mathbf{y}}_1 \in \mathbb{R}^{m_1}, \tilde{\mathbf{y}}_2 \in \mathbb{R}^{m_2}$: measurement vectors

$\mathbf{v}_1 \in \mathbb{R}^{m_1}, \mathbf{v}_2 \in \mathbb{R}^{m_2}$: measurement noise vectors

$\mathbf{H}_1 \in \mathbb{R}^{m_1 \times n}, \mathbf{H}_2 \in \mathbb{R}^{m_2 \times n}$: known regressors

An initial estimate of the parameters using the least squares solution for the first subset gives $\hat{\boldsymbol{\theta}}_1 = \left(\mathbf{H}_1^T W_1 \mathbf{H}_1\right)^{-1} \mathbf{H}_1^T W \tilde{\mathbf{y}}_1$. The final solution should resolve the following problem, taking into account both sets of measurements:

$$\tilde{\mathbf{y}} = \mathbf{H}\boldsymbol{\theta} + \mathbf{v} \tag{A.24}$$

where,

$$\tilde{\mathbf{y}} = \begin{bmatrix} \tilde{\mathbf{y}}_1 \\ \tilde{\mathbf{y}}_2 \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}, \quad W = \begin{bmatrix} W_1 & 0 \\ 0 & W_2 \end{bmatrix}$$

In defining the matrix W as block diagonal, the assumption that only measurements of the same subset can be correlated is made. The optimal solution of (A.24), based on the first two measurements, will therefore be

$$\hat{\boldsymbol{\theta}}_2 = \left(\mathbf{H}^T W \mathbf{H}\right)^{-1} \mathbf{H}^T W \tilde{\mathbf{y}} \tag{A.25}$$

Since $\mathbf{W}$ is block diagonal,

$$\hat{\boldsymbol{\theta}}_2 = \left[\mathbf{H}_1^T W_1 \mathbf{H}_1 + \mathbf{H}_2^T W_2 \mathbf{H}_2\right]^{-1} \left(\mathbf{H}_1^T W_1 \tilde{\mathbf{y}}_1 + \mathbf{H}_2^T W_2 \tilde{\mathbf{y}}_2\right) \tag{A.26}$$

Defining the following

$$P_1 \equiv \left[\mathbf{H}_1^T W_1 \mathbf{H}_1\right]^{-1} \tag{A.27}$$

$$P_2 \equiv \left[\mathbf{H}_1^T W_1 \mathbf{H}_1 + \mathbf{H}_2^T W_2 \mathbf{H}_2\right]^{-1} \tag{A.28}$$

Equation A.26 can be then rewritten as

$$\hat{\boldsymbol{\theta}}_2 = P_2 \left(\mathbf{H}_1^T W_1 \tilde{\mathbf{y}}_1 + \mathbf{H}_2^T W_2 \tilde{\mathbf{y}}_2\right) \tag{A.29}$$

and P_2 as

$$P_2^{-1} = P_1^{-1} + \mathbf{H}_2^T W_2 \mathbf{H}_2 \tag{A.30}$$

Algebraic manipulation results in a sequential approach for the estimates

$$\hat{\boldsymbol{\theta}}_{k+1} = \hat{\boldsymbol{\theta}}_k + K_{k+1} \left[\tilde{\mathbf{y}} - \mathbf{H}_{k+1} \hat{\boldsymbol{\theta}}_k\right] \tag{A.31}$$

where

$$K_{k+1} = P_{k+1} \mathbf{H}_{k+1}^T W_{k+1} \tag{A.32}$$

$$P_{k+1}^{-1} = P_k^{-1} + \mathbf{H}_{k+1}^T W_{k+1} \mathbf{H}_{k+1} \tag{A.33}$$

where K is the correction term, or the Kalman gain, and equation A.31 is the Kalman update equation. Instead of performing matrix inversions at every update of P in eq. A.33, use of the Sherman-Woodbury-Morrison matrix inversion lemma for rank adjustment, along with some algebraic manipulation results in the covariance recursion form,

$$\hat{\boldsymbol{\theta}}_{k+1} = \hat{\boldsymbol{\theta}}_k + K_{k+1} \left[\tilde{\mathbf{y}} - \mathbf{H}_{k+1} \hat{\boldsymbol{\theta}}_k\right] \tag{A.34}$$

$$K_{k+1} = P_{k+1} \mathbf{H}_{k+1}^T \left[\mathbf{H}_{k+1} P_k \mathbf{H}_{k+1}^T + W_{k+1}^T\right] \tag{A.35}$$

$$P_{k+1} = \left[I - K_{k+1} \mathbf{H}_{k+1}\right] P_k \tag{A.36}$$

Note that these are exactly the update and Kalman gain equations used in the Kalman filter (table A.1), with W_k being the inverse of the measurement error covariance, R_k. If W is indeed block diagonal, i.e., the measurements from different subsets are not correlated, the recursive least squares solution will be equivalent to that of the batch least squares, barring arithmetic errors.

A.2.2 Sequential estimators

Sequential state estimators are usually employed when the measurements are related to the estimated parameters by models that consist of dynamic and algebraic equations. In other words, they account for the propagation of the dynamic system state between measurements. These techniques also apply to the estimation of parameters that change over time and are widely employed in state estimation.

The Kalman filter (KF) [181] is a classical sequential estimation method. Being an optimal filter with solid stability properties, the KF is a propagation-update framework that yields the best estimate in the mean-squared error sense for systems with linear process and measurement models, Gaussian distributed initial state and Gaussian noise distribution. As a result of these assumptions, the filter only propagates the mean and the covariance of the states rather than propagating the entire probability distribution function. The KF is also used as an implementation of the Bayes filter for belief computation, representing the belief through its mean and covariance [182]. For this, the Markov assumptions of the Bayes filter have to hold, in addition to the linearity and Gaussian of the KF assumptions mentioned above.

Consider a linear system with the discrete process and measurement models of the following form

$$\mathbf{x}_{k+1} = \Phi_k \mathbf{x}_k + B_k \mathbf{u}_k + \mathbf{w}_k, \;\; \mathbf{w}_k \sim N(0, Q) \tag{A.37}$$

$$\tilde{\mathbf{y}}_k = H_k \mathbf{x}_k + \mathbf{v}_k, \;\; \mathbf{v}_k \sim N(0, R) \tag{A.38}$$

where $\mathbf{x}$ represents the states of the system and $\tilde{y}$, the measurements. Φ, and B and H are matrices, with Φ denoting the state transition matrix. The Gaussian process and measurement noise are denoted by $\mathbf{w}$ and $\mathbf{v}$, respectively. Table A.1 summarizes the discrete Kalman Filter equations for estimation of state $\mathbf{x}$

Table A.1: Discrete Kalman Filter equations

Kalman Gain	$K_k = P_k^- H_k^T \left[H_k P_k^- H_k^T + R\right]^{-1}$
Update	$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k \left[\tilde{\mathbf{y}}_k - H_k \hat{\mathbf{x}}_k^-\right]$ $P_k^+ = \left[I - K_k H_k\right] P_k^-$
Propagation	$\hat{\mathbf{x}}_{k+1}^- = \Phi_k \hat{\mathbf{x}}_k^+ + B_k \mathbf{u}_k$ $P_{k+1}^- = \Phi_k P_k^+ \Phi_k^T + Q$

Dynamic systems are usually nonlinear, with process and measurement models of the form

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k, \;\; \mathbf{w}_k \sim N(0, Q) \tag{A.39}$$

$$\tilde{\mathbf{y}}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k, \;\; \mathbf{v}_k \sim N(0, R) \tag{A.40}$$

For sequential estimation of nonlinear systems, the Kalman Filter finds extensions in the Extended Kalman Filter (EKF) [71] (Table A.2), where the system dynamics are linearized about the previous estimate, and the linear transformations H_k and Φ_k in the KF equations are substituted by the corresponding Jacobians [183]. Some limitations of the EKF can arise from poor linear approximations and difficulty in computing dense Jacobians. The Unscented Kalman Filter (UKF) [183, 184] addresses these drawbacks by using a fixed number of parameters (sigma points) to approximate the Gaussian distribution of

Table A.2: Discrete Extended Kalman Filter equations

Kalman Gain	$H_k(\hat{\mathbf{x}}_k^-) \equiv \left.\frac{\partial \mathbf{h}}{\partial \hat{\mathbf{x}}}\right\rvert_{\hat{\mathbf{x}}_k^-}$ $K_k = P_k^- H_k^T \left[H_k P_k^- H_k^T + R\right]^{-1}$
Update	$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k \left[\tilde{\mathbf{y}}_k - h(\hat{\mathbf{x}}_k^-)\right]$ $P_k^+ = \left[I - K_k H_k\right] P_k^-$
Propagation	$\hat{\mathbf{x}}_{k+1}^- = f(\hat{\mathbf{x}}_k^+, \mathbf{u}_k)$ $\Phi_k(\hat{\mathbf{x}}_k^+) \equiv \left.\frac{\partial \mathbf{f}}{\partial \hat{\mathbf{x}}}\right\rvert_{\hat{\mathbf{x}}_k^+}$ $P_{k+1}^- = \Phi_k P_k^+ \Phi_k^T + Q$

the state, propagating them non-linearly, and retrieving the mean and covariance of this transformation via the unscented transform (UT). Fig.A.2 illustrates the difference between the mean and covariance propagation approaches of the EKF and the UKF.

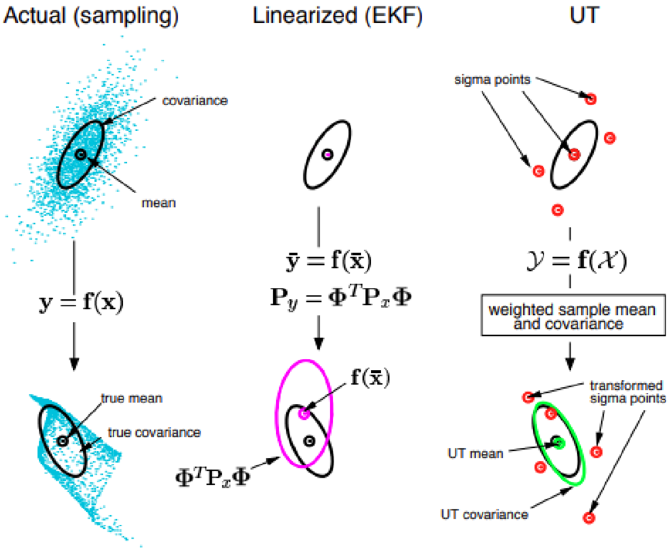


Figure A.2: An illustration of mean and covariance propagation adapted from [184]. L-R: sampling, first-order linearization (EKF), and unscented transform (UT)

Appendix B

Testing platforms

Robotic experimental platforms in space provide an extraordinary opportunity to test hardware payload and software, such as autonomy algorithms, in microgravity. They are a flexible, inexpensive, risk-tolerant means of validating under-development algorithms and simulation-based research in space [185]. This section discusses SPHERES and Astrobee, two testbeds aboard the International Space Station (ISS) that played a role in developing the algorithms presented in this thesis.

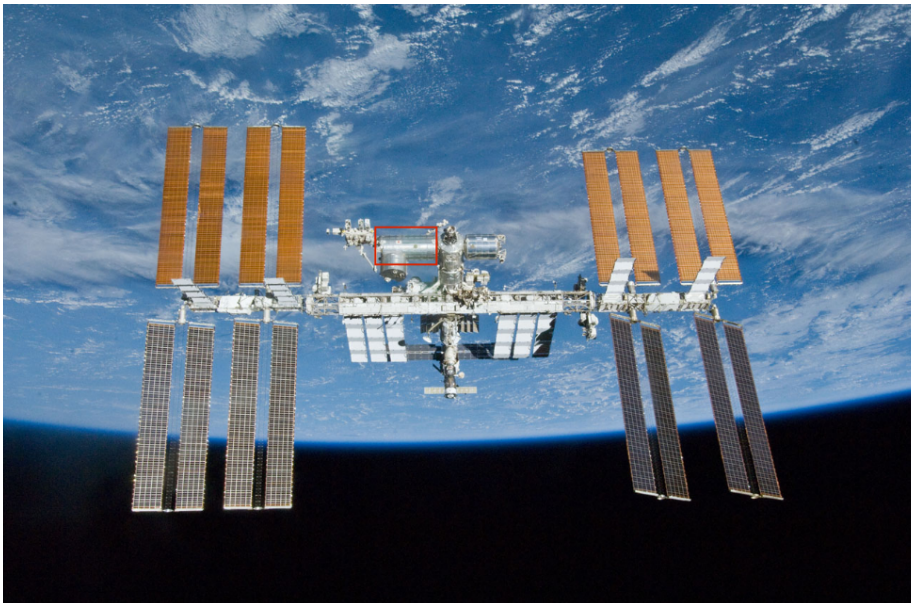


Figure B.1: The International Space Station with the Japanese Experiment Module (JEM) highlighted [Credit: NASA/ISS]

SPHERES

The Synchronized Position Hold Engage and Reorient Experimental Satellite (SPHERES)[1] are free flying experimental testbeds, set up with the goal of validating formation flight and docking technologies in space [185]. Three SPHERES were on board the Japanese Experiment or Kibo module of the ISS

---

[1] `https://www.nasa.gov/spheres/home`

(fig. B.1), operational from 2006 to 2019. Serving as software and hardware testbeds, the SPHERES are polyhedral satellites of about 18 cms in diameter. Their propulsion system consists of twelve valves, grouped in pairs of six, that expel pressurized cold gas CO_2 from propellant tanks through the nozzles. The global position and attitude determination system is based on ultrasonic time of flight measurements, while on-board Inertial Measurement Units (IMU) provide acceleration and rate data. [12, 186].

Figure B.2: SPHERES with astronaut Scott Kelly in 2010 [Credit: NASA]

A variety of experiments have been carried out on the SPHERES, testing formation flight, fluid sloshing [187], vision based navigation [188] and control [189], and self-supervised learning [190], to name a few.

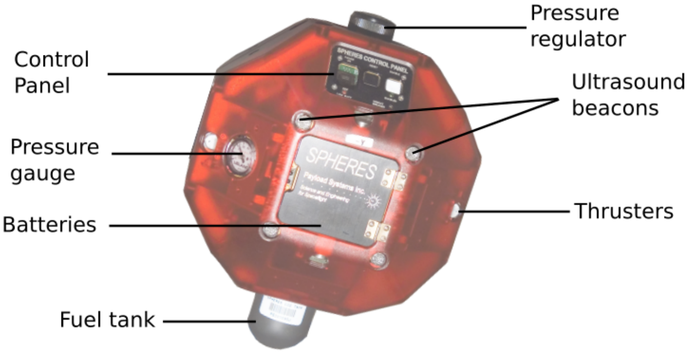


Figure B.3: A SPHERES satellite with its components labeled, based on [191]. [SPHERES image: NASA]

Astrobee

Though the SPHERES are among the most used payloads on the ISS, they required continuous supervision by the crew and assistance for setup, stowing and replacing consumables such as batteries and fuel tanks, in addition to having limited computational bandwidth [10]. Astrobee, NASA's current free-flying robotic system, has rechargeable batteries and is capable of docking, recharging and undocking autonomously. Astrobee is a successor of SPHERES as the space-station's robotic test facility[2] with improved autonomy, computing and sensing capabilities. First launched to the ISS in 2019, the Astrobee project builds on the lessons learned from SPHERES. The cube shaped Astrobee, measuring 32 cm per side [155], is equipped with a perching arm and a station for docking and charging. Its holonomic propulsion system draws in air through two central impellers, which is expelled precisely through 12 exhaust nozzles for thrust [192]. Astrobee uses a combination of visual tracking system, IMU measurements and AR tags for localization. These algorithms run on Astrobee's Mid- and low- level Snapdragon ARM processors, while a high-level Android processor controls interaction devices. Further, Astrobee's flight software stack, along with high-fidelity simulation models of Astrobee's susbsystems is open sourced for the evaluation of guest science algorithms.

Potential tasks for Astrobee include documenting experiments using their video cameras, taking inventory and co-operatively manipulating cargo. Three Astrobee robots have been launched to the ISS (named Honey, Queen and Bumble), the latter two being currently active on station. Among the experiments conducted and planned for Astrobee are testing of a gecko-adhesive inspired gripper [193], tumbling target inspection [194], and formation flight for cooperative payload transportation. Furthermore, as a part of the Integrated System for Autonomous and Adaptive Caretaking (ISAAC) project, Astrobee routinely ventures out of the JEM to U.S. Destiny module[3].

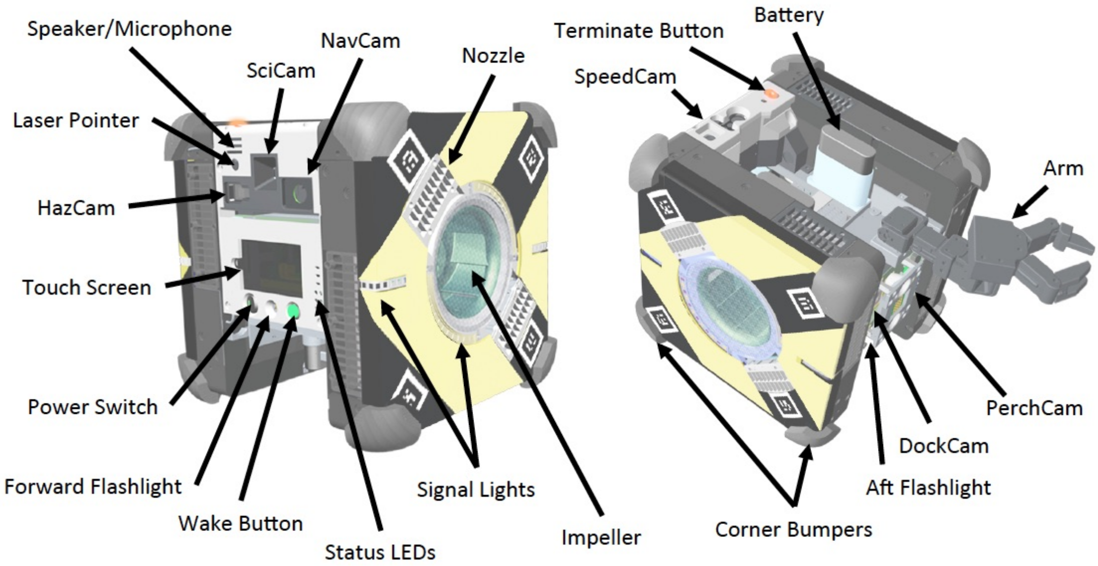


Figure B.4: Astrobee with its components labeled [Credit: NASA]

---

[2] `https://www.nasa.gov/astrobee`

[3] https://www.nasa.gov/collection-asset/isaacastrobee-team-successfully-completes-11th-joint-space-station-activity