Attitude Control and Estimation

by

Yizhou Wang

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering – Mechanical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Masayoshi Tomizuka, Chair
Professor J. Karl Hedrick
Professor Pieter Abbeel

Spring 2015

Attitude Control and Estimation

Abstract

Attitude Control and Estimation

by

Yizhou Wang

Doctor of Philosophy in Engineering – Mechanical Engineering

University of California, Berkeley

Professor Masayoshi Tomizuka, Chair

The attitude control problem, or the control of a spacecraft's orientation with respect to a frame of reference, is a challenging problem in space missions and has attracted much attention as it involves highly nonlinear characteristics of the governing equations. The attitude control task requires an estimation algorithm that deduces the attitude from strapdown sensor inputs and a control algorithm that computes the necessary torques so that the vehicle can follow a desired attitude.

From the perspective of control, feedback control laws are sought for the purpose of asymptotic trajectory tracking, with the ability to reject unexpected external disturbances, and be insensitive to parameter variations. An adaptive sliding mode spacecraft attitude controller that fulfills those requirements is discussed in this dissertation. Unit quaternions and Rodrigues parameters are used to parameterize attitude. Lyapunov stability theory is used to prove the stability of the closed-loop system.

For attitude estimation with increased accuracy, strap-down gyroscopes and vector measurements are fused together. Because of the nonlinear nature of the attitude kinematics equation and the measurement model, the problem becomes a nonlinear state estimation problem, which is typically tackled by Bayesian inference. In this dissertation we discuss a marginalized particle filtering algorithm, to possibly increase the estimation accuracy and reduce the computation load compared with other non-parametric methods. We exploit the linear-substructure and further show that the linear state evolution is completely independent of the nonlinear partition.

We have also investigated a computationally efficient and easy-to-tune sensor fusion algorithm, based on the complementary filter and the TRIAD algorithm. It is beneficial to use a complementary filter because rate and angle sensor possess benefits and drawbacks in different frequency regimes. The proposed algorithm shows comparable performance to the EKF but with less computational burden. It aims to be implementable on a small portable

platform. In applications of mobile robots, the cutoff frequency can be adapted based on a fuzzy logic in real-time to adjust trust to different sensors, to cope with problems such as motion accelerations and magnetic distortions.

To my family

Contents

List of Figures

List of Tables

Acknowledgments

I am indebted to many people who are always accompanying and ready to give their endless support in the past five years along the journey towards the doctorate degree. This dissertation would not have been possible without them.

I would like to thank my research advisor, Professor Masayoshi Tomizuka. Sensei kindly took me into the lab so I have the opportunity to work on research projects under his insightful and patient guidance, to work with many talented labmates who later became great friends. Sensei's knowledge and vision will always inspire me.

I am grateful to Professor O'Reilly, without whom I could have missed the chance to study at Berkeley. I am also appreciative of Professor Hedrick and Professor Abbeel for being on my dissertation committee.

In the early days, I worked under the mentorship of Dr. Hoday Stearns and Dr. Evan Chang-Siu. Later I collaborated closely with Dr. Wenjie Chen and Dr. Xu Chen. I learned not only valuable knowledge, but more importantly passion and meticulous attitude towards research from the seniors.

My labmates have always made the lab a joyful place to work. I will always keep the memory of them: Joonbum Bae, Shu-Wen Yu, Nancy Feng Dan Dong, Sanggyum Kim, Michael Chan, Kan Kanjanapas, Chi-Shen Tsai, Pedro Reynoso, Cong Wang, Wenlong Zhang, Raechel Tan, Minghui Zheng, Junkai Lu, Chung-Yen Lin, Chen-Yu Chan, Changliu Liu, Yaoqiong Du, Kevin Haninger, Xiaowen Yu, Shiying Zhou, Dennis Wai, Shuyang Li, Te Tang, Hsien-Chung Lin, Yu Zhao, Yongxiang Fan, Wei Zhan, Cheng Peng, Daisuke, Kaneishi.

I would like to thank Dr. Mohammed Almajed and Dr. Badr Alsuwaidan with National Satellite Technology Program at King Abdulaziz City for Science and Technology (KACST) for sponsoring our project and providing valuable advice.

Finally, I would like to thank my wife Jinghong Qian, my parents and parents-in-law for their love and sacrifices. You are the meaning of my life.

Chapter 1

Introduction

1.1 Background

The first spacecraft can be traced back to Russian's Sputnik 1, which entered space on October 4th, 1957. Since then, thousands of spacecraft, including Earth satellite and deep-space probes, have been launched to travel and operate outside Earth's atmosphere.

One challenging issue is *attitude control*, or the control of a spacecraft's orientation with respect to a frame of reference. The attitude must be controlled for various reasons [1]:

- the antenna needs to be aligned accurately to receivers on Earth or other satellites for communication
- the onboard experiments may require precise pointing for accurate collection and interpretation of data
- the heating and cooling effects of sunlight and shadow may be fully taken advantage of for thermal control
- short propulsive maneuvers may be executed in the right direction for guidance

The attitude control task requires

1. *sensors* that provide useful orientation-related information
2. *estimation algorithms* that deduce the attitude from sensor inputs
3. *control algorithms* that compute the necessary torques so that the vehicle can follow a desired attitude
4. *actuators* that apply the computed torques to re-orient the vehicle

A block diagram of the general attitude control problem is shown in Figure 1.1.

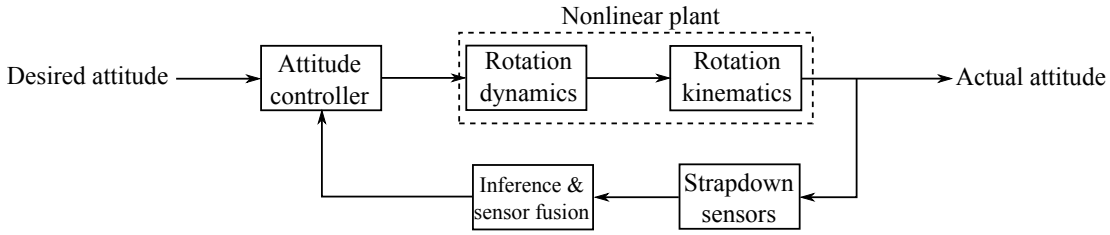


Figure 1.1: A general attitude control system block diagram

1.2 Motivation and Contribution

1.2.1 Attitude Control

The attitude control problem has attracted much attention as it involves highly nonlinear characteristics of the governing motion equations. From the perspective of control, feedback control laws are sought for the purpose of asymptotic trajectory following, with the ability to reject unexpected external disturbances, and be insensitive to parameter variations.

The simplest large-angle maneuver uses quaternion and velocity feedback similar to a proportional derivative controller [2]. Model-based control techniques are also investigated such as sliding-mode control [3] and adaptive control [4]. Sliding-mode control was investigated for the purpose of robust attitude following for various attitude parameterizations (Rodrigues parameters [5, 6], Modified Rodrigues parameters [7], quaternions [8, 9, 10]). Adaptive attitude tracking control based on Lyapunov stability was studied for quaternions [11, 12] and rotation matrices [13].

An adaptive sliding-mode attitude tracking controller is studied in this thesis [14]. A similar methodology has been applied to control of robot manipulators by Slotine *et. al.* [15]. We use two approaches to synthesize the reaching law, the direct approach and the Hamiltonian approach. In the direct approach, the controller is directly constructed such that the Lie derivative of the Lyapunov function candidate is negative semi-definite. In the Hamiltonian approach [16], the plant model is first cast into the form of robotic manipulators, then the adaptive controller is designed using the exact same method for robots. Rodrigues parameters and unit quaternions are used to parameterize rotations. The advantage of the latter is that it is the minimal singularity-free rotation representation. Based on the quaternion kinematic relation, a sliding manifold is chosen according to the optimality criterion proposed in [17].

Compared with the existing methodologies in the attitude control literature, the proposed controller in this thesis takes a simpler form. The asymptotic (robust) stability of attitude following is rigorously proved. No prior information of the inertial parameters is required.

Furthermore, one important contribution of this work is showing the equivalence of the two approaches. It is proved that the resulting Lyapunov function is exactly the same (with properly chosen controller parameters), even though the two controllers generate two different closed-loop trajectories.

In the author's opinion, attitude control literature is hard to understand due to the extensive use of rotation parameterizations (some of which may be less familiar to a general audience) and their associated properties. To reduce the difficulty in reading, this dissertation is made relatively self-contained and the framework used here should be straightforward to understand.

1.2.2 Attitude Estimation

Attitude estimation can be traced back to 1965 when Wahba posed the question to estimate the attitude of a spacecraft in the sense of least squares given noisy observations [18], resolved in the body frame and the reference frame. Some well-known estimators that robustly solve Wahba's problem are Davenport's q method [19], and the QUaternion ESTimator (QUEST) [20].

For attitude estimation with increased accuracy, strap-down gyroscopes are used in combination with vector measurements. Because of the nonlinear nature of the attitude kinematics equation and the measurement model, the problem becomes a nonlinear state estimation problem, which is typically tackled by Bayesian inference. The extended Kalman filter (EKF) was first studied in attitude estimation and in particular, Lefferts *et. al.* proposed a multiplicative error approach in which the error quaternion and the gyroscope bias are defined as the filter states [21, 22]. The drawback of EKF is that the mean and covariance of the state is propagated analytically through the first-order linearization of the nonlinear dynamics, which may introduce large approximation errors and lead to sub-optimal filter performance. The unscented Kalman filter (UKF) is superior to EKF in terms of capturing the posterior mean and covariance of the state distribution (accurately up to the 3rd order) [23]. The UKF formulation for attitude estimation has been proposed by Crassidis *et. al.*[24].

Marginalized particle filter
Due to recent development in computational power, the use of particle filters (PF) gained much traction and became practical for a broad area of applications. PF computes the posterior state distribution by drawing random samples of the state vector (termed as particles) and evaluates the likelihood of getting the actual system measurements conditioned on each particle [25]. Cheng *et. al.* applied a bootstrap particle filter for sequential spacecraft attitude estimation [26]. Because of the high dimensionality of the state vector, a prohibitively large number of particles are needed to span the state space to support the state distribution. In contrast to this approach, Oshman *et. al.* reduces the computational burden by sampling only the attitude of the spacecraft and using a genetic algorithm to estimate the gyro bias

[27].

If there exists a linear sub-structure inherent in the nonlinear dynamics, it is possible to marginalize out the linear state variables and estimate them instead with the Kalman filter (KF) while the nonlinear state variables are estimated using the PF. This powerful combination of PF and KF, called the marginalized particle filter (MPF) or the Rao-Blackwellized particle filter, can effectively increase the estimation accuracy and possibly reduce the computation load [28]. The scheme has been directly applied for attitude estimation by Liu *et. al.* [29]. This dissertation contributes is to further exploit the underlying linear-substructure, and show that the linear state evolution is completely independent of the nonlinear part [30].

Complementary filter
In contrast to Bayesian filters, which are optimal under the assumption of some probabilistic models but are usually computationally demanding, we seek an efficient, easy-to-tune sensor fusion algorithm.

By combining two computationally efficient schemes, the TRIAD algorithm and the time-varying complementary filter (TVCF), we present an algorithm that shows comparable performance to the EKF with less computional burden [31]. It aims to be implementable on a small portable platform with low computational power (e.g. inertial measurement units). This application can benefit from complementary filtering because rate and angle sensors (gyroscope versus accelerometer and magnetometer) possess benefits and drawbacks in different frequency regimes [32, 33, 34]. The TVCF, which uses a fuzzy logic scheme to adjust trust to different sensors, has already been found to be useful for 1-DOF attitude estimation due to its low computational requirement and the ability to discern stationary and motion states [35]. This dissertation's contribution is to first generalize the previous work to 3-DOF attitude estimation. It is shown in the simulation that when the angle measurements are from a star camera, the complementary filtering method with a tuned fixed cutoff frequency also shows comparable performance to the EKF.

1.3 Dissertation Outline

The overall structure of this dissertation is depicted in Figure 1.2. More specifically, this dissertation is organized as follows.

Chapter 2 presents various attitude representations and associated kinematics, including rotation matrices, Euler angles, unit quaternions, Rodrigues parameters, which are used throughout the entire dissertation. Chapter 4 presents the mathematical models of the essential strapdown sensors for attitude estimation, including gyroscopes, accelerometers,

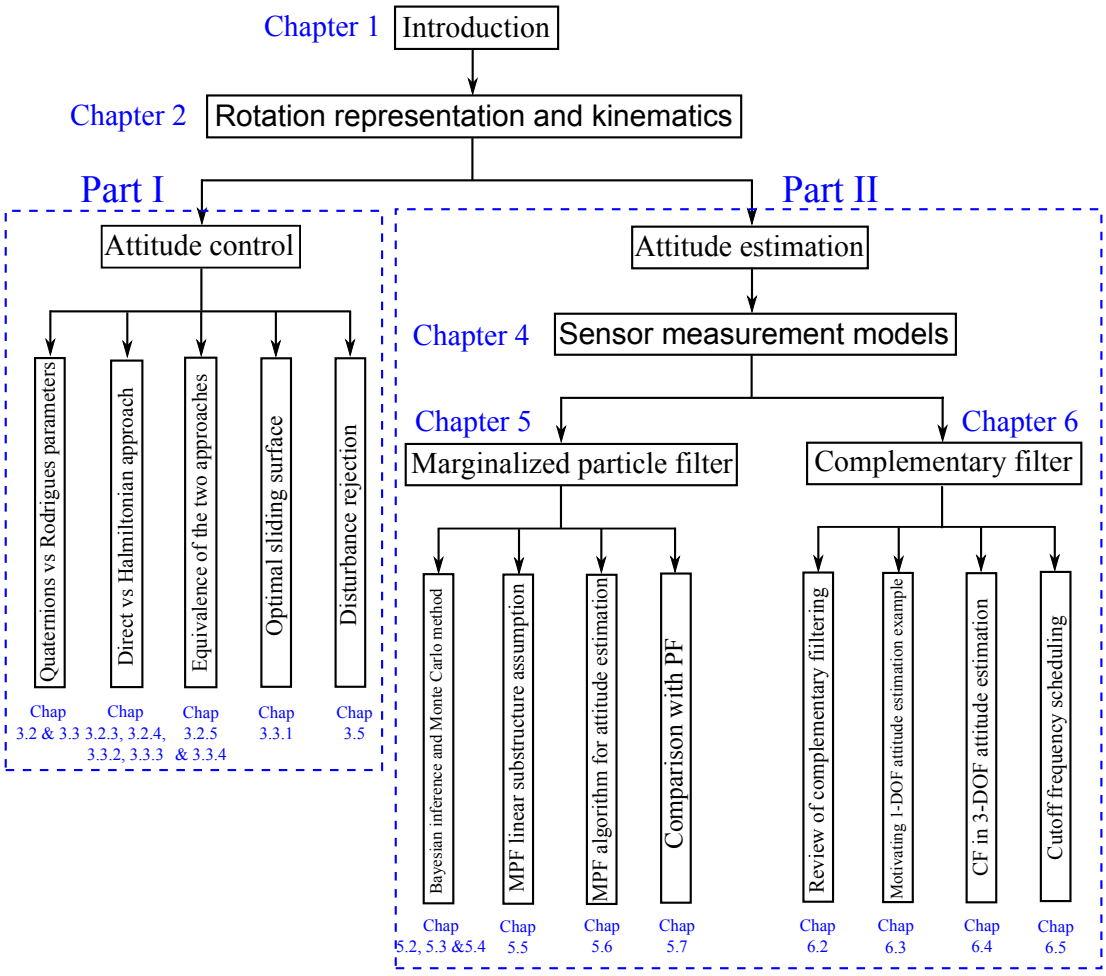


Figure 1.2: Anatomy of the dissertation

magnetometers, star cameras. These models will be referred to in the development of the proposed attitude estimation algorithms.

Details about my major contributions will be presented in two parts. In Part I, the attitude control problem is solved using the idea of sliding mode control and Lyapunov stability analysis in Chapter 3. In Part II, the attitude estimation problem is tackled using two approaches, that is, the marginalized particle filter in Chapter 5 and the complementary filter in Chapter 6.

Chapter 2

Attitude Representations

This chapter gives a basic review of the attitude representations for the development of the rest of the thesis. A much detailed presentation can by found in a survey paper by Shuster [36].

2.1 Rotation Matrices

A general expression of the rotation matrix R, in terms of an arbitrary axis of rotation r and an angle of rotation θ, can be written using the *Rodrigues' rotation formula*

$$R(r,\theta) = \cos\theta I_{3\times 3} + (1-\cos\theta)rr^T - \sin(\theta)[r\times] \tag{2.1}$$

where $[\bullet\times] : \mathbb{R}^3 \mapsto \mathbb{R}^{3\times 3}$ denotes a cross-product matrix. The cross product between two three-dimensional vectors can be written as a matrix-vector multiplication

$$u \times v = [u\times]v = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}, \quad \forall u, v \in \mathbb{R}^3 \tag{2.2}$$

Rotation matrices are orthogonal and proper

$$RR^T = R^T R = I_{3\times 3}, \quad \det(R) = 1 \tag{2.3}$$

Hence the inverse rotation of R is its transpose R^T.

Rotation matrices are useful as they map the representation of a vector from the reference frame to the body frame

$$b = Rr \tag{2.4}$$

where $b = [b_x,\, b_y,\, b_z]^T$ represents a vector expressed in the body frame and $r = [r_x,\, r_y,\, r_z]^T$ is the *same* vector but expressed in the reference frame. In the field of robotics,the convention is to use the transpose of R defined here to represent rotation.

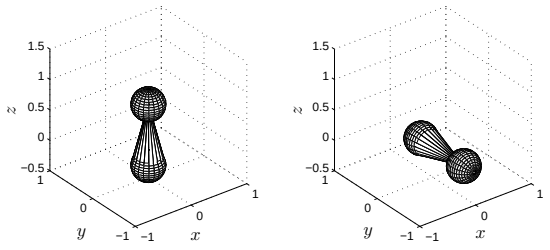
The rotation kinematics equation is given by

$$\dot{R} = -[\omega\times]R \tag{2.5}$$

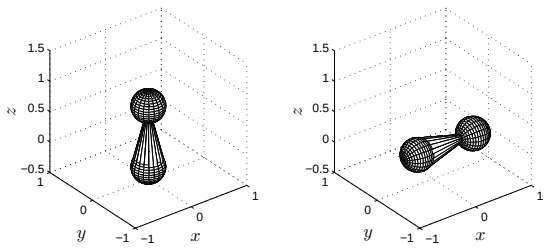
where ω is the angular velocity of the body frame relative to the reference frame.

2.2 Euler Angles

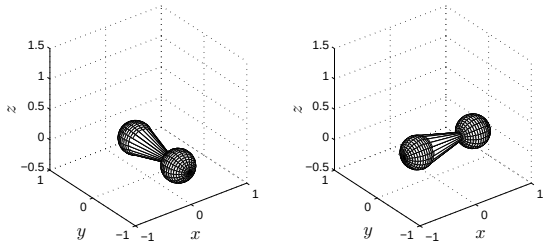
A general rotation is usually broke down into three consecutive fundamental rotations about body axes. The fundamental rotations are the ones about the body x, y, z axes, illustrated in Figure 2.1. The corresponding rotation matrices are given by



(a) rotation about x-axis by 90^o



(b) rotation about y-axis by 90^o



(c) rotation about z-axis by 90^o

Figure 2.1: Illustration of three fundamental rotations.

$$
\begin{aligned}
R(x,\phi) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \\
R(y,\theta) &= \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \\
R(z,\psi) &= \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{2.6}
$$

Composition of rotations is represented by pre-multiplying the rotation matrix of the i-th rotation by the $(i+1)$-th rotation, i.e.

$$
\begin{aligned}
&\mathcal{F}'' \xleftarrow{R(r_2,\theta_2)} \mathcal{F}' \xleftarrow{R(r_1,\theta_1)} \mathcal{F} \\
&R(r,\theta) = R(r_2,\theta_2)\cdot R(r_1,\theta_1)
\end{aligned} \tag{2.7}
$$

where $R(r,\theta)$ parameterizes the overall rotation, from $\mathcal{F}$ to $\mathcal{F}''$. Note that rotation matrices are generally not commutative. We use this composition rule to compute the overall rotation of three successive fundamental rotations. For example, if 3-2-1 Euler angles $\{\psi,\theta,\phi\}$ are selected, the overall rotation is then

$$
R(z,y,x;\psi,\theta,\phi) = R(x,\phi)\cdot R(y,\theta)\cdot R(z,\psi) \tag{2.8}
$$

There are twelve possible sets of Euler angles: six symmetric sets, e.g. 1-3-1, 3-1-3, and six asymmetric sets, e.g. 3-2-1, 1-2-3.

2.3 Unit Quaternions

Using the double-angle formulas

$$
\sin\theta = 2\sin(\theta/2)\cos(\theta/2), \quad \cos\theta = \cos^2(\theta/2) - \sin^2(\theta/2) = 1 - 2\sin^2(\theta/2) \tag{2.9}
$$

Equation (2.1) can be rewritten as,

$$
\begin{aligned}
R(r,\theta) =& (\cos^2(\theta/2) - \sin^2(\theta/2))I_{3\times 3} + 2(\sin(\theta/2)r)(\sin(\theta/2)r)^T \\
&- 2\cos(\theta/2)[\sin(\theta/2)r\times] \\
=& (q_4^2 - ||\rho||_2^2)I_{3\times 3} + 2\rho\rho^T - 2q_4[\rho\times] \\
=& \Xi(q)^T\Phi(q) \\
=& R(q)
\end{aligned} \tag{2.10}
$$

In the above expression, the rotation matrix is expressed in terms of a unit quaternion q, defined as

$$
q = \begin{bmatrix} \rho \\ q_4 \end{bmatrix} \triangleq \begin{bmatrix} \sin(\theta/2)r \\ \cos(\theta/2) \end{bmatrix} \tag{2.11}
$$

$\Xi(q)$ and $\Phi(q)$ are defined as

$$\Xi(q) = \begin{bmatrix} q_4 I_{3\times 3} + [\rho\times] \\ -\rho^T \end{bmatrix}, \quad \Phi(q) = \begin{bmatrix} q_4 I_{3\times 3} - [\rho\times] \\ -\rho^T \end{bmatrix} \tag{2.12}$$

A unit quaternion satisfies the unity-norm constraint

$$||q||_2 = \sqrt{\sin^2(\theta/2) + \cos^2(\theta/2)} = 1 \tag{2.13}$$

since it uses four parameters to describe three degrees of freedom.

Directly from the definition, it is easy to see that q and $-q$ represent the same physical rotation because one can think of $-q$ representing a rotation of $\theta + 2\pi$ angle along the rotation axis r. One implication of this is that arithmetic mean is not well-defined in quaternion space. To see this, consider $q_{ave} = (q + (-q))/2 = [0,\ 0,\ 0,\ 0]$, which is not a unit quaternion. The actual average of q and $-q$ should obviously be either q or $-q$. Another complication is that subtraction is not a good way to represent (tracking) errors. Instead, the error is computed using quaternion multiplication, discussed later.

The quaternion kinematics equation is given by

$$\dot{q} = \frac{1}{2}\Xi(q)\omega = \frac{1}{2}\Omega(\omega)q \tag{2.14}$$

where

$$\Omega(\omega) = \begin{bmatrix} -[\omega\times] & \omega \\ -\omega^T & 0 \end{bmatrix} \tag{2.15}$$

Observe that the kinematics is free of singularities as well as bilinear in q and ω. In other words, it is linear in q for fixed ω and vice versa.

The matrix $\Xi(q)$ obeys the following relations:

$$\Xi^T(q)\Xi(q) = I_{3\times 3} \tag{2.16a}$$

$$\Xi(q)\Xi^T(q) = I_{4\times 4} - qq^T \tag{2.16b}$$

$$\Xi^T(q)q = 0_{3\times 1}, \quad \forall q \in \mathbb{R}^4 \tag{2.16c}$$

$$\Xi^T(q)\lambda = -\Xi^T(\lambda)q, \quad \forall q, \lambda \in \mathbb{R}^4 \tag{2.16d}$$

$$\Phi(q)\omega = \Gamma(\omega)q, \quad \Gamma(\omega) = \begin{bmatrix} [\omega\times] & \omega \\ -\omega^T & 0 \end{bmatrix}, \quad \forall q \in \mathbb{R}^4, \forall \omega \in \mathbb{R}^3 \tag{2.16e}$$

Note that the last three equalities do not require unity norm constraint on the variables. The third property is intuitive from the norm conservation

$$\frac{d}{dt}(\underbrace{q^T q}_{=1}) = 0 = \frac{1}{2}q^T\Xi(q)\omega, \quad \forall \omega \tag{2.17}$$

Successive rotations can be accomplished using quaternion multiplication, which is also bilinear

$$q_2 \otimes q_1 \triangleq [\Phi(q_2) \vdots q_2]q_1 = [\Xi(q_1) \vdots q_1]q_2 \tag{2.18}$$

where q_2 is the rotation applied after q_1. This convention is adopted here because the order of operation is consistent with that of rotation matrices, where the second rotation pre-multiplies the first rotation

$$R(q_2)R(q_1) = R(q_2 \otimes q_1) \tag{2.19}$$

It can be checked that an alternative way to write the kinematics equation is

$$\dot{q} = \frac{1}{2}\Omega(\omega)q = \frac{1}{2}\begin{bmatrix}\omega \\ 0\end{bmatrix} \otimes q \tag{2.20}$$

The inverse quaternion is defined as

$$q^{-1} = \begin{bmatrix}-\rho \\ q_4\end{bmatrix} \tag{2.21}$$

Note that $q \otimes q^{-1} = q^{-1} \otimes q = [0,\ 0,\ 0,\ 1]^T$, which is the identity quaternion.

2.4 Rodrigues Parameters

Rodrigues parameters (Gibbs vector) is a minimal attitude representation (but not free of singularities) closely related to unit quaternions. It is defined as

$$p \triangleq \tan(\theta/2)r = \rho/q_4 \tag{2.22}$$

Note that rotations with $\theta = (2k+1)\pi, \forall k \in \mathbb{Z}$ are not defined using Rodrigues parameters. However, in this representation, subtraction can be used to represent errors, as $p_e = p_1 - p_2 = 0_{3\times 1}$ means that the two rotations represented by p_1 and p_2 physically coincide.

Successive rotations can be computed as

$$p_2 \otimes p_1 = \frac{p_2 + p_1 - p_2 \times p_1}{1 - p_2^T p_1} \tag{2.23}$$

which does not possess the bilinearity property like unit quaternions.

The kinematics associated with Rodrigues parameters can be written as

$$\dot{p} = T(p)\omega, \quad \text{where } T(p) = \frac{1}{2}(I_{3\times 3} + [p\times] + pp^T) \tag{2.24}$$

One can observe that $T(p)$ is positive-definite regardless of p. In fact

$$T(p) - \frac{1}{2} I_{3\times 3} \succeq 0 \tag{2.25}$$

Hence, the inverse matrix always exists, and is given by

$$T^{-1}(p) = \frac{2}{1 + p^T p}(I_{3\times 3} - [p\times]) \tag{2.26}$$

Chapter 3

Adaptive Sliding Mode Attitude Control

An adaptive sliding mode attitude controller is derived in this chapter. It has the advantage of not requiring knowledge of the inertia of the spacecraft and rejecting unexpected external disturbances, with global asymptotic position and velocity tracking.

We use both Rodrigues parameters and quaternions to parameterize rotations. With Rodrigues parameters, the tracking error is simply the arithmetic difference. With quaternions, the error must be quantified using quaternion composition rule. Furthermore, the sliding manifold can be designed using optimal control analysis of the kinematics, trading off error (which is physically meaningful in this case) and control effort.

The sliding mode control law and the parameter adaptation law are designed using Lyapunov stability theory. We use two approaches to design the reaching law: the direct approach and the Hamiltonian approach. We prove that the resulting controllers yield the same Lyapunov function but different closed-loop trajectories. Numerical simulations are performed to demonstrate both the nominal and the robust performance.

3.1 Introduction

As discussed in Chapter 1, feedback control laws are sought for the purpose of asymptotic trajectory tracking, with the ability to reject unexpected external disturbances, and be insensitive to parameter variations. The presence of highly nonlinear characteristics of the governing motion equations complicates the controller synthesis process. In this chapter, an adaptive sliding-mode attitude tracking controller is discussed. There are two attitude representations under consideration, and two approaches are used to synthesize the controller. Furthermore, we will show the equivalence of the two approaches.

The remainder of the chapter is organized as follows. In Sec. 3.2 and Sec. 3.3, the adaptive controller is designed using Rodrigues parameters and unit quaternions respectively. In each section, a sliding manifold is designed. In the case of quaternions, an optimality criterion is proposed. Lyapunov stability analysis is then used to derive an asymptotically stabilizing sliding control law and a parameter adaptation law under two different approaches. Equivalence between the two approaches are discussed. Numerical simulations are shown to demonstrate the closed-loop performance of the proposed controller. In Sec. 3.4, the sufficient condition for parameter convergence is stated. In Sec. 3.5, a robust controller is designed to reject unexpected external disturbances.

3.2 Rodrigues Parameters

In the following section, we discuss the attitude controller when the attitude is parameterized by Rodrigues Parameters. The state-space equation of the plant is given by

$$\text{plant}: \quad \begin{cases} \dot{p} = T(p)\omega & \text{kinematic equation} \\ J\dot{\omega} = -[\omega\times]J\omega + \tau & \text{dynamic equation} \end{cases} \tag{3.1}$$

Assume that the desired attitude profile is given by $p_d(t)$ which is at least twice-differentiable. The control objective is to design a feedback controller τ

$$\tau = \tau(p, \omega, p_d, \dot{p}_d, \ddot{p}_d) \tag{3.2}$$

so that $p \to p_d$ as $t \to \infty$.

We apply the sliding mode control for controller synthesis. The idea of sliding mode control is to allow the transformation of a controller design problem for a general n-th order system to a simple stabilization problem with reduced order. That is, we stabilize the dynamics associated with the switching function. Then for the equivalent reduced-order system, intuitive feedback control strategies can be applied.

Sliding mode controller design consists of the following two steps:

1. design a *stable* sliding manifold on which the control objective is achieved
2. design a reaching law and the corresponding control input so that the switching function is attracted to 0

3.2.1 Sliding manifold

The formal definition of the stability of a sliding manifold/switching function is given as follows.

Definition 3.1. *A switching function s is stable, if the control objective is achieved when the state variables are confined in the sliding manifold $s = 0$ as $t \to \infty$.*

We consider the following three switching functions.

Proposition 3.1. *The switching function s_1 defined by*

$$s_1 = \omega - \omega_r, \quad \text{where } \omega_r = T^{-1}(p)(\dot{p}_d - \Lambda(p - p_d)) \tag{3.3}$$

is stable if Λ is positive-definite. ω_r is called the virtual velocity reference.

Proof. Define the (additive) tracking error

$$e = p - p_d \tag{3.4}$$

On the sliding manifold

$$\begin{aligned} & s_1 = 0 \\ \Rightarrow \quad & T(p)s_1 \\ & = \dot{p} - \dot{p}_d + \Lambda(p - p_d) \\ & = \dot{e} + \Lambda e \\ & = 0 \end{aligned} \tag{3.5}$$

Because of the positive-definiteness of Λ, we conclude the exponential convergence of e to 0. ■

Proposition 3.2. *The switching function s_2 defined by [16]*

$$s_2 = \dot{p} - \dot{p}_r, \quad \text{where } \dot{p}_r = \dot{p}_d - \Lambda(p - p_d) \tag{3.6}$$

is also stable. $\dot{p}_r$ is also a virtual velocity reference, but represented differently from the previous one. Furthermore, the sliding manifolds of s_1 and s_2 are equivalent since $T(p)$ is positive-definite regardless of p

$$s_2 = T(p)s_1 = 0 \quad \Rightarrow \quad s_1 = 0 \tag{3.7}$$

and vice versa.

Proof. Same as the proof of **Proposition 2.1**. ■

Proposition 3.3. *The switching function s_3 defined by [6]*

$$s_3 = \omega - T^{-1}(p)\dot{p}_d + \Lambda(p - p_d) \tag{3.8}$$

is also stable.

Proof. On the sliding manifold,

$$\begin{aligned} s_3 = 0 \quad \Rightarrow \quad & T(p)s_3 \\ & = \dot{p} - \dot{p}_d + T(p)\Lambda(p - p_d) \\ & = \dot{e} + T(p)\Lambda e \\ & = 0 \end{aligned} \tag{3.9}$$

Consider a candidate Lyapunov function $V = \frac{1}{2}e^T\Lambda e$, which is positive-definite in e. The Lie derivative on the manifold is

$$\begin{aligned} \dot{V} &= e^T\Lambda\dot{e} \\ &= -e^T\Lambda T(p)\Lambda e \end{aligned} \tag{3.10}$$

where we used the error dynamics in the last equality. Since $T(p) \succeq \frac{1}{2}I_{3\times 3}$, we further have

$$\dot{V} \leq -\frac{1}{2}||\Lambda e||_2^2 \leq 0 \tag{3.11}$$

That is, $\dot{V}$ is negative-definite in e. The tracking error e converges to 0 as $t \to \infty$. ■

3.2.2 Linearity in system parameters

In the dynamic equation (Equation (3.1)), we observe that the inertial parameters $J_{ij}, i, j = 1, 2, 3$ appear linearly. To make this more explicit, we follow [11] to use the following linear operator $\mathcal{L} : \mathbb{R}^3 \mapsto \mathbb{R}^{3\times 6}$ acting on any three-dimensional vector $\omega = [\omega_1,\ \omega_2,\ \omega_3]^T$ by

$$\mathcal{L}(\omega) = \begin{bmatrix} \omega_1 & 0 & 0 & 0 & \omega_3 & \omega_2 \\ 0 & \omega_2 & 0 & \omega_3 & 0 & \omega_1 \\ 0 & 0 & \omega_3 & \omega_2 & \omega_1 & 0 \end{bmatrix} \tag{3.12}$$

For any $J = J^T$, it follows easily that

$$\underbrace{\begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{12} & J_{22} & J_{23} \\ J_{13} & J_{23} & J_{33} \end{bmatrix}}_{=J} \omega = \mathcal{L}(\omega) \underbrace{\begin{bmatrix} J_{11} \\ J_{22} \\ J_{33} \\ J_{23} \\ J_{13} \\ J_{12} \end{bmatrix}}_{\triangleq a} \tag{3.13}$$

where we define a six-dimensional vector a of the inertial parameters J_{ij}. We can further denote the parameter estimate and the estimation error by $\hat{a}$ and $\tilde{a}$ respectively

$$\tilde{a}(t) = \hat{a}(t) - a \tag{3.14}$$

In the following two subsections, we present two approaches for the design of reaching laws, (i) the direct approach and (ii) the Hamiltonian approach. Later, we will also show that the resulting reaching laws are closely related to each other.

3.2.3 Direct approach

The reaching law derived from the direct approach is given in the following proposition.

Proposition 3.4. *The following control law*

$$\tau_1 = Y\hat{a}_1 - K_d s_1 \tag{3.15}$$

and the adaptation law

$$\dot{\hat{a}}_1 = -\Gamma^{-1} Y^T s_1 \tag{3.16}$$

yields a globally stable adaptive controller, where Γ *and* K_d *are positive-definite matrices and* $Y \in \mathbb{R}^{3\times 6}$ *is given by*

$$Y(p, \dot{p}, p_d, \dot{p}_d, \ddot{p}_d) = [\omega\times]\mathcal{L}(\omega) + \mathcal{L}(\dot{\omega}_r) \tag{3.17}$$

where

$$\begin{aligned} \dot{\omega}_r &= -T^{-1}\dot{T}T^{-1}(\dot{p}_d - \Lambda(p - p_d)) + T^{-1}(\ddot{p}_d - \Lambda(\dot{p} - \dot{p}_d)) \\ \dot{T} &= \frac{1}{2}([\dot{p}\times] + \dot{p}p^T + p\dot{p}^T) \end{aligned} \tag{3.18}$$

Proof. Consider a Lyapunov function candidate

$$V(t) = \frac{1}{2}s_1^T J s_1 + \frac{1}{2}\tilde{a}_1^T \Gamma \tilde{a}_1 \tag{3.19}$$

which is positive-definite in s and $\tilde{a}_1$. The Lie derivative $\dot{V}(t)$ yields

$$\begin{aligned} \dot{V}(t) &= s_1^T J\dot{s}_1 + \tilde{a}_1^T \Gamma \dot{\tilde{a}}_1 \\ &= s_1^T J(\dot{\omega} - \dot{\omega}_r) + \tilde{a}_1^T \Gamma \dot{\hat{a}}_1 \\ &= s_1^T(-[\omega\times]J\omega + \tau - J\dot{\omega}_r) + \tilde{a}_1^T \Gamma \dot{\hat{a}}_1 \\ &= s_1^T(-[\omega\times]\mathcal{L}(\omega)a - \mathcal{L}(\dot{\omega}_r)a + \tau) + \tilde{a}_1^T \Gamma \dot{\hat{a}}_1 \\ &= s_1^T(-Ya + \tau) + \tilde{a}_1^T \Gamma \dot{\hat{a}}_1 \end{aligned} \tag{3.20}$$

Now substitute in the control law (Equation (3.15)) and the parameter adaptation law (Equation (3.16)). We have

$$\begin{aligned} \dot{V} &= s_1^T(Y\tilde{a}_1 - K_d s_1) - \tilde{a}_1^T \Gamma \Gamma^{-1} Y^T s_1 \\ &= -s_1^T K_d s_1 \\ &\leq 0 \end{aligned} \tag{3.21}$$

Hence $\dot{V}$ is negative semi-definite in s_1. Recognizing the boundedness of $\ddot{V}$ and the uniform continuity of $\dot{V}$, we invoke Barbalat's Lemma to conclude

$$\lim_{t\to 0} \dot{V} = 0 \quad \Rightarrow \quad \lim_{t\to 0} s_1 = 0 \tag{3.22}$$

Therefore, the proposed adaptive controller is globally stable. ■

3.2.4 Hamiltonian approach

Alternatively, we can derive an adaptive controller which is similar to the controller synthesis for robot manipulators, hence called the Hamiltonian approach.

Proposition 3.5. *The state-space equation of the plant can be cast into the form of a second-order differential equation in p*

$$H(p)\ddot{p} + C(p,\dot{p})\dot{p} = F \tag{3.23}$$

where $p, H(p), C(p,\dot{p}), F$ resemble the joint displacement, the manipulator inertia matrix, the Coriolis matrix, and the joint torques respectively.

$$\begin{aligned} H(p) &= T^{-T}JT^{-1} \\ C(p,\dot{p}) &= -T^{-T}JT^{-1}\dot{T}T^{-1} - T^{-T}[J\omega\times]T^{-1} \\ F &= T^{-T}\tau \end{aligned} \tag{3.24}$$

Proof. Differentiating the kinematics equation to get

$$\ddot{p} = \dot{T}\omega + T\dot{\omega} \tag{3.25}$$

Pre-multiplying JT^{-1} on both sides and substitute in the dynamic equation

$$JT^{-1}\ddot{p} + (-JT^{-1}\dot{T} - [J\omega\times])\omega = \tau \tag{3.26}$$

Finally, pre-multiplying T^{-T} and replacing ω by $T^{-1}\dot{p}$ yields

$$\underbrace{T^{-T}JT^{-1}}_{H(p)}\ddot{p} + \underbrace{(-T^{-T}JT^{-1}\dot{T}T^{-1} - T^{-T}[J\omega\times]T^{-1})}_{C(p,\dot{p})}\dot{p} = \underbrace{T^{-T}\tau}_{F} \tag{3.27}$$

■

Just as already pointed out in research literatures of robotic manipulator control, we have two important properties associated with this new system equation described below.

Lemma 3.1. *The matrix $\dot{H}(p) - 2C(p,\dot{p})$ is skew-symmetric.*

Proof. The proof is done by direct algebraic manipulation. Note $\frac{d}{dt}(T^{-1}) = -T^{-1}\dot{T}T^{-1}$.

$$\begin{aligned} \dot{H} - 2C &= \frac{d}{dt}(T^{-T}JT^{-1}) - 2T^{-T}J\frac{d}{dt}(T^{-1}) + 2T^{-T}[J\omega\times]T^{-1} \\ &= \underbrace{\frac{d}{dt}(T^{-T})JT^{-1} - T^{-T}J\frac{d}{dt}(T^{-1})}_{\text{skew-symmetric}} + 2T^{-T}\cdot\underbrace{[J\omega\times]}_{\text{skew-symmetric}}\cdot T^{-1} \end{aligned} \tag{3.28}$$

■

Lemma 3.2. *The new state equation is still linear in terms of the inertial parameters J. In other words, one can write*

$$H\ddot{p}_r + C\dot{p}_r = \bar{Y}a \tag{3.29}$$

where $a = [J_{11}, J_{22}, J_{33}, J_{23}, J_{13}, J_{12}]^T$. The expression of $\bar{Y}$ is given by

$$\bar{Y} = T^{-T}\big[\mathcal{L}(\dot{\omega}_r) + [\omega_r\times]\mathcal{L}(\omega)\big] \tag{3.30}$$

Proof.

$$\begin{aligned} H\ddot{p}_r + C\dot{p}_r &= T^{-T}JT^{-1}\ddot{p}_r - T^{-T}JT^{-1}\dot{T}T^{-1}\dot{p}_r - T^{-T}[J\omega\times]T^{-1}\dot{p}_r \\ &= T^{-T}JT^{-1}\frac{d}{dt}(T\omega_r) - T^{-T}JT^{-1}\dot{T}\omega_r - T^{-T}[J\omega\times]\omega_r \\ &= T^{-T}J\dot{\omega}_r + T^{-T}[\omega_r\times]J\omega \\ &= T^{-T}\big[\mathcal{L}(\dot{\omega}_r) + [\omega_r\times]\mathcal{L}(\omega)\big]a \\ &\triangleq \bar{Y}a \end{aligned} \tag{3.31}$$

Hence we get the expression of $\bar{Y}$, which resembles Y. Recall

$$Y = [\omega\times]\mathcal{L}(\omega) + \mathcal{L}(\dot{\omega}_r) \tag{3.32}$$

■

The reaching law derived from the Hamiltonian approach is given in the following proposition.

Proposition 3.6. *The following control law*

$$\begin{aligned} F &= \bar{Y}\hat{a}_2 - \bar{K}_d s_2 \\ \tau_2 &= T^T F \end{aligned} \tag{3.33}$$

and the adaptation law

$$\dot{\hat{a}}_2 = -\Gamma^{-1}\bar{Y}^T s_2 \tag{3.34}$$

yields a global stable adaptive controller, where $\bar{Y}$ is given by Equation (3.30)*, $\bar{K}_d$ is positive-definite.*

Proof. Consider a candidate Lyapunov function $V(t) = \frac{1}{2}s_2^T H s_2 + \frac{1}{2}\tilde{a}_2^T\Gamma\tilde{a}_2$. The Lie derivative is

$$\begin{aligned} \dot{V}(t) &= s_2^T(\frac{1}{2}\dot{H}s_2 + H\dot{s}_2) + \tilde{a}_2^T\Gamma\dot{\tilde{a}}_2 \\ &= \underbrace{s_2^T(\frac{1}{2}\dot{H}s_2 - Cs_2}_{=0} + Cs_2 + H\dot{s}_2) + \tilde{a}_2^T\Gamma\dot{\hat{a}}_2 \end{aligned} \tag{3.35}$$

The first two terms are zero because of the skew-symmetry of $\dot{H} - 2C$. Then we combine the definition $s_2 = \dot{p} - \dot{p}_r$, the new system equation $H\ddot{p} + C\dot{p} = F$, the control input (Equation

(3.33)), the parameter adaptation (Equation (3.34)) and the linearity property (Equation (3.29))

$$\begin{aligned}\dot{V}(t) &= s_2^T(F - H\ddot{p}_r - C\dot{p}_r) + \tilde{a}_2^T\Gamma\dot{\hat{a}}_2 \\ &= s_2^T(\bar{Y}\hat{a}_2 - \bar{K}_d s_2 - \bar{Y}a) - \tilde{a}_2^T\Gamma\Gamma^{-1}\bar{Y}^T s_2 \\ &= -s_2^T\bar{K}_d s_2 \\ &\leq 0\end{aligned} \tag{3.36}$$

Again, we prove $s_2 \to 0$ as $t \to 0$ by invoking the Barbalat's Lemma. Therefore, the proposed controller is globally stable. ■

3.2.5 Equivalence

In this subsection, we show that the two controllers are closely related.

Proposition 3.7. *If τ_2 is designed such that*

$$\bar{K}_d = T^{-T}K_d T^{-1} \tag{3.37}$$

The Lyapunov functions and the Lie derivatives associated with the two closed-loop system will be the same if given the same initial conditions and desired trajectory. However, the resulting closed-loop trajectories will still be different.

Proof. Since $s_2 = Ts_1$, the two Lyapunov functions can be shown to be the same

$$\begin{aligned}V(t) &= \frac{1}{2}s_2^T H s_2 + \frac{1}{2}\tilde{a}^T\Gamma\tilde{a} \\ &= \frac{1}{2}s_1^T T^T \cdot T^{-T}JT^{-1} \cdot Ts_1 + \frac{1}{2}\tilde{a}^T\Gamma\tilde{a} \\ &= \frac{1}{2}s_1^T J s_1 + \frac{1}{2}\tilde{a}^T\Gamma\tilde{a}\end{aligned} \tag{3.38}$$

The Lie derivatives are

$$\begin{aligned}\dot{V}(t) &= -s_2^T\bar{K}_d s_2 \\ &= -s_1^T T^T \cdot T^{-T}K_d T^{-1} \cdot Ts_1 \\ &= -s_1^T K_d s_1\end{aligned} \tag{3.39}$$

We can also see that the two controllers look very similar. Through easy manipulations, one can show that the two controllers are given by

$$\begin{aligned}&\text{Hamiltonian approach}: \begin{cases}\tau_2 = \Big[[\omega_r\times]\mathcal{L}(\omega) + \mathcal{L}(\dot{\omega}_r)\Big]\hat{a}_2 - K_d s_1 \\ \dot{\hat{a}}_2 = -\Gamma^{-1}\Big[[\omega_r\times]\mathcal{L}(\omega) + \mathcal{L}(\dot{\omega}_r)\Big]^T s_1\end{cases} \\ &\text{direct approach}: \begin{cases}\tau_1 = \Big[[\omega\times]\mathcal{L}(\omega) + \mathcal{L}(\dot{\omega}_r)\Big]\hat{a}_1 - K_d s_1 \\ \dot{\hat{a}}_1 = -\Gamma^{-1}\Big[[\omega\times]\mathcal{L}(\omega) + \mathcal{L}(\dot{\omega}_r)\Big]^T s_1\end{cases}\end{aligned} \tag{3.40}$$

The difference is highlighted in blue. ■

3.2.6 Simulation results

Regulation

This subsection briefly illustrates the performance of the proposed controllers with simple simulation. The selected scheme is regulating the attitude of a spacecraft with inertial matrix

$$a = \begin{bmatrix} 15 & 5 & 5 \\ 5 & 10 & 7 \\ 5 & 7 & 20 \end{bmatrix} \tag{3.41}$$

from its initial attitude $p_{init} = [0.1,\ 1,\ 0.5]^T$ to zero, i.e. $p_d = \dot{p}_d = \ddot{p}_d = [0,\ 0,\ 0]^T$. The controller parameters are chosen to be

$$\Lambda = 10 \cdot I_{3\times 3}, \quad K_d = 20 \cdot I_{3\times 3}, \quad \Gamma = 30 \cdot I_{6\times 6} \tag{3.42}$$

The closed loop system response is shown in Figure 3.1. The state is successfully regulated to zero with both control laws. However, there is a noticeable difference between the trajectories. As pointed out previously, the Lyapunov functions and the Lie derivatives are

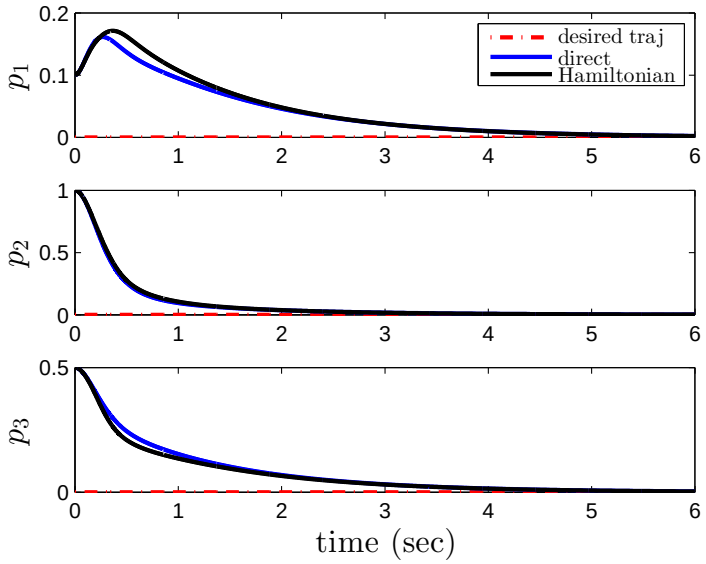


Figure 3.1: The closed-loop system response using the proposed adaptive attitude controllers. The desired trajectory is in red. The actual attitude profile using the direct approach is in blue. The attitude profile using the Hamiltonian approach is in **black**.

exactly the same under two controllers, which is depicted in Figure 3.2. The control inputs

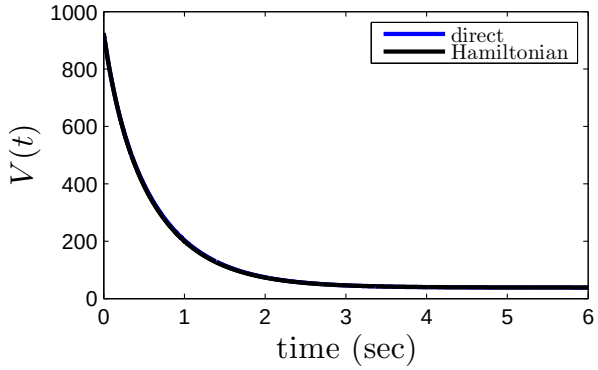


Figure 3.2: The Lyapunov function values corresponding to the closed-loop response. The two curves overlap expectedly.

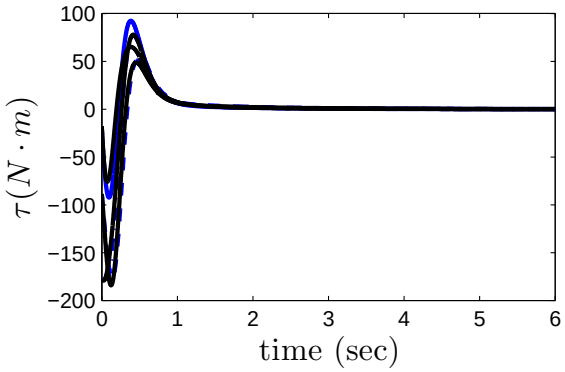


Figure 3.3: The feedback control inputs (blue: direct approach, **black**: Hamiltonian approach).

are shown in Figure 3.3. It is worth noting that the parameter estimation is not guaranteed to converge to the true values. Indeed, in the simulation, we start with $\hat{a}_{init} = a$ but end with

$$\hat{a}_{final} = \begin{bmatrix} 15.0300 \\ 9.0771 \\ 19.6904 \\ 5.7353 \\ 4.9323 \\ 5.0332 \end{bmatrix}, \quad \text{but } a = \hat{a}_{init} = \begin{bmatrix} 15 \\ 10 \\ 20 \\ 7 \\ 5 \\ 5 \end{bmatrix} \tag{3.43}$$

3.3 Quaternions

In the following section, we discuss the attitude controller when the attitude is parameterized by unit quaternions. The state-space equation of the plant is given by

$$\text{plant}: \quad \begin{cases} \dot{q} = \frac{1}{2}\Xi(q)\omega & \text{new kinematic equation} \\ J\dot{\omega} = -[\omega\times]J\omega + \tau & \text{dynamic equation} \end{cases} \tag{3.44}$$

Assume that in this case the desired attitude profile is given by $q_d(t)$ which is at least twice differentiable. By the properties of the quaternion kinematics, we can easily compute the desired body angular velocity and acceleration as

$$\begin{aligned} \omega_d &= 2\Xi(q_d)\dot{q}_d \\ \dot{\omega}_d &= 2\Xi(q_d)\ddot{q}_d \end{aligned} \tag{3.45}$$

Again, the control objective is to design a feedback controller

$$\tau = \tau(q, \omega, q_d, \dot{q}_d, \ddot{q}_d) \tag{3.46}$$

so that $q \to q_d$ as $t \to \infty$.

3.3.1 Optimal design of sliding manifold

Crassidis *et. al.* proposed an optimal sliding manifold [17]. The optimality is evaluated when we only consider the quaternion kinematics equation and treat ω as the input. The tracking error is defined using quaternion composition rule

$$\delta q = q \otimes q_d^{-1} = \begin{bmatrix} \Xi(q_d^{-1}), & q_d^{-1} \end{bmatrix} q = \begin{bmatrix} \delta\rho \\ \delta q_4 \end{bmatrix} = \begin{bmatrix} \Xi^T(q_d)q \\ q^T q_d \end{bmatrix} \tag{3.47}$$

This resolves the sign ambiguity and is physically meaningful. The following functional is minimized

$$J^*(q(t), t) = \min_{\omega} \int_t^{\infty} \frac{1}{2}\left\{ r^2 \delta\rho^T \delta\rho + (\omega - \omega_d)^T(\omega - \omega_d) \right\} d\tau \tag{3.48}$$

subject to the kinematic constraint

$$\dot{q} = \frac{1}{2}\Xi(q)\omega$$

and the endpoint constraint

$$\delta q(\infty) = [\ 0,\ 0,\ 0,\ 1]^T \tag{3.49}$$

where $r > 0$ is the weighting factor, q_d and ω_d satisfy Equation (3.45). Without loss of generality, we only consider $\delta q_4(t) \geq 0$.

There exists two main approaches to optimal control [37]: (1) calculus of variations (the

Lagrangian $L(q, \omega, t)$	$\frac{1}{2}r^2\delta\rho^T\delta\rho + \frac{1}{2}(\omega - \omega_d)^T(\omega - \omega_d)$
Lagrange multipliers	$\lambda \in \mathbb{R}^4$
Hamiltonian $H(q, \lambda, \omega, t)$	$L + \frac{1}{2}\lambda^T\Xi(q)\omega$
State equation	$\dot{q} = \frac{1}{2}\Xi(q)\omega$
Costate equation	$\dot{\lambda} = -\frac{\partial H}{\partial q}$
Stationary condition	$\frac{\partial H}{\partial \omega} = 0$
Boundary condition	$\delta q(\infty) = [\ 0,\ 0,\ 0,\ 1]^T$

Table 3.1: The necessary conditions for optimality from calculus of variations

maximum principle) and (2) dynamic programming (the principle of optimality).

Calculus of variations

In [17], the necessary conditions for optimality are derived from calculus of variations. For the functional minimization problem in Equation(3.48), the necessary conditions are summarized in Table 3.1. It can be shown by direct substitution that the following optimal angular velocity ω^*

$$\omega^* = \omega_d - r\Xi^T(q_d)q \tag{3.50}$$

with $\lambda^* = -2rq_d$, satisfies all the conditions except the boundary condition. To prove the satisfaction of the boundary condition, we use the kinematic equation for δq

$$\begin{aligned} \delta\dot{\rho} &= \frac{1}{2}\delta q_4(\omega - \omega_d) + \frac{1}{2}[\delta\rho\times](\omega + \omega_d) \\ \delta\dot{q}_4 &= -\frac{1}{2}(\omega - \omega_d)^T\Xi^T(q_d)q \end{aligned} \tag{3.51}$$

and a Lyapunov function candidate

$$V = \frac{1}{2}\delta\rho^T\delta\rho \tag{3.52}$$

The Lie derivative taken with respect to the kinematic relation is

$$\dot{V} = -\frac{1}{2}r\delta q_4\delta\rho^T\delta\rho \leq 0 \tag{3.53}$$

The Lyapunov function value will keep decreasing until $\delta\rho = 0_{3\times1}$ and $\delta q_4 = \pm 1$. From Equation (3.51) and the minimizer ω^*, δq_4 can converge only to 1 since

$$\delta\dot{q}_4 = \frac{1}{2}r(1 - \delta q_4^2) \geq 0, \quad = 0 \text{ only if } \delta q_4 = 1 \tag{3.54}$$

Therefore, all the necessary conditions are satisfied. The optimal value $J^*(q(t),t)$ can be derived to be

$$\begin{aligned} J^*(q(t),t) &= \int_t^\infty r^2(1-\delta q_4^2)d\tau \\ &= 2r\int_t^\infty \frac{r}{2}(1-\delta q_4^2)d\tau \\ &= 2r\int_t^\infty \delta\dot{q}_4 d\tau \\ &= 2r(1-\delta q_4(t)) \end{aligned} \tag{3.55}$$

Dynamic programming
We can also prove optimality by showing that ω^* satisfies the following Hamilton-Jacobi-Bellman partial differential equation

$$\frac{\partial J^*}{\partial t}(q,t) = -H(q, \frac{\partial J^*(q,t)}{\partial q}, \omega^*, t) \tag{3.56}$$

Proof: We expand $\frac{\partial J^*}{\partial t}(q,t)$ by the chain rule,

$$\begin{aligned} LHS &= \frac{\partial J^*(q,t)}{\partial q_d}\frac{dq_d}{dt} \\ &= -rq^T\Xi(q_d)\omega_d \end{aligned} \tag{3.57}$$

On the other hand, by substituting $\frac{\partial J^*}{\partial q} = -2rq_d$ into the Hamiltonian, the right-hand side becomes,

$$\begin{aligned} H(q,-2rq_d,\omega^*,t) &= r^2\delta\rho^T\delta\rho - rq_d^T\Xi(q)(\omega_d - r\Xi^T(q_d)q) \\ &= -rq^T\Xi(q_d)\omega_d \end{aligned} \tag{3.58}$$

Optimal sliding surface
For optimal tracking performance, it is natural to select the following switching function s_1.

Proposition 3.8. *The optimal switching function s_1 defined by*

$$s_1 = \omega - \omega_r, \quad \text{where } \omega_r = \omega_d - r\,sgn[\delta q_4]\Xi^T(q_d)q \tag{3.59}$$

is stable. Note that $sgn[\delta q_4(t)]$ is added for generality.

Proof. The stability of this sliding manifold has already been seen from the boundary condition, i.e. $q \to q_d$ as $t \to \infty$. In view of the sliding condition, we can further show the velocity tracking

$$\omega = \omega_d - r\underbrace{\Xi^T(q_d)q}_{\delta\rho\to 0} \to \omega_d, \quad t\to\infty \tag{3.60}$$

■

We will use this switching function in the direct approach. Alternatively, we design a different switching function for the Hamiltonian approach.

Proposition 3.9. *The switching function s_2 defined by*

$$s_2 = \dot{q} - \dot{q}_r, \quad \text{where } \dot{q}_r = \frac{1}{2}\Xi(q)\omega_r \tag{3.61}$$

is also stable.

Proof. The stability can be inferred from the stability of s_1. When the state variables are confined in the sliding manifold $s_2 = 0$, for any $q \neq q_d$, we must have

$$s_1 = 2\Xi^T(q)s_2 = 0 \tag{3.62}$$

Hence $q \to q_d$ as $t \to \infty$. ■

3.3.2 Direct approach

The reaching law derived from the direct approach is given in the following proposition.

Proposition 3.10. *The following controller*

$$\tau_1 = Y\hat{a}_1 - K_d s_1 \tag{3.63}$$

and the parameter adaptation law

$$\hat{a}_1 = -\Gamma^{-1}Y^T s_1 \tag{3.64}$$

yields a globally stable adaptive controller, where Γ and K_d are positive -definite matrices and $Y \in \mathbb{R}^{3\times 6}$ is given by

$$Y(q, \dot{q}, q_d, \dot{q}_d, \ddot{q}_d) = [\omega\times]\mathcal{L}(\omega) + \dot{\omega}_r \tag{3.65}$$

where

$$\dot{\omega}_r = \dot{\omega}_d - r\,sgn[\delta q_4](\Xi^T(\dot{q}_d)q + \Xi^T(q_d)\dot{q}) \tag{3.66}$$

Proof. The proof is the same as **Proposition 2.4**. The only modification we need to make is to change the switching function. ■

3.3.3 Hamiltonian approach

The derivation here is similar to the previous derivation of the Hamiltonian approach using Rodrigues parameters.

Lemma 3.3. *The state-space equation of the plant (using quaternion kinematics) can be cast into the form of a second-order differential equation in q*

$$\begin{aligned} & H(q)\ddot{q} + C(q,\dot{q})\dot{q} = F \\ \textit{where} \quad & H(q) = 4\Xi J \Xi^T \\ & C(q,\dot{q}) = -4\Xi J \Xi^T \dot{\Xi}\Xi^T - 4\Xi[J\omega\times]\Xi^T \\ & F = 2\Xi\tau \end{aligned} \tag{3.67}$$

Proof. The proof follows easily by noting that the left inverse of $\Xi(q)$ is $\Xi^T(q)$ for any unit quaternion q. Differentiating the quaternion kinematic equation $\dot{q} = \frac{1}{2}\Xi(q)\omega$

$$\ddot{q} = \frac{1}{2}\dot{\Xi}(q)\omega + \frac{1}{2}\Xi(q)\dot{\omega} \tag{3.68}$$

Pre-multiplying $2J\Xi^T(q)$ on both sides and substituting in the dynamic equation

$$2J\Xi^T\ddot{q} + (-J\Xi^T\dot{\Xi} - [J\omega\times])\omega = \tau \tag{3.69}$$

Pre-multiplying 2Ξ and replacing ω by $2\Xi^T\dot{q}$ yields

$$\underbrace{4\Xi J\Xi^T}_{H(q)}\ddot{q} + \underbrace{(-4\Xi J\Xi^T\dot{\Xi}\Xi^T - 4\Xi[J\omega\times]\Xi^T)}_{C(q,\dot{q})}\dot{q} = \underbrace{2\Xi\tau}_{F} \tag{3.70}$$

Here we intentionally kept the common factor 2 for showing the equivalence of the two approaches shortly. ■

Lemma 3.4. *The matrix $\dot{H}(q) - 2C(q,\dot{q})$ is also skew-symmetric. The new state equation is linear in terms of the inertial parameters J. One can write*

$$H\ddot{q}_r + C\dot{q}_r = \bar{Y}a \tag{3.71}$$

The expression of $\bar{Y}$ is given by

$$\bar{Y} = \Xi \cdot \left[\mathcal{L}(\dot{\omega}_r) + [\omega_r\times]\mathcal{L}(\omega)\right] \tag{3.72}$$

Proof.

$$\begin{aligned} H\ddot{q}_r + C\dot{q}_r &= 4\Xi J\Xi^T\ddot{q}_r + (-4\Xi J\Xi^T\dot{\Xi}\Xi^T - 4\Xi[J\omega\times]\Xi^T)\dot{q}_r \\ &= 2\Xi J\Xi^T(\dot{\Xi}\omega_r + \Xi\dot{\omega}_r) - 2\Xi J\Xi^T\dot{\Xi}\omega_r - 2\Xi[J\omega\times]\omega_r \\ &= 2\Xi \cdot \left[\mathcal{L}(\dot{\omega}_r) + [\omega_r\times]\mathcal{L}(\omega)\right]a \end{aligned} \tag{3.73}$$

■

Proposition 3.11. *The following control law*

$$\begin{aligned} F &= \bar{Y}\hat{a}_2 - \bar{K}_d s_2 \\ \tau_2 &= \frac{1}{2}\Xi^T F \end{aligned} \tag{3.74}$$

and the adaptation law

$$\dot{\hat{a}}_2 = -\Gamma^{-1}\bar{Y}^T s_2 \tag{3.75}$$

yield a global stable adaptive controller, where $\bar{Y}$ is given by Equation (3.72)*.*

Proof. The proof is the same as **Proposition 2.6**. ■

3.3.4 Equivalence

The two approaches can be shown to be closely related.

Proposition 3.12. *If τ_2 is designed such that*

$$\bar{K}_d = 4\Xi K_d \Xi^T \tag{3.76}$$

The Lyapunov functions and the Lie derivatives associated with the two closed-loop system will be the same if given the same initial conditions and desired trajectory.

Proof. Since $s_2 = \frac{1}{2}\Xi s_1$, the two Lyapunov functions can be shown to be the same

$$\begin{aligned} V(t) &= \frac{1}{2}s_2^T H s_2 + \frac{1}{2}\tilde{a}^T\Gamma\tilde{a} \\ &= \frac{1}{2}(\frac{1}{2}s_1^T\Xi^T)\cdot 4\Xi J\Xi^T\cdot(\frac{1}{2}\Xi s_1) + \frac{1}{2}\tilde{a}^T\Gamma\tilde{a} \\ &= \frac{1}{2}s_1^T J s_1 + \frac{1}{2}\tilde{a}^T\Gamma\tilde{a} \end{aligned} \tag{3.77}$$

The derivatives are

$$\begin{aligned} \dot{V}(t) &= -s_2^T\bar{K}_d s_2 \\ &= -(\frac{1}{2}s_1^T\Xi^T)\cdot 4\Xi K_d\Xi^T\cdot(\frac{1}{2}\Xi s_1) \\ &= -s_1^T K_d s_1 \end{aligned} \tag{3.78}$$

The expressions of the two controllers are exactly the same as Equation (5.1), except that the definition of s_1 is now Equation (3.59). ■

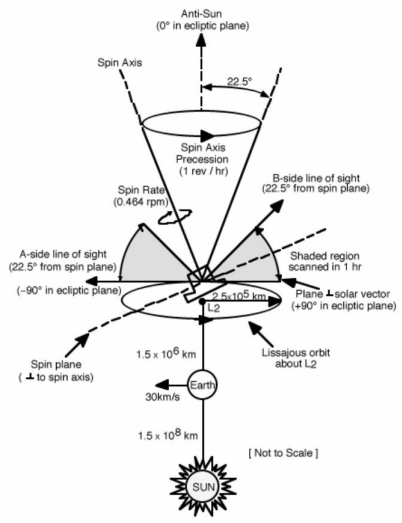


Figure 3.4: Map spacecraft specifications. Source: [17].

3.3.5 Simulation results

Trajectory following

We test the proposed controllers in a more realistic scheme, for controlling the attitude of the Microwave Anisotropy Probe (MAP) spacecraft [17]. The objective of the MAP mission is to create a full-sky map of the cosmic microwave background and measure anisotropy with 0.3^o angular resolution, in order to answer fundamental cosmological questions such as inflationary versus non-inflationary "big bang" models, accurate determination of the Hubble constant, and the existence and nature of dark matter.

The spacecraft orbit and attitude specifications are shown in Figure 3.4. The spacecraft spins about its body z-axis at 0.464 rpm, and the z-axis cones about the Sun-line at 1 rev/hour. A $22.5^o \pm 0.25^o$ angle between the z-axis and the Sun direction must be maintained to provide a constant power input, and to provide constant temperatures for alignment stability and science quality. The desired trajectory is animated in Figure 3.5, in which the precession motion can be clearly seen.

The spacecraft's desired attitude is defined by 3-1-3 Euler angles. The three Euler angles are ϕ_d, θ_d, ψ_d

$$\begin{aligned} \dot{\phi}_d &= 1\ rev/hr = 0.001745\ rad/sec \\ \dot{\psi}_d &= 0.464\ rpm = 0.04859\ rad/sec \\ \theta_d &= 22.5^o = 0.3927\ rad \end{aligned} \tag{3.79}$$

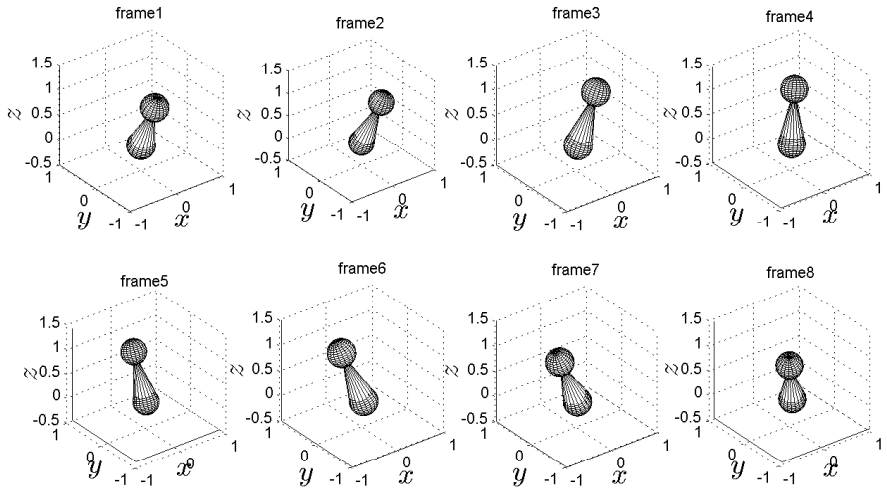


Figure 3.5: Animated desired trajectory of the MAP spacecraft. The spacecraft is rotating about its body z-axis in addition to the precession motion.

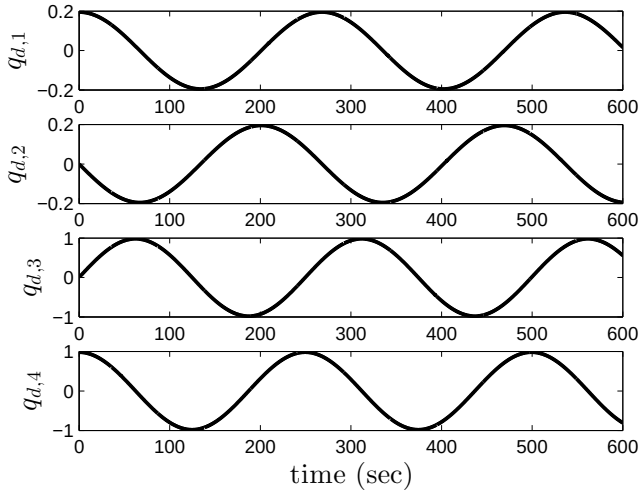


Figure 3.6: Desired quaternion profile of the MAP spacecraft.

ϕ_d and ψ_d are computed by integrating the Euler rates from zero initial conditions. Then we can convert from Euler angles to quaternions as

$$q_d(t) = \begin{bmatrix} \sin(\frac{\theta_d}{2}) \cdot \cos(\frac{\phi_d - \psi_d}{2}) \\ \sin(\frac{\theta_d}{2}) \cdot \sin(\frac{\phi_d - \psi_d}{2}) \\ \cos(\frac{\theta_d}{2}) \cdot \sin(\frac{\phi_d + \psi_d}{2}) \\ \cos(\frac{\theta_d}{2}) \cdot \cos(\frac{\phi_d + \psi_d}{2}) \end{bmatrix} \tag{3.80}$$

The desired quaternion profile is plotted in Figure 3.6. $\dot{q}_d$ and $\ddot{q}_d$ are obtained by numerically differentiating q_d. We introduce an initial attitude error

$$q(t_0) = [0,\ 0,\ \sin(60^o/2),\ \cos(60^o/2)]^T \otimes q_d(t_0) \tag{3.81}$$

i.e. the spacecraft initially has a 60^o error angle from the desired attitude along the body z-axis. The controller parameters are chosen to be

$$r = 1, \quad K_d = 2 \cdot I_{3\times 3}, \quad \Gamma = 3 \cdot I_{6\times 6} \tag{3.82}$$

No prior information of J is assumed. That is

$$\hat{a}_{init} = \mathbf{0}_{6\times 1} \tag{3.83}$$

A plot of the roll-pitch-yaw attitude errors are shown in Figure 3.7. It is observed that the tracking errors converge to zero asymptotically under both control laws from the initial 60^o error. The two curves are overlapping in the plot, indicating that the two closed-loop trajectories are really close to each other. The Lyapunov function values along the closed-loop trajectories under the two approaches is shown in Figure 3.8. It is guaranteed to monotonically decrease until $\dot{V} = 0$. The low convergence rate is due to the fact that no information of the inertial parameters is given. The feedback control inputs is shown in Figure 3.9. In all of the plots, the difference in the numerical values under the two approaches is not noticeable.

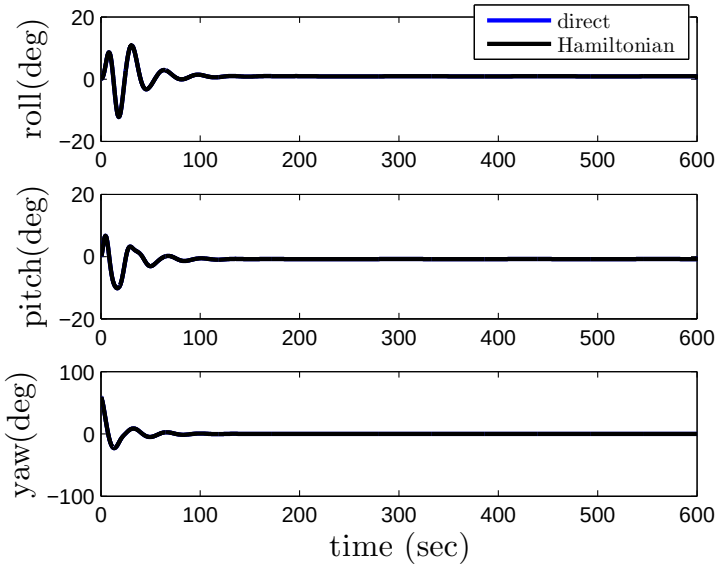


Figure 3.7: The tracking errors represented by the Euler angles using the proposed adaptive attitude controller (quaternions). blue: the direct approach, **black**: Hamiltonian approach.

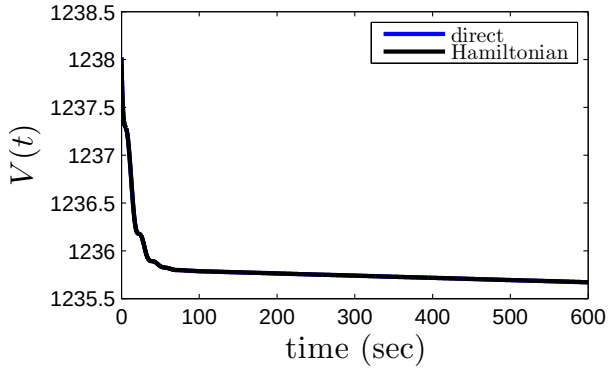


Figure 3.8: The Lyapunov function values corresponding to the closed-loop response. The two curves overlap, as expected.

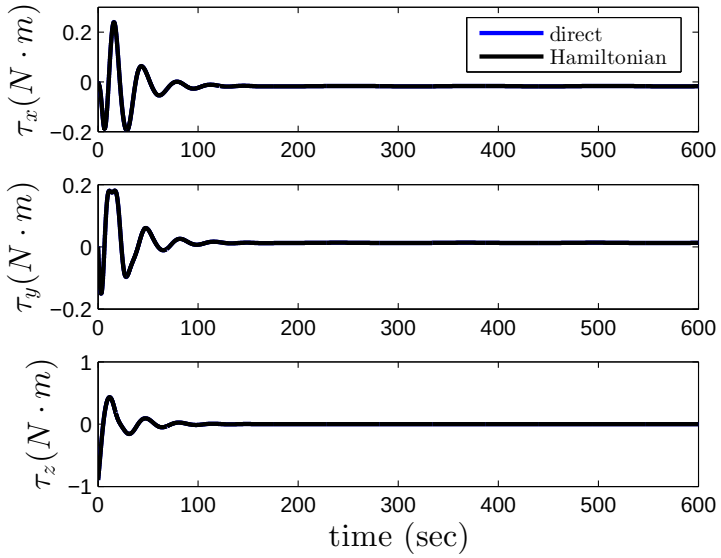


Figure 3.9: The feedback control inputs (blue: direct approach, **black**: Hamiltonian approach).

3.4 Parameter convergence

To enforce the asymptotic parameter estimation, namely $\tilde{a} \to 0$ as $t \to 0$, the following persistent excitation condition must be satisfied

$$\int_t^{t+T} F^T(\delta q, \omega, \omega_d, \dot{\omega}_d) F(\delta q, \omega, \omega_d, \dot{\omega}_d) d\tau \geq \epsilon I_{6\times 6}, \quad \forall t \geq t_o \tag{3.84}$$

where T, t_o, ϵ are some positive scalars. However, the parameter convergence will not have an impact on the tracking performance.

3.5 Disturbance rejection

The framework used so far makes it very easy to modify the controller to account for possible unexpected disturbances appearing in the dynamic equation

$$J\dot{\omega} = -[\omega\times]J\omega + \tau + d \tag{3.85}$$

Physically, the disturbance can be a combination of air drag, solar pressure, gravity gradient, magnetic force, spherical harmonics, etc.. The disturbance can safely be assumed to be bounded with known bounds

$$|d_i(t)| \leq D_i, \quad i = 1, 2, 3 \tag{3.86}$$

D_i can also be time-varying.

The *robust* controller is given in the following proposition. Since the notations in the direct or the Hamiltonian approaches with Rodrigues parameters or quaternions are very similar, we will present the theory using general notations.

Proposition 3.13. *The following control law*

$$\tau = Y\hat{a} - K_d s - (D + \eta) \cdot sgn(s) \tag{3.87}$$

and the adaptation law

$$\dot{\hat{a}} = -\Gamma^{-1}Y^T s \tag{3.88}$$

yield a globally stable adaptive controller with disturbance rejection. $(D + \eta) \cdot sgn(s)$ *is calculated component-wise*

$$(D + \eta) \cdot sgn(s) = \begin{bmatrix} (D_1 + \eta_1)sgn(s_1) \\ (D_2 + \eta_2)sgn(s_2) \\ (D_3 + \eta_3)sgn(s_3) \end{bmatrix} \tag{3.89}$$

Proof. With the same Lyapunov function used before, it is straight-forward to show that

$$\dot{V} \leq -\sum_{i=1}^{3} \eta_i |s_i| - s^T K_d s \leq 0 \tag{3.90}$$

The system trajectories are thus guaranteed to reach the sliding manifold $s = 0$ regardless of unknown disturbances, and therefore the control objective is achieved. ■

Chattering

Chattering is a major problem with sliding mode controller. A standard solution is replacing the $sgn(s_i)$ function by the saturation function $sat(s_i/\phi_i)$, where ϕ_i represents the thickness of the corresponding *boundary layer*. After doing this, the switching function s is guaranteed to converge to the boundary layer, corresponding to small tracking errors.

Simulation results

Assume that the MAP spacecraft is now subject to the following external disturbance

$$d = \begin{bmatrix} 0.005 \cdot \sin(0.05t) \\ 0.003 \\ 0.005 \cdot \cos(0.05t) \end{bmatrix} \tag{3.91}$$

The parameters of the robust controller are set to

$$D = 0.005 \cdot \mathbf{1}_{3\times 1}, \quad \eta = 0.001 \cdot \mathbf{1}_{3\times 1} \tag{3.92}$$

To highlight the ability to reject disturbance, we assume no initial attitude error and perfect knowledge of J. The roll-pitch-yaw error angles are plotted in Figure 3.10. It is seen in the plot that the tracking error is kept small with the presence of external disturbances, under the robust controller.

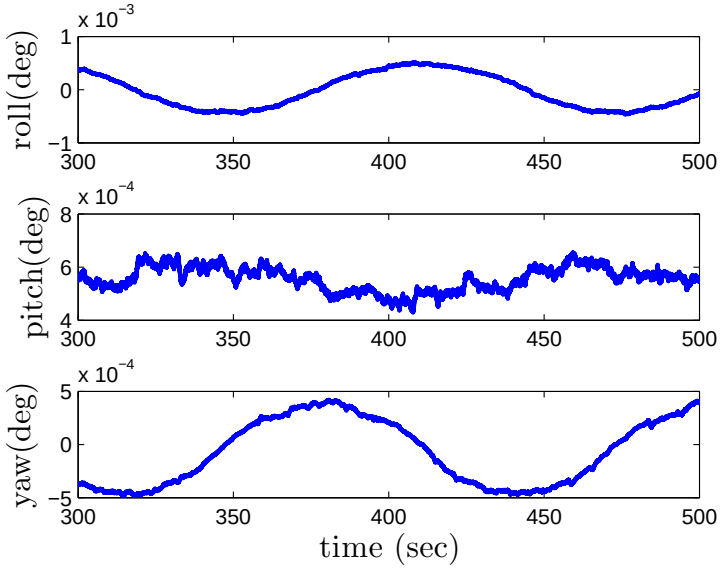


Figure 3.10: Tracking performance of the robust controller when the MAP spacecraft is subject to external disturbances.

3.6 Summary

In this chapter, an adaptive sliding mode attitude controller is designed for asymptotic quaternion tracking, which assumes no inertial information and can reject unexpected external disturbances. The stability was shown through a Lyapunov analysis. Both the nominal performance and the robust performance are demonstrated in numerical simulations.

Chapter 4

Sensor Measurement Models

This chapter provides a basic review of the models of rate measurements (rate gyroscopes) and angle measurements (accelerometers, magnetometers, star cameras). Two algorithms for determining attitude from angle measurements are reviewed.

4.1 Gyroscope Model

A gyroscope is a commonly used device/sensor that measures angular velocity. There are three types of gyroscope, mechanical, optical and MEMS. They are *strap-down*, meaning that they are mounted rigidly on the object, and therefore output quantities expressed in the body frame rather than the reference frame. The measurement is usually mathematically modeled as

$$\begin{aligned} \omega &= \omega_{true} + \beta + \eta_v \\ \dot{\beta} &= \eta_u \end{aligned} \tag{4.1}$$

where ω_{true} and β denote the true angular rate and the bias drift. η_v is thermo-mechanical zero-mean Gaussian white noise, appearing additive in the measurement. It is usually referred to as the angle random walk (ARW), with units $rad/sec^{1/2}$ or $^o/hr^{1/2}$. The bias of a gyroscope wanders over time due to flicker noise. It is modeled as a random signal, driven by a zero-mean Gaussian white noise η_u. It is referred to as the rate random walk (RRW), with units $rad/sec^{3/2}$ or $^o/hr^{3/2}$. The two noises are not correlated.

There are other error sources not included in the model above, (i) a constant bias, (ii) temperature effects (temperature dependent residual bias), (iii) calibration error (deterministic errors in scale factors, axes alignment, linearities), (iv) quantization error introduced during analog-digital conversion. Those are either ignored or taken care of by proper calibrations.

One simple way to obtain orientation information is to simply integrate rate measurements

from a gyroscope. However, due to the presence of noises, the uncertainty will grow unbounded as time increases. Therefore, we need some other sensors to provide angle measurements directly.

4.2 Accelerometer and Magnetometer Models

Accelerometers and magnetometers are also common in inertial measurement units (IMU). They are also strap-down devices, directly measuring quantities in the body frame. An accelerometer measures both the gravitational acceleration and the motion acceleration. Its model can be written as

$$a = R \cdot g + a_{motion} + \eta_a \tag{4.2}$$

where R is the rotation matrix of the body frame with respect to the Earth frame, a_{motion} is the motion acceleration expressed in the body frame. η_a is a three-dimensional additive zero-mean Gaussian white noise.

If the device is at rest, i.e. the measurement only picks off gravity, we can calculate the relative orientation between the body frame and the Earth frame, with the orientation along the Earth z-axis still being ambiguous.

Similarly, a magnetometer measures the environmental magnetic field intensity, expressed in the body frame

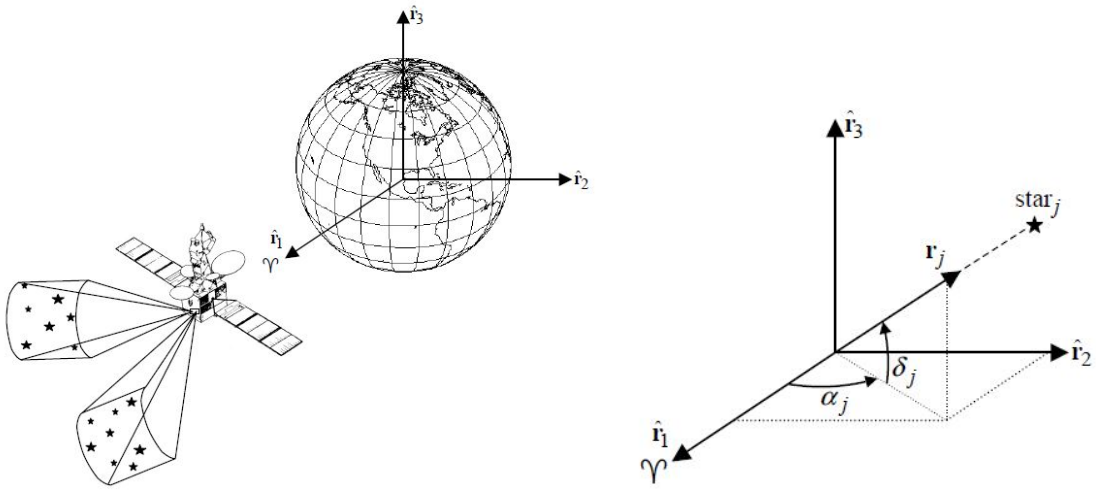
$$m = R \cdot m_{env} + \eta_m \tag{4.3}$$

where m_{env} denotes the magnetic field intensity in the Earth frame, η_m is also a three-dimensional additive zero-mean Gaussian white noise. By using these two vector measurements, one can compute the relative orientation without ambiguities.

4.3 Star Camera Model

In space application, one common angle sensor is the star camera (starcam). Photographs of the stars can be made from one or more spacecraft fixed cameras. A schematic of starcam measurements is shown in Figure 4.1(a). The brightest 250,000 stars spherical coordinate angles (i.e. the right ascension angle α and the declination angle δ) are available in a computer accessible catalog. Namely, the vector measurement expressed in the reference frame is known (Figure 4.1(b))

$$r_j = \begin{bmatrix} r_{xj} \\ r_{yj} \\ r_{zj} \end{bmatrix} = \begin{bmatrix} \cos\delta_j \cos\alpha_j \\ \cos\delta_j \sin\alpha_j \\ \sin\alpha_j \end{bmatrix} \tag{4.4}$$



(a) (b)

Figure 4.1: (a) Schematics of starcam measurements, (b) Position of a cataloged star expressed in the reference frame. Source: [22].

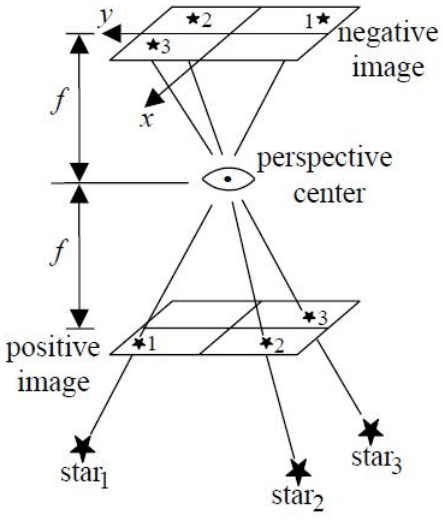


Figure 4.2: Collinearity of Perspective Center, Image and Object. Source: [22].

4.3.1 Focal plane model

Referring to Figure 4.2, if the relative rotation between the reference frame and the camera frame is denoted as A, the photograph image plane x, y coordinates of the j-th star can be

determined by the stellar collinearity equations

$$\begin{cases} x_j = -f\left(\frac{A_{11}r_{xj}+A_{12}r_{yj}+A_{13}r_{zj}}{A_{31}r_{xj}+A_{32}r_{yj}+A_{33}r_{zj}}\right) \\ y_j = -f\left(\frac{A_{21}r_{xj}+A_{22}r_{yj}+A_{23}r_{zj}}{A_{31}r_{xj}+A_{32}r_{yj}+A_{33}r_{zj}}\right) \end{cases} \tag{4.5}$$

where f is the camera focal length, which is known from a priori calibration. The sensor noise is usually modeled as an additive zero-mean Gaussian noise

$$\tilde{p}_j = \begin{bmatrix} -x_j \\ -y_j \end{bmatrix} + \epsilon_j, \quad \text{where } \mathbb{E}[\epsilon_j] = 0,\ \mathbb{E}[\epsilon_j \epsilon_j^T] = \sigma_j^2 I_{2\times 2} \tag{4.6}$$

4.3.2 QUEST model

The vector measurement b_j expressed in the body frame is (choosing the z-axis of the image coordinate system to be directed outward along the boresight)

$$b_j = \frac{1}{\sqrt{x_j^2 + y_j^2 + f^2}} \begin{bmatrix} -x_j \\ -y_j \\ f \end{bmatrix} \tag{4.7}$$

It is straightforward to see that

$$b_j = A r_j \tag{4.8}$$

and this holds true for every star $j = 1, \ldots, N$, if N stars are captured and cataloged.

Shuster [38] has shown that nearly all the probability of the errors is concentrated on a very small area about the direction of Ar_j

$$\tilde{b}_j = A r_j + v_j, \quad v_j^T A r_j = 0 \tag{4.9}$$

The measurement error v_j is modeled as a zero-mean Gaussian noise, with the following covariance

$$\mathbb{E}[v_j v_j^T] = \sigma_j^2 \left[I_{3\times 3} - (Ar_j)(Ar_j)^T \right] \tag{4.10}$$

Remark 4.1. *The noise polluted measurement vector $\tilde{b}_j$ is used in the covariance matrix because the true value Ar_j is not available in practice.*

4.4 Attitude Determination Algorithm

As mentioned early, with vector measurements along (i.e. without rate information), one can determine the attitude, parameterized by quaternions or rotation matrices. Therefore, we also referred to these types of measurements as *angle* measurements earlier.

In this section, two attitude determination algorithms are discussed for determining 3 DOF attitude from at least two vector observations. *Attitude determination*, which is intentionally phrased differently from *attitude estimation*, refers to the technique for obtaining a proper rotation (direction cosine) matrix so that the measured vector observations in the body frame are equal to the observations in the reference frame mapped by such a rotation matrix.

The vector observations could be unit-vector measurements (e.g. b_j from a star cameras) or non-unit-vector measurements (e.g. a from an accelerometers or m from a magnetometers). We are seeking to solve the following equation for A

$$b_i = Ar_i, \quad i = 1, \dots, N \tag{4.11}$$

where r_i's are a set of measured unit vectors in the reference frame (or normalized vectors), b_i's are the ones in the body frame and N is the number of the available vector sensors. Both observation vectors, measured in reference and sensor frames, are inevitably contaminated by measurement noises. Hence, an exact solution to Equation (4.11) for the rotation matrix A does not necessarily exist.

4.4.1 TRIAD algorithm

The TRIAD algorithm [20], which is the most popular attitude determination method due to its simplicity, provides a deterministic, non-optimal solution based on two nonparallel vector observation pairs $\{b_1, r_1\}$ and $\{b_2, r_2\}$. These vectors are assumed to have unity magnitude or have been normalized. The exact solution satisfying the following two equations simultaneously usually doesn't exist considering the presence of measurement noise in vector sensors

$$\begin{cases} b_1 = Ar_1 \\ b_2 = Ar_2 \end{cases} \tag{4.12}$$

An approximate solution, however, can be found by constructing two triads based on measured vectors in the reference and the sensor frame respectively as follows

$$\begin{aligned} v_1 &= r_1 \\ v_2 &= \frac{r_1 \times r_2}{||r_1 \times r_2||_2} \\ v_3 &= v_1 \times v_2 \end{aligned} \tag{4.13}$$

Namely, in order to construct the triad $\{v_1, v_2, v_3\}$ from the measurements $\{r_1, r_2\}$, first set v_1 equal to r_1, then calculate v_2 from the normalized cross product of v_1 and r_2 and finally calculate v_3 from the cross product of v_1 and v_2. Then, $\{v_1, v_2, v_3\}$ are three (orthonormal) axes of the triad. The triad $\{w_1, w_2, w_3\}$ can be constructed similarly from measurements $\{b_1, b_2\}$. This procedure is depicted is Figure 4.3.

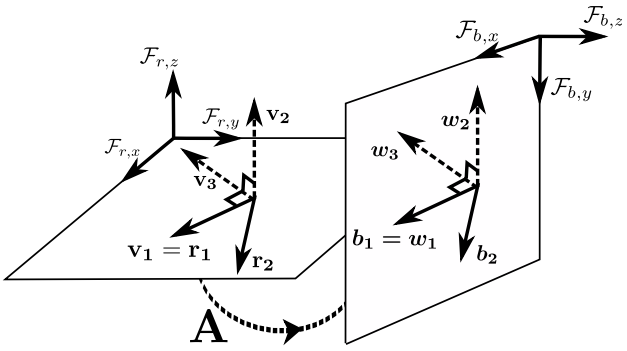


Figure 4.3: The diagram of the triad method. The rotation matrix **A** represents the transformation between the frames $\mathcal{F}_r$ and $\mathcal{F}_b$.

Lemma 4.1. *There exists a unique rotation matrix A satisfying*

$$w_i = Av_i, \quad i = 1, 2, 3 \tag{4.14}$$

and the solution is given by

$$A = \sum_{i=1}^{3} w_i v_i^T \tag{4.15}$$

Proof. To prove that the given expression is a solution, we simply plug it back to the right-hand side of the equality. One can easily see that

$$RHS = (\sum_{i=1}^{3} w_i v_i^T) v_j = w_j \tag{4.16}$$

since v_i's are orthonormal

$$v_i^T v_j = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases} \tag{4.17}$$

To show the uniqueness, we assume that there exists another solution $\bar{A} \neq A$. We then have

$$\begin{aligned} w_i &= A\bar{A}^T w_i, \quad i = 1, 2, 3 \\ &\Rightarrow A\bar{A}^T = I_{3\times 3} \\ &\Rightarrow A = \bar{A} \end{aligned} \tag{4.18}$$

which leads to a contradiction. ■

The rotation matrix A given in the lemma is an approximate solution to Equation (4.11) (it doesn't solve the equation with i = 2). Clearly, the TRIAD algorithm discards some

information in $\{b_2, r_2\}$. However, the approximate solution is exact if the following condition is true

$$b_1 \cdot b_2 = r_1 \cdot r_2 \tag{4.19}$$

The drawback of the TRIAD algorithm is that it can accommodate at most two observation pairs. In cases where many vector observations are available, the TRIAD algorithm can be cumbersomely repeated for every two pairs and the final determined attitude can fuse all estimates by some means.

4.4.2 q-method

An optimal attitude solution is

$$\begin{aligned} A = \arg\min_A \underbrace{\frac{1}{2}\sum_{i=1}^{N} \sigma_i^{-2} ||b_i - Ar_i||_2^2}_{\triangleq J(A)} \\ \text{subject to} \quad AA^T = I_{3\times 3} \end{aligned} \tag{4.20}$$

where σ_i is the weight on each observation pair, accounting for the confidence in each sensor. This problem was first posed by Grace Wahba in 1965 [18] and the solution is the maximum-likelihood solution, or the least square solution. The equality constraint is directly from the definition of the rotation matrix. Its presence makes this problem harder to solve than an ordinary least-square problem. However, the closed-form solution does exist, and is given by the following lemma.

Lemma 4.2. *The optimal attitude solution $\hat{q}$ (parameterized in quaternions) can be found by performing eigenvalue decomposition on K*

$$K\hat{q} = \lambda_{max}\hat{q}, \quad \textit{where } K = \sum_{i=1}^{N} \sigma_i^{-2} \Omega(b_i)\Gamma(r_i) \tag{4.21}$$

That is, $\hat{q}$ is the normalized eigenvector of K associated with its largest eigenvalue.

Proof. We first rewrite the cost function

$$J(A) = -\sum \sigma_i^{-2} b_i^T A r_i + \text{const.} \tag{4.22}$$

The optimization problem is equivalent to maximizing

$$J'(A) = \sum \sigma_i^{-2} b_i^T A r_i \tag{4.23}$$

To simplify the calculation further, we change the parameterization to the quaternion

$$J'(q) = \sum \sigma_i^{-2} b_i^T \underbrace{\Xi(q)\Psi(q)}_{A} r_i \tag{4.24}$$

with a equality constraint $q^T q = 1$. Two identities are important here

$$\begin{aligned} \Xi(q)b &= \Omega(b)q \\ \Psi(q)r &= \Gamma(r)q \end{aligned} \tag{4.25}$$

Now the problem becomes

$$\hat{q} = \arg\max_q q^T K q \quad \text{subject to} \quad q^T q = 1 \tag{4.26}$$

One should be able to recognize that this problem is equivalent to the eigenvalue decomposition problem. ■

In contrast to the TRIAD algorithm, the q-method can accommodate arbitrary number of observation pairs and optimally determine the attitude. Despite that, [4] has shown that there is little advantage to use the q-method instead of the TRIAD algorithm when there are only two observation pairs, by analyzing the difference between two corresponding error covariance matrices. The TRIAD algorithm is less demanding in terms of the computation than the q-method because the latter involves an eigenvalue decomposition in the solution.

Chapter 5

Marginalized Particle Filter

A marginalized particle filter (MPF) is designed for attitude estimation problem. Unit quaternions are used to parameterize rotations. The linear structure in the gyroscope bias dynamics enables us to completely decouple its evolution from quaternion particles. We further show that the linear part of the proposed MPF reaches a steady state, similar to what Kalman filter does for controllable and observable linear stochastic systems. Although the MPF is similar to the particle filter in structure, it has two advantages: (i) the theoretical superiority of marginalizing linear substructure, and (ii) the reduction in total computational time. Numerical simulations are performed to demonstrate the performance of the proposed filter.

5.1 Introduction

As mentioned previously, due to recent development in computational power, the use of particle filters (PF) gained much traction and became practical for a broad area of applications. Cheng *et. al.* applied a bootstrap particle filter for sequential spacecraft attitude estimation [26]. Because of the high dimensionality of the state vector, a prohibitively large number of particles are needed to span the state space to support the state distribution. In contrast to this approach, Oshman *et. al.* reduced the computational burden by sampling only the attitude of the spacecraft and using a genetic algorithm to estimate the gyro bias [27].

If there exists a linear sub-structure within the nonlinear dynamics, it is possible to marginalize out the linear state variables and estimate them instead with the Kalman filter (KF) while the nonlinear state variables are estimated using the PF. This powerful combination of PF and KF, called the marginalized particle filter (MPF) or the Rao-Blackwellized particle filter, can effectively increase the estimation accuracy and possibly reduce the computation load [28]. The scheme has been directly applied for attitude estimation by Liu *et. al.* [29]. This dissertation's contribution is to further exploit the underlying linear-substructure, and show that the linear state evolution is completely independent of the nonlinear part.

The remainder of the chapter is organized as follows. In Sec. 5.2 through 5.5, a review of Bayesian inference, the sequential Monte Carlo method and the marginalized particle filter is provided. In Sec. 5.6, the MPF is applied to tackle the attitude estimation problem. The algorithm is discussed in details. In Sec. 5.7, an algorithmic comparison of PF and the steady-state MPF is presented. In Sec. 5.8, a numerical study is performed to show superior performance of MPF over PF and EKF.

5.2 Stochastic Model and Bayesian Inference

Consider a general discrete-time nonlinear stochastic model

$$x_{n+1} = f(x_n, u_n, w_n), \quad x_1 \sim p(x_1), w_n \sim p(w) \tag{5.1}$$

where x_n is a random vector representing the uncertain state variable at step n, w_n is a random vector representing the process noise, u_n is the deterministic control input, "$\sim$" means distributed according to, $p(x_1)$ is the probability density function of the initial state. If the probability density of w_n is known, the model above will further give $p(x_{n+1}|x_n)$, which is the probability density associated with the time evolution from x_n to x_{n+1}. Suppose we can make statistically independent observations at each time according to

$$y_n = h(x_n, u_n, v_n), \quad v_n \sim p(v) \tag{5.2}$$

That is, we are given $p(y_n|x_n)$, the likelihood of getting a measurement y_n conditioned on the state x_n. Note that the dependence on u_n has been dropped for notation simplicity.

The Bayesian filtering problem is to compute the posterior distribution of $x_{1:n}$ given a collection of observations $y_{1:n}$, denoted as $p(x_{1:n}|y_{1:n})$

$$p(x_{1:n}|y_{1:n}) = \frac{p(x_{1:n}, y_{1:n})}{p(y_{1:n})} \tag{5.3}$$

where the joint probability density function $p(x_{1:n}, y_{1:n})$ and the normalizing factor $p(y_{1:n})$ can be computed using the process model, the measurement model and the initial state distribution as follows

$$\begin{aligned} p(x_{1:n}, y_{1:n}) &= p(x_{1:n})p(y_{1:n}|x_{1:n}) \\ p(x_{1:n}) &= p(x_1)\prod_{k=2}^{n} p(x_k|x_{k-1}) \\ p(y_{1:n}|x_{1:n}) &= \prod_{k=1}^{n} p(y_k|x_k) \\ p(y_{1:n}) &= \int p(x_{1:n}, y_{1:n}) dx_{1:n} \end{aligned} \tag{5.4}$$

5.3 Recursive Bayesian Inference

Through straightforward algebraic manipulation and probability theory, one can show that the posterior distribution $p(x_{1:n}|y_{1:n})$ can be computed in a recursive way. The unnormalized posterior distribution $p(x_{1:n}, y_{1:n})$ can be expanded using the chain rule and simplified by the Markov assumption

$$p(x_{1:n}, y_{1:n}) = p(y_n|x_n) \cdot p(x_n|x_{n-1}) \cdot p(x_{1:n-1}, y_{1:n-1}) \tag{5.5}$$

Dividing both sides by $p(y_{1:n}) = p(y_n|y_{1:n-1}) \cdot p(y_{1:n-1})$, we get

$$p(x_{1:n}|y_{1:n}) = p(x_{1:n-1}|y_{1:n-1}) \cdot \frac{p(x_n|x_{n-1}) \cdot p(y_n|x_n)}{p(y_n|y_{1:n-1})} \tag{5.6}$$

Marginalizing out $x_{1:n-1}$ in the equation above, we have the following *update* equation (5.7)

$$p(x_n|y_{1:n}) = \frac{p(y_n|x_n) \cdot p(x_n|y_{1:n-1})}{p(y_n|y_{1:n-1})} \tag{5.7}$$

The prior distribution term $p(x_n|y_{1:n-1})$ comes from the fact that

$$\begin{aligned} &\int\int p(x_{1:n-1}|y_{1:n-1}) \cdot p(x_n|x_{n-1}) dx_{1:n-2} dx_{n-1} \\ =&\int p(x_{n-1}|y_{1:n-1}) \cdot p(x_n|x_{n-1}) d_{x_{n-1}} \\ =&p(x_n|y_{1:n-1}) \end{aligned} \tag{5.8}$$

Equation (5.8) above is known as the *prediction* equation.

5.4 Sequential Monte Carlo method

5.4.1 Monte Carlo Estimator

Consider a random variable x having a generic probability density function $\pi(x)$. If we take N independent samples of the random variable, $X^i \sim \pi(x)$ for $i = 1, \ldots, N$, then the Monte Carlo method approximates $\pi(x)$ by $\hat{\pi}(x)$, called *the Monte Carlo estimator (perfect sampling)*

$$\hat{\pi}(x) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X^i}(x), \quad X^i \sim \pi(x) \tag{5.9}$$

where $\delta_X(\bullet)$ denotes the Dirac delta function at X. The term "perfect sampling" comes from the fact that $X^i \sim \pi(x)$.

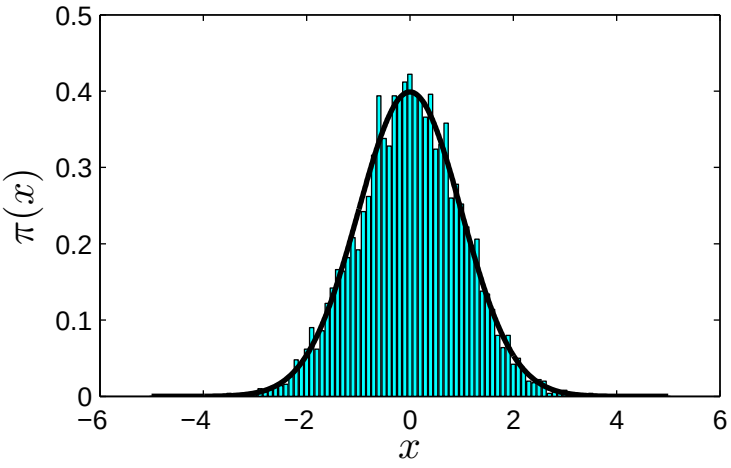


Figure 5.1: Comparison of the target distribution $\mathcal{N}(0,1)$ (in black) and its Monte Carlo estimator (in cyan bars). The estimator is essentially a (normalized) histogram of 5000 samples drawn from the target distribution.

Figure 5.1 shows an example demonstrating the approximation accuracy between a target distribution $\mathcal{N}(0,1)$ and its Monte Carlo estimator constructed using 5000 samples. We can also approximate the expectation of any test function $\phi(x)$ given by

$$\begin{aligned} \mathbb{E}[\phi(x)] &= \int \phi(x)\pi(x)dx \\ \hat{\mathbb{E}}[\phi(x)] &= \int \phi(x)\hat{\pi}(x)dx = \frac{1}{N}\sum_{i=1}^{N}\phi(X^i), \quad X^i \sim \pi(x) \end{aligned} \tag{5.10}$$

The last equality is obtained by simply plugging in the definition of $\hat{\pi}(x)$.

5.4.2 Importance Sampling

One problem associated with the basic Monte Carlo approach is that one is not able to directly draw samples from the target distribution. For example, in order to do Bayesian inference in the previous stochastic model, one has to sample from $p(x_{1:n}, y_{1:n})$, which is not possible.

Importance sampling, as opposed to perfect sampling, addresses the aforementioned problem. Writing the target probability density function $\pi(x)$ as

$$\pi(x) = \frac{\gamma(x)}{Z}, \quad \text{where } Z = \int \gamma(x)dx \tag{5.11}$$

where $\gamma(\bullet)$ is the unnormalized probability density which is assumed to be known only *pointwise*, Z is the normalizing constant which might be unknown. We introduce a *proposal density* $q(x)$ that is known and has a wider support than $\pi(x)$, i.e.

$$\forall x, \pi(x) > 0 \Rightarrow q(x) > 0 \tag{5.12}$$

We have

$$\pi(x) = \frac{w(x) \cdot q(x)}{Z}, \quad \text{where } Z = \int w(x)q(x)dx \tag{5.13}$$

where $w(x)$ called the *unnormalized weight* function is by definition

$$w(x) = \frac{\gamma(x)}{q(x)} \tag{5.14}$$

Note $w(x)$ is known pointwise. One should recognize that Z is the expectation of $w(x)$ taken over $q(x)$, hence it has a Monte Carlo estimator $\hat{Z}$

$$\hat{Z} = \hat{\mathbb{E}}[w(x)] = \frac{1}{N}\sum_{i=1}^{n} w(X^i), \quad X^i \sim q(x) \tag{5.15}$$

Therefore, the *Monte Carlo estimator (importance sampling)* of $\pi(x)$ is

$$\begin{aligned} \hat{\pi}(x) &= \frac{1}{N\hat{Z}}\sum_{i=1}^{N} w(X^i)\delta_{X^i}(x) \\ &= \sum_{i=1}^{N} \frac{w(X^i)}{\sum_{j=1}^{N} w(X^j)}\delta_{X^i}(x), \quad X^i \sim q(x) \end{aligned} \tag{5.16}$$

Remark 5.1. *The fundamental modification here is that now the samples are drawn from a proposal distribution $q(x)$, instead of the target distribution $\pi(x)$.*

Remark 5.2. *The last equality simply means the unnormalized weights have to be normalized before approximating the target distribution.*

The choice of the proposal distribution is important since it has a big impact on the accuracy of the approximation. For example, we use two proposal distributions $q_1(x) = \mathcal{U}(-5, 5)$ and $q_2(x) = \mathcal{N}(0, 1.2)$ to approximate the target distribution $\pi(x) = \mathcal{N}(0, 1)$. The number of samples is 5000 in both cases. The comparison is shown in Figure 5.2. It can be observed that the approximation accuracy is higher when we use $\mathcal{N}(0, 1.2)$.

In summary, a good proposal distribution $q(x)$ should have the following properties:

- $q(x)$ has a wider support than $\pi(x)$
- $q(x)$ should be as close as possible to $\pi(x)$
- it should be easy to draw samples from $q(x)$
- it should be easy to compute $q(X)$ where X is any realization of x

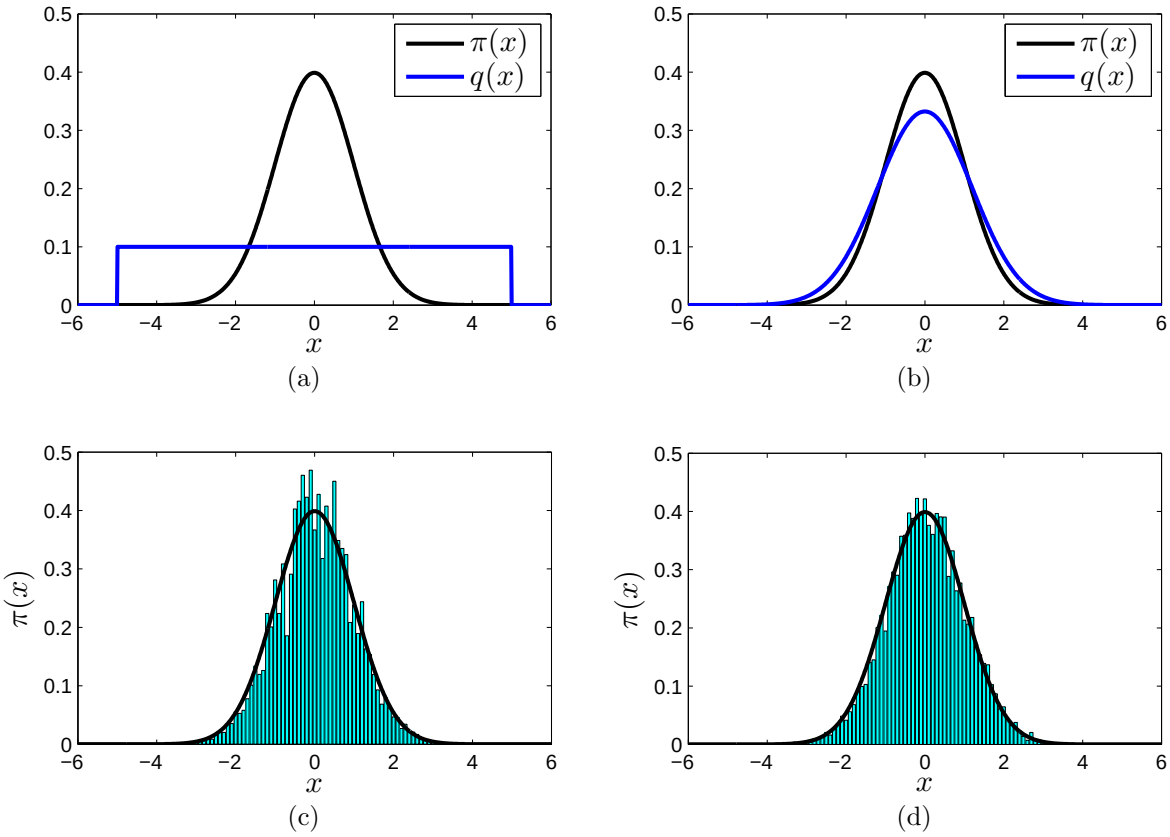


Figure 5.2: Two different proposal distributions used to approximate the target distribution $\mathcal{N}(0,1)$. (a) uniform distribution in $[-5,5]$, (b) normal distribution $\mathcal{N}(0,1.2)$, (c) the approximated target distribution when $q_1(x) = \mathcal{U}(-5,5)$, (d) the approximated target distribution when $q_2(x) = \mathcal{N}(0,1.2)$.

5.4.3 Sequential Importance Sampling

Another problem with the Monte Carlo method is that the random vector may live in a complex multi-dimensional space. Hence sampling in this space is not computationally efficient. For example, samples of $(x_{1:n}, y_{1:n})$ need to be drawn from the aforementioned stochastic model. In this following section, *sequential* importance sampling (SIS) will be discussed in this specific context.

The target distribution is

$$\pi_n \triangleq p(x_{1:n}|y_{1:n}) = \frac{p(x_{1:n}, y_{1:n})}{p(y_{1:n})} \triangleq \frac{\gamma_n}{Z_n} \tag{5.17}$$

where the subscript n indicates the time index. One can use the following proposal distribution

$$q_n = p(x_{1:n}) \tag{5.18}$$

Note that q_n satisfies that

$$\begin{aligned} q_n &= q_{n-1} \cdot p(x_n|x_{1:n-1}) \\ &= q_1 \cdot \prod_{k=2}^{n} p(x_k|x_{1:k-1}) \\ &= q_1 \cdot \prod_{k=2}^{n} p(x_k|x_{k-1}) \end{aligned} \tag{5.19}$$

In order to generate samples $\{X_{n+1}\} \sim p(x_{n+1})$, one simply pass the samples in the previous iteration $\{X_n\}$ through the true process

$$X_{n+1}^i = f(X_n^i, u_n, W_n^i), \quad i = 1, \dots, N \tag{5.20}$$

where W_n^i is an instantiation of the process noise. Hence, sampling can be done recursively.

The weight can also be calculated in a recursive way

$$\begin{aligned} w_n &= \frac{\gamma_n}{q_n} \\ &= \frac{\gamma_{n-1}}{q_{n-1}} \cdot \frac{\gamma_n}{\gamma_{n-1} \cdot p(x_n|x_{n-1})} \\ &= w_{n-1} \cdot \frac{p(x_{1:n}, y_{1:n})}{p(x_{1:n-1}, y_{1:n-1}) \cdot p(x_n|x_{n-1})} \\ &= w_{n-1} \cdot p(y_n|x_n) \end{aligned} \tag{5.21}$$

where the last equality is directly from Equation (5.6). Although $p(y_n|y_{1:n-1})$ is unknown, it is nothing but a normalizing factor. The recursion starts with

$$w_1 = \frac{\gamma_1}{q_1} = \frac{p(x_1, y_1)}{p(x_1)} = p(y_1|x_1) \tag{5.22}$$

The SIS algorithm is summarized as follows

5.4.4 Resampling

We have seen that SIS provides a recursive way to generate samples of $q_n = p(x_{1:n})$. Although the importance weights take into account of the measurements $y_{1:n}$ to make sure that the Monte Carlo estimator is valid, the approximation accuracy will be poor since the proposal distribution will gradually drift away from the target distribution because we sample

Data: Measurements $y_{1:T}$
Result: Posterior distribution $\hat{\pi}(x_n), \quad n = 1, \ldots, T$
for *n = 1:T* **do**
  if *n = 1* **then**
    for *i = 1:N* **do**
      Sample $X_1^i \sim p(x_1)$;
      Compute the weights $w_1(X_1^i) = p(y_1|X_1^i)$;
    end
  else
    for *i = 1:N* **do**
      Sample $X_n^i \sim p(x_n|X_{n-1}^i)$;
      Compute the weights $w_n(X_{1:n}^i) = w_{n-1}(X_{1:n-1}^i) \cdot p(y_n|X_n^i)$;
    end
  end
  $\hat{\pi}(x_n) = \sum_{i=1}^{N} \frac{w_n(X_{1:n}^i)}{\sum_{j=1}^{N} w_n(X_{1:n}^j)} \delta_{X_n^i}(x_n)$;
end

Algorithm 1: Sequential importance sampling algorithm

regardless of the available measurements.

Resampling techniques are the real *trick* of the sequential Monte Carlo method. Resampling transforms a set of particles to another set of the same size but corresponding to a different distribution.

Suppose we are at the n-th step, and we approximate $p(x_{1:n}|y_{1:n})$ using the Monte Carlo estimator $\hat{p}(x_{1:n}|y_{1:n})$ with the weighted samples $X_{1:n}^i$ from $p(x_{1:n})$. For better approximation accuracy in the next step, we want to draw samples from $p(x_{1:n}|y_{1:n})$. We can simply sample from its approximation $\hat{p}(x_{1:n}|y_{1:n})$. Equivalently, we draw with replacements N samples from $\{X_{1:n}\}$ to form a new set $\{\bar{X}_{1:n}\}$. The probability of drawing $X_{1:n}^i$ is given by its normalized importance weight $\frac{w_n(X_{1:n}^i)}{\sum_{j=1}^{N} w_n(X_{1:n}^j)}$. After resampling, the weights need to be reset to $1/N$, i.e. the weights are not accumulated as in SIS. The new Monte Carlo estimator is

$$\hat{p}_{new}(x_{1:n}|y_{1:n}) = \frac{1}{N} \sum_{i=1}^{N} \delta_{\bar{X}_{1:n}^i}(x_{1:n}) \tag{5.23}$$

The three most popular improved resampling algorithms found in the literature are:

1. systematic resampling
2. residual resampling

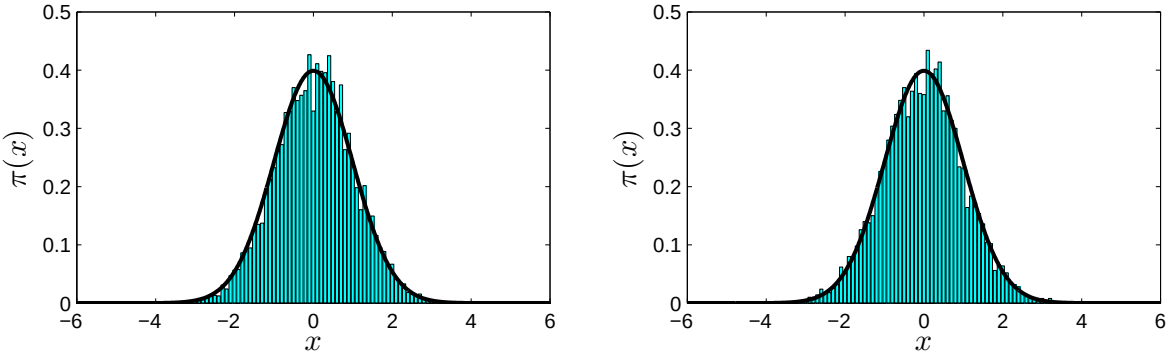


Figure 5.3: The Monte Carlo estimator of $\mathcal{N}(0,1)$, (a) before resampling, (b) after resampling.

3. multinomial resampling

Figure 5.3 shows the Monte Carlo estimator of $\mathcal{N}(0,1)$ using 5000 samples drawn from $\mathcal{N}(0,1.2)$ before and after resampling.

5.4.5 Sequential Importance Resampling

Resampling is equivalent to using a new proposal distribution $p(x_{1:n+1}, y_{1:n})$, as opposed to $p(x_{1:n+1})$. We obtain samples from it by passing $\{\bar{X}_{1:n}\} \sim p(x_{1:n}, y_{1:n})$ through the process model since

$$p(x_{1:n+1}, y_{1:n}) = p(x_{n+1}|x_n) \cdot p(x_{1:n}, y_{1:n}) \tag{5.24}$$

The weight in this case is

$$\begin{aligned} w_{n+1} &= \frac{p(x_{1:n+1}, y_{1:n+1})}{p(x_{1:n+1}, y_{1:n})} \\ &= p(y_{n+1}|x_{n+1}) \end{aligned} \tag{5.25}$$

which is not accumulative.

The algorithm of the SIS with resampling is called the *sequential importance resampling*

(SIR), a.k.a. the *particle filter* (PF). The algorithm is summarized as follows

Data: Measurements $y_{1:T}$
Result: Posterior distribution $\hat{\pi}(x_n), \quad n = 1, \ldots, T$
for n = *1:T* **do**
  if n = *1* **then**
    for i = *1:N* **do**
      Sample $X_1^i \sim p(x_1)$;
      Compute the weights $w_1(X_1^i) = p(y_1|X_1^i)$;
    end
    Resample $\{X_1^i\}_{i=1}^N$ to obtain N new equally weighted samples $\{\bar{X}_1^i\}_{i=1}^N$;
  else
    for i = *1:N* **do**
      Sample $X_n^i \sim p(x_n|\bar{X}_{n-1}^i)$;
      Compute the weights $w_n(X_{1:n}^i) = p(y_n|X_n^i)$ (*does not require* w_{n-1});
    end
    Resample $\{X_n^i\}_{i=1}^N$ to obtain N new equally weighted samples $\{\bar{X}_n^i\}_{i=1}^N$;
  end
  $\hat{\pi}(x_n) = \frac{1}{N}\sum_{i=1}^N \delta_{\bar{X}_n^i}(x_n)$
end

Algorithm 2: Sequential importance resampling algorithm

5.5 Marginalized Particle Filter

We have demonstrated that a recursive algorithm offers computational efficiencies. Another improvement can be done by partitioning the state into two parts and apply the Monte Carlo method to one part, then apply efficient Bayesian filters, such as the KF, to the other. This idea is called *Rao-Blackwellization*. The resulting SIR algorithm is called the *Rao-Blackwellized particle filter*, or the *marginalized particle filter*.

5.5.1 Monte Carlo Estimator

Let us denote the two partitions of the random vector of interest as (x, z). The target distribution $\pi(x, z)$ can be written as

$$\pi(x, z) = p(z|x) \cdot p(x) \tag{5.26}$$

where $p(x)$ is the marginal probability density of x, and $p(z|x)$ is the conditional probability density of z given x. If we are able to draw samples from $p(x)$ and only have an analytical expression for $p(z|x)$, the Monte Carlo estimator (perfect sampling) of the target distribution is

$$\hat{\pi}(x, z) = \frac{1}{N}\sum_{i=1}^{N} p(z|X_i) \cdot \delta_{X^i}(x), \quad X^i \sim p(x) \tag{5.27}$$

Example: Suppose we are to approximate the probability density of a two-dimensional Gaussian random vector

$$\begin{aligned} \pi(x,z) &= \frac{1}{2\pi\sqrt{\det(Q)}} \exp\Big\{ -\frac{1}{2} \begin{bmatrix} z \\ x \end{bmatrix}^T Q^{-1} \begin{bmatrix} z \\ x \end{bmatrix} \Big\} \\ &\mathbb{E}[x] = 0, \quad \mathbb{E}[z] = 0 \\ &\mathbb{E}\Big[\begin{bmatrix} z \\ x \end{bmatrix} \begin{bmatrix} z \\ x \end{bmatrix}^T \Big] = Q = \begin{bmatrix} 1 & 0.7 \\ 0.7 & 1 \end{bmatrix} \end{aligned} \tag{5.28}$$

Using the Monte Carlo method, we can draw samples from $p(x)$,

$$X^i \sim p(x) = \int \pi(x,z)dz = \frac{1}{\sqrt{2\pi}} \exp\{-\frac{1}{2}x^2\} \tag{5.29}$$

It is known that conditional probability density $p(z|X^i)$ is also Gaussian

$$\begin{aligned} p(z|X^i) &= \mathcal{N}(Q_{12}Q_{22}^{-1}X^i, Q_{11} - Q_{12}Q_{22}^{-1}Q_{21}) \\ &= \mathcal{N}(0.7X^i, 0.51) \end{aligned} \tag{5.30}$$

Hence the Monte Carlo estimator will be

$$\hat{\pi}(x,z) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{N}(0.7X^i, 0.51)\delta_{X^i}(x), \quad X^i \sim \mathcal{N}(0,1) \tag{5.31}$$

The estimated density is shown in Figure 5.4. Note it is a combination of slices, each slice being exactly a Gaussian density $\mathcal{N}(0.7X^i, 0.51)$ scaled by $p(X^i)$ approximately. Moreover, the projection of this surface onto $z - \pi(x,z)$ plane will look like $\hat{\pi}(x)$ Figure 5.1.

5.5.2 Linear Substructure Assumption

Moving forward, we will use a more specific system model

$$\begin{aligned} x_{n+1} &= f(x_n) + A(x_n)z_n + B(x_n)w_n \\ z_{n+1} &= z_n + Ge_n \\ y_n &= h(x_n) + v_n \\ x_1 &\sim p(x_1), \quad z_1 \sim \mathcal{N}(\hat{z}_{1|0}, P_{1|0}) \\ w_n &\sim \mathcal{N}(\mathbf{0}, Q_1), \quad e_n \sim \mathcal{N}(0, Q_2), \quad \forall n \\ v_n &\sim \mathcal{N}(\mathbf{0}, R), \quad \forall n \end{aligned} \tag{5.32}$$

where the state is partitioned such that x and z appear nonlinearly and linearly respectively. Also, the process noise w, e and the measurement noise v are white, Gaussian and uncorrelated with each other and (x_1, z_1), and appear linearly in the system. The Bayes' net is shown in Figure 5.6.

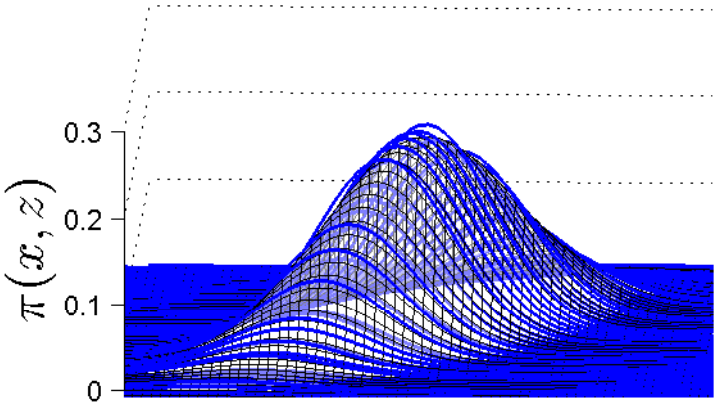


Figure 5.4: The Monte Carlo estimator of a 2D Gaussian random vector is shown in blue. The true density is shown in black.

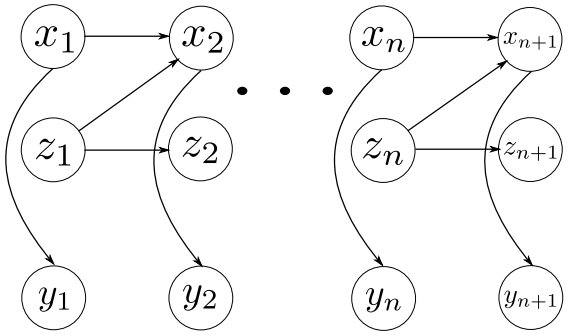


Figure 5.5: The underlying Markov assumption of the new model.

We marginalize out z from the target distribution

$$\pi_n = p(x_{1:n}, z_{1:n}|y_{1:n}) = \underbrace{p(z_{1:n}|x_{1:n}, y_{1:n})}_{\text{KF}} \cdot \underbrace{p(x_{1:n}|y_{1:n})}_{\text{PF}} \tag{5.33}$$

The trick here is that we will use the particle filter to estimate $p(x_{1:n}|y_{1:n})$ while using the Kalman filter to estimate $p(z_{1:n}|x_{1:n}, y_{1:n})$ for tractability. By doing this, we guarantee optimality for the linear sub-structure as well as significantly reduce the computational complexity.

5.5.3 Particle Filtering

Because of the partition of the state, we need to modify Equation (5.6) to get

$$p(x_{1:n}.y_{1:n}) = p(y_n|x_n) \cdot p(x_n|x_{n-1}, y_{1:n-1}) \cdot p(x_{1:n-1}, y_{1:n-1}) \tag{5.34}$$

Note the difference here is $p(x_n|x_{n-1}, y_{1:n-1}) \neq p(x_n|x_{n-1})$ because of the change in the Markov assumption. Suppose at time $n-1$ after resampling, we have samples of equal weights from $p(x_{1:n-1}|y_{1:n-1})$, we can use a proposal distribution $p(x_{1:n}, y_{1:n-1})$

$$p(x_{1:n}, y_{1:n-1}) = p(x_n|x_{n-1}, y_{1:n-1}) \cdot p(x_{1:n-1}, y_{1:n-1}) \tag{5.35}$$

The weight is

$$w_n = \frac{p(x_{1:n}.y_{1:n})}{p(x_{1:n}, y_{1:n-1})} = p(y_n|x_n) \tag{5.36}$$

Therefore, the two density functions we need are $p(x_n|x_{n-1}, y_{1:n-1})$ and $p(y_n|x_n)$. They are given by

$$p(x_n|x_{n-1}, y_{1:n-1}) = \mathcal{N}\Big(f_{n-1} + A_{n-1}\hat{z}_{n-1|n-1},\ A_{n-1}P_{n-1|n-1}A_{n-1}^T + B_{n-1}Q_1B_{n-1}^T\Big) \tag{5.37}$$

$$p(y_n|x_n) = \mathcal{N}\Big(h_n, R\Big) \tag{5.38}$$

where $f_{n-1} \triangleq f(x_{n-1}), h_n \triangleq h(x_n), A_{n-1} \triangleq A(x_{n-1})$, $\hat{z}_{n-1|n-1}$ and $P_{n-1|n-1}$ are the state estimate and the error covariance from the Kalman filter, respectively.

5.5.4 Kalman Filtering

To recursively compute $p(z_{n+1}|X_{1:n+1}^i, y_{1:n+1})$ from $p(z_n|X_{1:n}^i, y_{1:n})$, we can apply the Kalman filter theory directly and think of X_{n+1}^i as the second measurement

$$y_n' = X_{n+1}^i - f_n = A_n z_n + B_n w_n \tag{5.39}$$

Then we can perform the KF *update* with X_{n+1}^i:

$$\begin{cases} \hat{z}_{n|n}^* = \hat{z}_{n|n} + K_n\Big(X_{n+1}^i - f_n - A_n\hat{z}_{n|n}\Big) \\ K_n = P_{n|n}A_n^T\Big(A_nP_{n|n}A_n^T + Q_1\Big)^{-1} \\ P_{n|n}^* = P_{n|n} - P_{n|n}A^T\Big(A_nP_{n|n}A_n^T + Q_1\Big)^{-1}A_nP_{n|n} \end{cases} \tag{5.40}$$

where $f_n \triangleq f(X_n^i), A_n \triangleq A(X_n^i), B_n \triangleq B(X_n^i)$.

And then perform KF *prediction*:

$$\begin{cases} \hat{z}_{n+1|n} = \hat{z}_{n|n}^* \\ P_{n+1|n} = P_{n|n}^* + GQ_2G^T \end{cases} \tag{5.41}$$

The KF *update* with y_n does nothing since y_n does not explicitly depend on z_n

$$\begin{cases} \hat{z}_{n+1|n+1} = \hat{z}_{n+1|n} \\ P_{n+1|n+1} = P_{n+1|n} \end{cases} \tag{5.42}$$

Remark 5.3. *Note that there is a Kalman filter associated with each particle.*

5.5.5 Algorithm

The algorithm of the marginalized particle filter is summarized as follows

Data: Measurements $y_{1:T}$
Result: Posterior distribution $\hat{p}(x_n, z_n|y_{1:n}), \quad n = 1, \dots, T$
for *i = 1:N* **do**
  Sample $X_1^i \sim p(x_1)$;
  Initialize the Kalman filter, setting $\hat{z}_{1|0}^i = \hat{z}_{1|0}$, $P_{1|0}^i = P_{1|0}$;
  Compute the weights $w_1(X_1^i) = p(y_1|X_1^i)$ (Eqn (5.38));
  Kalman filter y_1 update (Eqn (5.42));
end
Resample to get $\{\bar{X}_1^i\}_{i=1}^N$;
for *n = 2:T* **do**
  for *i = 1:N* **do**
    prediction:
      Sample $X_n^i \sim p(x_n|\bar{X}_{n-1}^i)$ (Eqn (5.37);
      Kalman filter X_n^i update (Eqn (5.40));
      Kalman filter prediction (Eqn (5.41));
    update:
      Compute the weights $w_n(X_{1:n}^i) = p(y_n|X_n^i)$ (Eqn (5.38));
      Kalman filter y_n update (Eqn (5.42));
  end
  Resample to get $\{\bar{X}_n^i\}_{i=1}^N$;
  $\hat{p}(x_n, z_n|y_{1:n}) = \frac{1}{N}\sum_{i=1}^N \mathcal{N}(\hat{z}_{n|n}, P_{n|n})\delta_{\bar{X}_n^i}(x_n)$
end

Algorithm 3: Marginalized particle filter algorithm

5.6 Attitude estimation

If we discretize the quaternion kinematics and the bias evolution using a fixed Euler step

$$\begin{aligned} q_{n+1} &= q_n + \frac{\Delta t}{2}\Xi(q_n)\tilde{\omega}_n - \frac{\Delta t}{2}\Xi(q_n)\beta_n - \frac{\Delta t}{2}\Xi(q_n)\zeta_n \\ \beta_{n+1} &= \beta_n + \Delta t \cdot \eta_n \end{aligned} \tag{5.43}$$

It can be readily seen that the filter dynamics in Equation (5.43), and the measurement model in Equation (4.9) are in the form of the marginalized particle filter model (Equation. (5.32)). In particular, the state partition is

$$\begin{bmatrix} x_n \\ z_n \end{bmatrix} = \begin{bmatrix} q_n \\ \beta_n \end{bmatrix} \tag{5.44}$$

Matching with the formulation in these two equations,

$$\begin{aligned} f(x_n) &= q_n + \frac{\Delta t}{2}\Xi(q_n)\tilde{\omega}_n \\ A(x_n) &= B(x_n) = -\frac{\Delta t}{2}\Xi(q_n) \\ G &= \Delta t \cdot I_{3\times 3} \\ h(x_n) &= \begin{bmatrix} A(q_n)r_1 \\ A(q_n)r_2 \\ \vdots \\ A(q_n)r_N \end{bmatrix} \end{aligned} \tag{5.45}$$

A direct implementation of MPF with the definitions of the matrices above will lead to the MPF formulation for attitude estimation. However, there are several important features inherent in this model, which can be seen from the Bayes' net depicted in Figure 5.6. Further exploitation of the underlying linear structure leads to a significantly simplified MPF formulation. The modifications are done in the following steps: (1) KF update for z_n^1, (2) PF propagation, (3) KF update for z_n^2, (4) KF propagation. The detailed MPF procedure is presented in the following subsections.

5.6.1 PF Update

After the most recent vector measurement y_n is obtained, the importance weights are calculated according to,

$$\begin{aligned} e_n^{(i)} &= y_n - \begin{bmatrix} A(q_n^{(i)})r_1 \\ A(q_n^{(i)})r_2 \\ \vdots \\ A(q_n^{(i)})r_M \end{bmatrix} \\ w_n^{(i)} &\propto \exp\Big\{ -\frac{1}{2} {e_n^{(i)}}^T R^{-1} e_n^{(i)} \Big\} w_{n-1}^{(i)} \end{aligned} \tag{5.46}$$

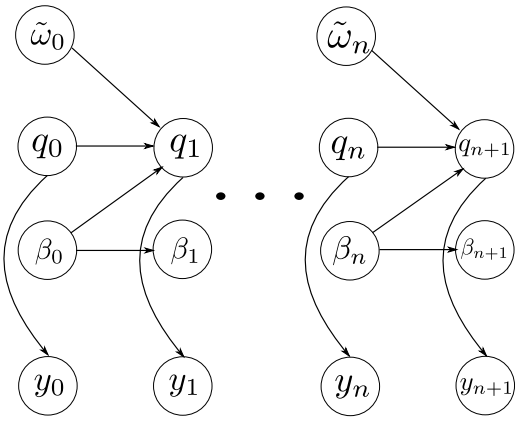


Figure 5.6: The Bayes' net representation of the attitude estimation problem

where $q_n^{(i)}$ is generated from the importance sampling function which will be discussed shortly. The importance weights should be normalized after this step.

5.6.2 Estimate

The a-posteriori bias estimate $\hat{\beta}_n$ can be obtained by the weighted average.

$$\hat{\beta}_n = \sum_{i=1}^{N} w_n^{(i)} \beta_{n|n}^{(i)} \tag{5.47}$$

However, because of the unit norm constraint and the sign ambiguity, the weighted average estimate for quaternions is not optimal. Following Markley *et. al.*[39], the optimal average quaternion is defined as the maximizer of a constrained quadratic programming (equivalently, a weighted sum of the squared Frobunius norms of attitude matrix differences). Hence, the a-posteriori attitude estimate $\hat{q}_n$ is,

$$\begin{aligned} \hat{q}_n &= \arg\max_q q^T L q \\ \text{subject to} \quad & q^T q = 1 \\ \text{where} \quad & L = \sum_{i=1}^{N} w_n^{(i)} q_n^{(i)} {q_n^{(i)}}^T \end{aligned} \tag{5.48}$$

The maximization problem can be solved analytically. $\hat{q}_n$ is an eigenvector of L corresponding to the maximum eigenvalue.

5.6.3 KF Update for y_n

The measurement y_n does not contain any information about the linear state variable β_n. The corresponding KF update cannot be used and thus left out in the algorithm.

5.6.4 PF propagation

The conditional a-priori distribution of the nonlinear state variables is,

$$\begin{aligned} &P(q_{n+1}|q_n^{(i)}, Y_n) = \\ &\mathcal{N}\left(q_n^{(i)} + \frac{\Delta t}{2}\Xi(q_n^{(i)})(\tilde{\omega}_n - \beta_{n|n-1}^{(i)}), \frac{\Delta t^2}{4}\Xi(q_n^{(i)})(P_{n|n}^{(i)} + \bar{Q}_1)\Xi^T(q_n^{(i)})\right) \end{aligned} \tag{5.49}$$

where $\mathcal{N}$ represents a normal distribution. $P(q_{n+1}|q_n^{(i)}, Y_n)$ is actually the importance sampling function in the MPF. One will instantiate N particles $q_{n+1}^{(i)}$ from this distribution. Equivalently, $q_{n+1}^{(i)}$ can also be generated by,

$$\begin{aligned} q_{n+1}^{(i)} &= q_n^{(i)} + \frac{\Delta t}{2}\Xi(q_n^{(i)})(\tilde{\omega}_n - \beta_{n|n}^{(i)} - \upsilon) \\ \upsilon &\sim \mathcal{N}(0_{3\times 1}, P_{n|n}^{(i)} + \bar{Q}_1) \end{aligned} \tag{5.50}$$

5.6.5 KF Update for y'_n

The second measurement y'_n is important because it is the only way that information in y_n can be incorporated in the linear state variable. Based on the following measurement equation

$$y'_n = q_{n+1}^{(i)} - q_n^{(i)} - \frac{\Delta t}{2}\Xi(q_n^{(i)})\tilde{\omega}_n = -\frac{\Delta t}{2}\Xi(q_n^{(i)})\beta_n - \frac{\Delta t}{2}\Xi(q_n^{(i)})\upsilon \tag{5.51}$$

KF update is performed,

$$\begin{aligned} \beta_{n|n}^{(i)} &= \beta_{n|n-1}^{(i)} + K_n(y'_n + \frac{\Delta t}{2}\Xi(q_n^{(i)})\beta_{n|n-1}^{(i)}) \\ P_{n|n}^{(i)} &= P_{n|n-1}^{(i)} - K_n M_n K_n^T \\ M_n &= \frac{\Delta t^2}{4}\Xi(q_n^{(i)})(P_{n|n-1}^{(i)} + \bar{Q}_1)\Xi^T(q_n^{(i)}) \\ K_n &= -\frac{\Delta t}{2}P_{n|n-1}^{(i)}\Xi^T(q_n^{(i)})M_n^\dagger \end{aligned} \tag{5.52}$$

where the pseudo-inverse of M_n, denoted as $M_n^\dagger$ is used in calculating the KF gain. $M_n^\dagger$ is found to be,

$$M_n^\dagger = \frac{4}{\Delta t^2}\Xi(q_n^{(i)})(P_{n|n-1}^{(i)} + \bar{Q}_1)^{-1}\Xi^T(q_n^{(i)}) \tag{5.53}$$

Also, Eqn. (5.51) simplifies the innovation error of KF,

$$y'_n + \frac{\Delta t}{2}\Xi(q_n^{(i)})\beta_{n|n}^{(i)} = -\frac{\Delta t}{2}\Xi(q_n^{(i)})\upsilon^{(i)} \tag{5.54}$$

where $\upsilon^{(i)}$ is the instantiation used to generate $q_{n+1}^{(i)}$. With those, the KF update can be simplified to,

Mean update:

$$\beta_{n|n}^{(i)} = \beta_{n|n-1}^{(i)} + P_{n|n-1}^{(i)}(P_{n|n-1}^{(i)} + \bar{Q}_1)^{-1}\upsilon^{(i)} \tag{5.55}$$

Covariance update:

$$P_{n|n}^{(i)} = P_{n|n-1}^{(i)} - P_{n|n-1}^{(i)}(P_{n|n-1}^{(i)} + \bar{Q}_1)^{-1}P_{n|n-1}^{(i)} \tag{5.56}$$

It should be noted that the KF update does not involve $q_n^{(i)}$.

5.6.6 KF propagation

The Kalman filter propagation equations are given by,

Mean propagation:

$$\beta_{n+1|n}^{(i)} = \beta_{n|n}^{(i)} \tag{5.57}$$

Covariance propagation:

$$P_{n+1|n}^{(i)} = P_{n|n}^{(i)} + \Delta t^2 \bar{Q}_2 \tag{5.58}$$

which imply that the Kalman filter equations (5.55) through (5.58) are independent of the quaternion particles. Furthermore, if each particle's bias covariances share the same initialization $\bar{P}_0$, then only one, instead of N, Riccati recursions is needed i.e the particle index of the covariance can be dropped, which can lead to a substantial reduction in computational complexity.

5.6.7 Steady-state KF for linear state

Combining Eqns. (5.56) and (5.58), we obtain the following algebraic Riccati equation of the a-priori linear state covariance matrix,

$$P_{n+1|n}^{(i)} = P_{n|n-1}^{(i)} + \Delta t^2 \bar{Q}_2 - P_{n|n-1}^{(i)}(P_{n|n-1}^{(i)} + \bar{Q}_1)^{-1}P_{n|n-1}^{(i)} \tag{5.59}$$

Following Kalman filter theory, the steady-state solution P_∞ is guaranteed to exist and be positive definite. Therefore, in the MPF algorithm, there is no need for covariance propagation. Moreover, the mean update (Eqn. (5.55)) uses the steady-state KF gain $K_\infty = P_\infty(P_\infty + \bar{Q}_1)^{-1}$.

5.7 Comparison with PF

In this section, we discuss the differences between the steady-state MPF and the PF for attitude estimation. An algorithmic comparison is summarized in Table 5.1.

	PF	**Steady-state MPF**
Initialization	Initialize particles $q_0^{(i)}, \beta_0^{(i)}$	Initialize quaternion particles $q_0^{(i)}$ $\beta_0^{(i)} = \bar{\beta}_0$ Solve the Riccati equation for P_∞
Update	Sec. 5.6.1	same as PF
Estimate	Sec. 5.6.2	same as PF
Resampling	Sec. 5.4.4	same as PF
Propagation	$\omega_n^{(i)} = \tilde{\omega}_n - \beta_n^{(i)} - \eta^{(i)}$ $q_{n+1}^{(i)} = q_n^{(i)} + \frac{\Delta t}{2}\Xi(q_n^{(i)})\omega_n^{(i)}$ $\beta_{n+1}^{(i)} = \beta_{n+1}^{(i)} + \Delta t\zeta^{(i)}$ $\eta^{(i)} \sim \mathcal{N}(0_{3\times1}, \bar{Q}_2)$ $\zeta^{(i)} \sim \mathcal{N}(0_{3\times1}, \bar{Q}_1)$	$\omega_n^{(i)} = \tilde{\omega}_n - \beta_n^{(i)} - \upsilon^{(i)}$ $q_{n+1}^{(i)} = q_n^{(i)} + \frac{\Delta t}{2}\Xi(q_n^{(i)})\omega_n^{(i)}$ $\beta_{n+1}^{(i)} = \beta_{n+1}^{(i)} + K_\infty \upsilon^{(i)}$ $\upsilon^{(i)} \sim \mathcal{N}(0_{3\times1}, P_\infty + \bar{Q}_1)$

Table 5.1: The algorithmic comparison of PF and steady-state MPF for attitude estimation

In the initialization step, the PF generates particles that represent the a-priori distribution of the state. However, the steady-state MPF only generates the quaternion particles, and the bias particles are set to be the a-priori mean of the bias. As discussed in Sec. 5.6.7, one needs to solve the algebraic Riccati equation for P_∞. The update, estimate, resampling steps for the two filters are exactly the same. In the propagation step, the PF instantiates two random variables $\eta \sim \mathcal{N}(0_{3\times1}, \bar{Q}_2)$ and $\zeta \sim \mathcal{N}(0_{3\times1}, \bar{Q}_1)$ in the gyroscope measurement model. The MPF only instantiates one random variable $\upsilon \sim \mathcal{N}(0_{3\times1}, P_\infty + \bar{Q}_1)$. Although the two algorithms are extremely similar, the steady-state MPF is superior to the PF.

5.8 Simulation results

In this section, we demonstrate the performance of the proposed marginalized particle filter in numerical simulations, in comparison with the EKF (see the algorithm in the appendix) and the PF. We use the same setup as in [22]. The noise parameters for the gyro measurements are given by

$$\begin{aligned} \eta_n &\sim \mathcal{N}(\mathbf{0}_{3\times1}, (\sqrt{10}\times10^{-10})^2 \cdot I_{3\times3}) \\ \zeta_n &\sim \mathcal{N}(\mathbf{0}_{3\times1}, (\sqrt{10}\times10^{-7})^2 \cdot I_{3\times3}) \end{aligned} \tag{5.60}$$

i.e. the rate random walk and angle random walk have standard deviations $\sqrt{10}\times10^{-10} rad/sec^{3/2}$ and $\sqrt{10}\times10^{-7} rad/sec^{1/2}$ respectively.

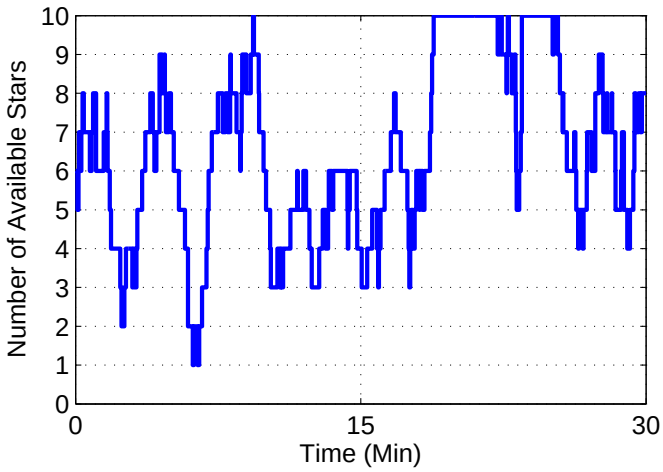


Figure 5.7: Number of available stars

A typical star camera is also used. The star camera can sense up to 10 stars in a $6^o \times 6^o$ field-of-view. If more than 10 stars are sensed, only the first 10 stars (in terms of magnitude, the logarithmic measure of the brightness) are used. A star catalog is used, which stores the positions of the stars in the reference frame as well as their magnitude. Uncorrelated zero-mean Gaussian random variables each with 3-σ value of 0.005 degrees are added to the raw star camera measurements (i.e. star positions in the camera sensor frame). The sampling time for the gyro and the star camera is 1 second. The number of particles used is 3000.

The true initial quaternion is set to be $[0, 1/\sqrt{2}, 0, 1/\sqrt{2}]^T$. The true angular velocity is set to $[-0.0012, 0, 0]^T rad/sec$ (i.e. the spacecraft always rotates along the body x axis and finishes a full revolution in 90 minutes). The number of stars available at each sampling time is shown in Figure 5.7.

Case1: with prior information:
In this case, we assume that some prior information of the attitude and bias is available. Concretely, the initial attitude and bias estimates are unbiased, and the initial covariance for the attitude error is set to $1^2(deg^2)$, the initial covariance for the gyro drift is set to $2^2(deg/hr)^2$. Thus the initial state error covariance in EKF is

$$P_{0|-1} = \begin{bmatrix} (3.05 \times 10^{-5}) \cdot I_{3\times 3} & \mathbf{0} \\ \mathbf{0} & (9.40 \times 10^{-5}) \cdot I_{3\times 3} \end{bmatrix} \tag{5.61}$$

The initial set of particles used in the MPF and the PF is drawn according to this distribution. Figure 5.8 shows the attitude error in Euler angles from the three methods. The MPF and the EKF (overlapping each other in the plot) are able to keep the errors small, which

are within the 3σ boundary computed in the EKF. The yaw error is much larger than the yaw pitch errors, due to the fact that the boresight of the camera is aligned with the body z-axis. The width of the boundary changes due to the number of available stars. The less available star indicates a larger uncertainty. The PF is not able to track the true attitude as well as the other two, especially in the yaw direction. The reason in the author's opinion is that 3000 particles is not enough to span a seven-dimensional space. A similar trend is observed in the bias estimate in Figure 5.9, which shows the bias estimates from the three methods. Again the MPF and the EKF are comparable, while the PF has a relatively larger error in the z direction.

Case2: without prior information:
The proposed MPF formulation is especially powerful and attractive when there is no prior information of the attitude. The EKF has no guarantee of convergence (it may still work with a random initial state estimate and a large error covariance). The MPF and the PF can uniformly sample from four-dimensional space, then normalize the obtained sample to form a unit quaternion. The system parameters are kept the same.

To accommodate larger uncertainties, especially at the beginning, two heuristics discussed in [26] are adopted:

- Use of an initially large but decaying measurement noise covariance:
  If the number of particles is limited and the measurement distribution is too narrow, it is possible that all the weights are tiny. This creates a numerical issue when one wants to normalize the weights. One possible heuristic is to have a large measurement noise covariance at the beginning, and then gradually decreases as the state distribution narrows down. In the simulation, the following time-varying covariance is implemented

$$R' = (1 + 5 \times 10^6 \cdot \exp\{-0.008 \cdot t\}) \cdot R \tag{5.62}$$

- Roughening:
  As mentioned earlier, resampling focuses the particles at the region which has a high density. Consequently, replicates are created for the particles with large weights. Roughening refers to adding small independent jitter to those duplicates, i.e. to increase variation/diversity in the particles.

Figure 5.10 shows the comparison of the convergence performance of the three methods. Empirically, the performance of the EKF really depends on the accuracy of the initial guess, which is random. For benchmarking, we are showing 90^o and 180^o(worst case) initial errors. When the initial state error covariance is not tuned properly, it is also possible for the filter to diverge. From the figure, we observe that the MPF and the PF have a comparable faster convergence rate (overlapping each other in the plot) than the EKF. The estimate error is reduced to around 3^o in one step. Both curves corresponding to the EKF initially drop down and then rise. This is very likely due to overshooting.

5.9 Summary

This chapter derived a steady-state marginalized particle filter for sequential attitude estimation. Marginalizing the linear gyroscope bias increases estimation accuracies. By further exploiting the linear substructure, we show that the bias evolution is independent of the quaternion particles and its covariance reaches a steady-state value, which will reduce the computation complexity. Comparison with the standard particle filter and the extended Kalman filter in numerical simulations validates the superior performance.

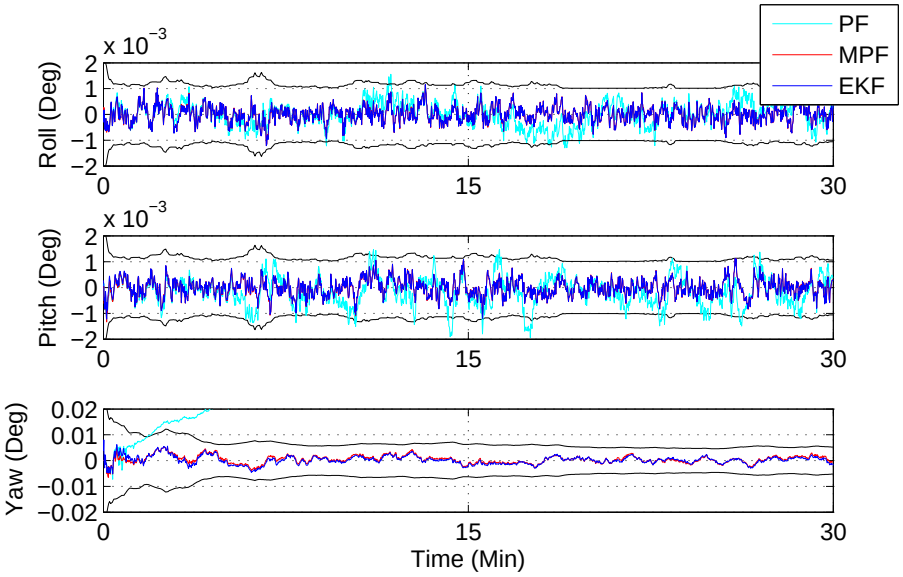


Figure 5.8: Attitude estimate errors in Euler angles from the three methods. The 3σ boundary from the EKF is provided.

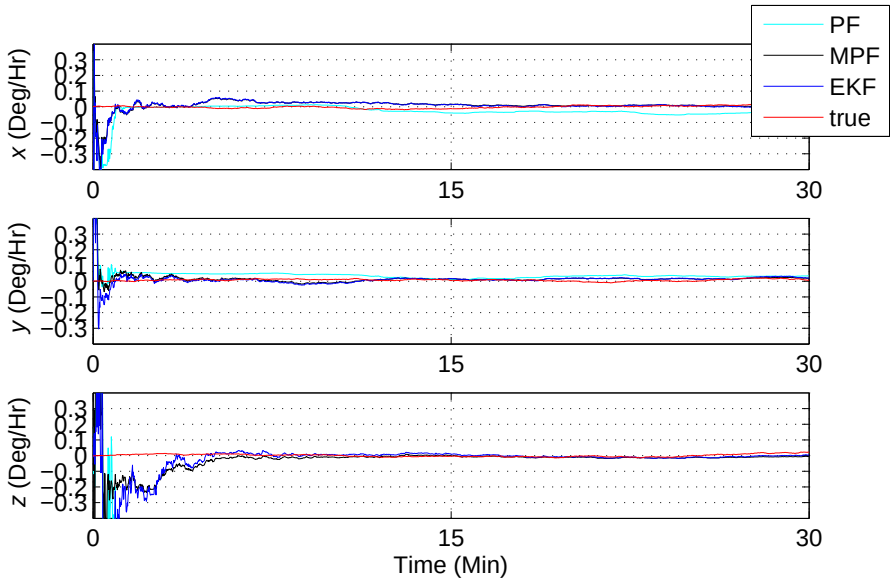
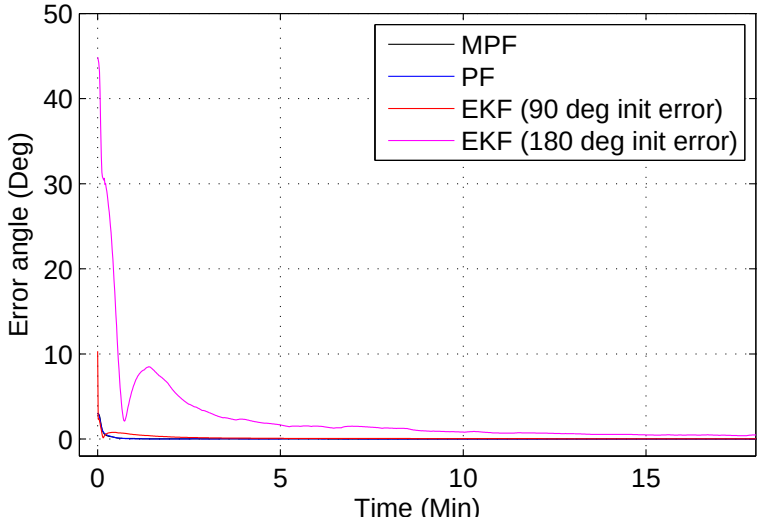
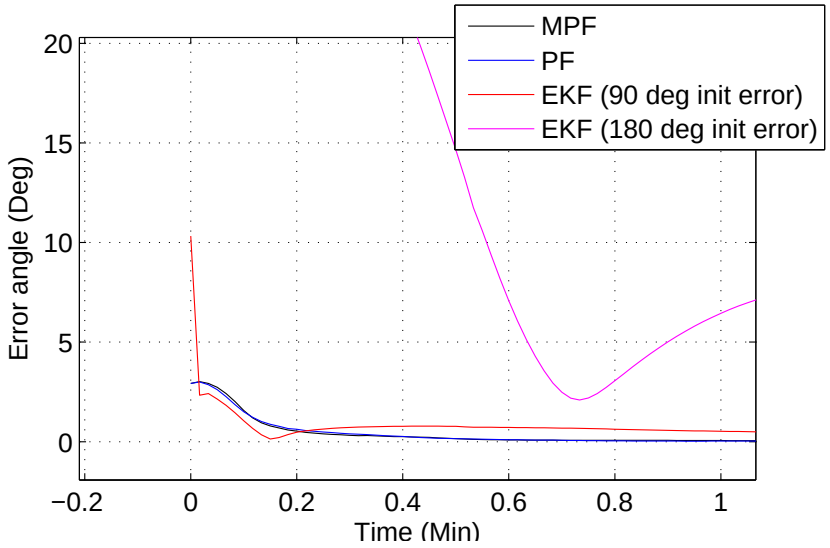


Figure 5.9: Bias estimate from the three methods.



(a)



(b)

Figure 5.10: (a) Absolute error angles from the three methods. (b) Zoom-in view of the errors.

Chapter 6

Complementary filter

An innovative implementation of attitude estimation in 3 degrees of freedom (3-DOF) combining the TRIAD algorithm [20] and a time-varying nonlinear complementary filter (TVCF) is derived. This work is inspired by the good performance of the TVCF in 1-DOF [35] developed for applications limited to small mobile platforms with low computational power. To demonstrate robust 3-DOF estimation, information from vector and rate-gyroscope measurements are fused. Simulation and experimental results demonstrate comparable performance to the extended Kalman filter (EKF) and improved performance over alternative methods such as sole gyroscope rate-integration and the TRIAD algorithm without the TVCF as a pre-filter.

6.1 Introduction

Orientation estimation in three degrees of freedom is useful and necessary for a broad area of applications. Examples include satellites, mobile robots, phones and motion tracking for video games, film industry and sports. Many sensors have been studied for this purpose; for example, inertial measurement units (IMUs), indoor/outdoor GPS, star cameras, vision, ultrasonic senors and range finders [40, 41, 42, 43]. For terrestrial applications, IMUs, while still computationally limited are attractive since they are becoming faster, more compact and affordable. For space/satellite applications, a star camera is convenient since unbiased vector measurements can be acquired from the surrounding stars.

The results of other researchers that have studied 3-DOF orientation estimation [41, 44] are promising, but can be improved in some form. Nonlinear forms of the Kalman filter have also been studied for attitude estimation [45], but the heavy computational complexity can make them less attractive in applications where only small scale processors are available.

We have combined two computationally efficient schemes, the TRIAD algorithm and the time-varying complementary filter (TVCF), and developed an algorithm that shows compa-

rable performance to the EKF with less computional burden. It aims to be implementable on a small portable platform with low computational power in Figure6.8a. This application can benefit from complementary filtering because rate and angle sensors (gyroscope versus accelerometer and magnetometer) possess benefits and drawbacks in different frequency regimes [32, 33, 34]. The TVCF, which uses a fuzzy logic scheme to adjust trust to different sensors, has already been found to be useful for 1-DOF attitude estimation due to its low computational requirement and the ability to discern stationary and motion states [35]. This chapter generalizes the previous work to 3-DOF attitude estimation. We will also show that when the angle measurements are from a star camera, the complementary filtering method with a tuned fixed cutoff frequency also shows comparable performance to the EKF. The experimental results are made possible by the Quanser 3-DOF gyroscope in Figure6.8b, to provide true attitude verification and sensor calibration through high resolution encoders.

The remainder of the chapter is organized as follows. In Sec. 6.2, the concept of complementary filtering is introduced. In Sec. 6.3, the complement filter is applied in 1-DOF attitude estimation problem to demonstrate its effectiveness. In Sec. 6.4, an innovative nonlinear complementary filter structure for 3-DOF attitude estimation is discussed. Sec. 6.5 presents the extension of the fuzzy logic based cutoff frequency scheduling which makes the algorithm suitable for use in many terrestrial applications. The simulation and experimental results are shown in Sec. 6.6.

6.2 Complementary Filter

Complementary filtering is a well-known signal processing method, implemented in various fields. The advantage of using CF is its simple structure yet it still yields good performance. Compared with some optimal filters such as the Kalman filter, the computations required by CF are much less, and can be easily handled by micro-processors such as Arduino. The conventional CF method utilizes linear-time-invariant filters with different frequency characteristics, such that only the reliable components are selectively extracted in the frequency domain.

The block diagram of a general two input complementary filter is shown in Figure 6.1. In the first-order implementation, G_{hp} can be set as a first-order high-pass filter with a fixed cut-off frequency ω_c. Namely

$$G_{hp}(s) = \frac{s}{s + \omega_c} \tag{6.1}$$

The complement low-pass filter is

$$G_{lp}(s) = \frac{\omega_c}{s + \omega_c} \tag{6.2}$$

Note that

$$G_{hp}(s) + G_{lp}(s) = 1, \quad \forall \omega, s = j\omega \tag{6.3}$$

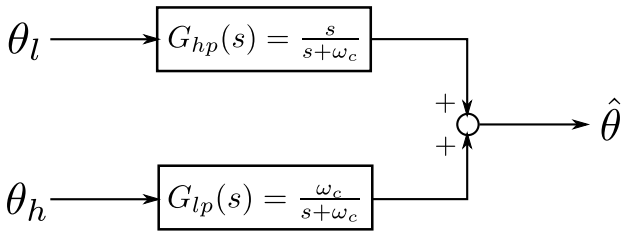


Figure 6.1: Block diagram of a general first-order complementary filter.

If two input signals θ_l and θ_h are reliable at high and low frequencies respectively

$$\begin{aligned} \theta_l(t) &= \theta(t) + \eta_l(t) \\ \theta_h(t) &= \theta(t) + \eta_h(t) \end{aligned} \tag{6.4}$$

where θ is the true value, η_l and η_h are low-frequency and high-frequency noises relative to ω_c. The estimator given by the CF , denoted as $\hat{\theta}$, can be obtained by utilizing the complementary filter to attenuate noises by the corresponding high-pass and low-pass filters

$$\begin{aligned} \hat{\theta} &= G_{hp}(s) \cdot (\theta + \eta_l) + G_{lp}(s) \cdot (\theta + \eta_h) \\ &= \theta + G_{hp}(s) \cdot \eta_l + G_{lp}(s) \cdot \eta_h \end{aligned} \tag{6.5}$$

Note that the signals in the equation above are in the Laplace domain. The actual performance of the filter depends on (i) how separated the frequencies are and (ii) how well ω_c is designed.

The discretized complementary filter can be obtained by applying Tustin's bilinear transformation. The time-domain difference equation is shown below

$$\hat{\theta}(n+1) = \frac{1}{2 + \omega_c \Delta t} \Big((2 - \omega_c \Delta t) \cdot \hat{\theta}(n) + \omega_c \Delta t \cdot (\theta_h(n+1) + \theta_h(n)) + 2 \cdot (\theta_l(n+1) - \theta_l(n)) \Big) \tag{6.6}$$

where Δt represents the sampling period.

6.3 Motivating 1-DOF attitude estimation example

In this section, we discuss the use of CF in determining the attitude in 1-DOF. Consider the case where we want to calculate the angle of a bar which swings from a pivot at one end. The schematic is shown in Figure 6.2. Two sensors are rigidly attached on the bar: a gyroscope and an accelerometer. The CF is used to combine two angle measurements, one from the rate integration and the other from the gravity vector.

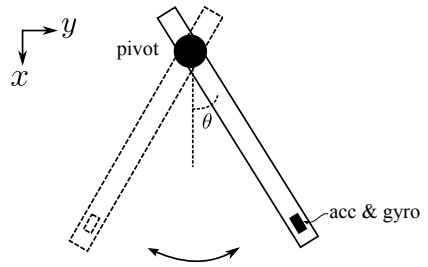


Figure 6.2: A imaginary scenario: a bar swings from a pivot at one end, with two inertial sensors attached at the other end.

By direct integrating the gyroscope measurements, one can write

$$\begin{aligned}\theta_g(t) &= \theta_0 + \int_{t_0}^{t} \omega(\tau) d\tau \\ &= \theta(t) + \underbrace{\int_{t_0}^{t} \beta(\tau) + \eta_v(\tau)\, d\tau}_{\eta_l(t)} \\ &\triangleq \theta_l(t)\end{aligned} \tag{6.7}$$

The second term is the integration of random noises, which has low frequency characteristics. The accelerometer (planar) measures the gravitational acceleration and the motion acceleration expressed in the body frame

$$\begin{cases} a_x(t) = g \cdot \cos\theta(t) + \eta_x(t) \\ a_y(t) = g \cdot \sin\theta(t) + \eta_y(t) \end{cases} \tag{6.8}$$

where η_x and η_y are zero-mean random noises. Motion acceleration is assumed to be negligible. The orientation can be computed simply as

$$\theta_a(t) = \text{atan2}\Big(a_x(t), a_y(t)\Big) \triangleq \theta_h(t) \tag{6.9}$$

The function atan2 produces results in the range $(\pi, \pi]$. This signal is usually reliable at low frequencies, since it is predominately corrupted by measurement noises.

Thus the CF is able to fuse the two estimates to obtain a more accurate estimate compared to each individual

$$\hat{\theta} = G_{hp}(s) \cdot \theta_g + G_{lp}(s) \cdot \theta_a \tag{6.10}$$

We assume that the true angle profile is sinusoidal, with amplitude 60^o and frequency $0.01 rad/sec$

$$\theta = 60^o \cdot \sin(0.01\ t) \tag{6.11}$$

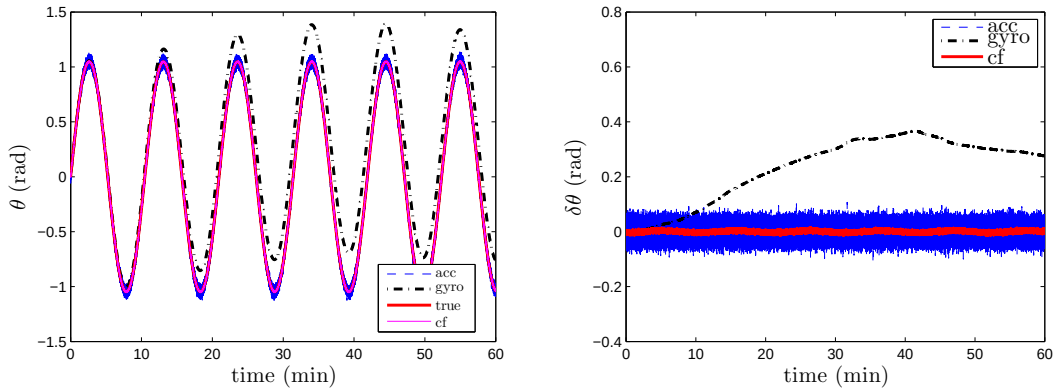


Figure 6.3: (a) Comparison of the angle signals. (b) Comparison of the estimate errors.

The gyroscope measurements are generated by adding the rate random walk and the angle random walk to the true angular rate. The covariances of these random signals are given by

$$q = 6.5 \times 10^{-11} rad^{1/2}/sec, \quad r = 7.6 \times 10^{-10} (rad/sec)^{1/2} \tag{6.12}$$

The accelerometer measures the gravitational vector projected in the body frame. The covariance of the additive Gaussian noise in the accelerometer is assumed to be $5 \times 10^{-4} m^{1/2}/s$. Equation (6.6) is used to fuse the two measurements. The cut-off frequency ω_c is selected to be $3 rad/s$. The comparison is shown in Figures 6.3a and 6.3b. It is clear in the latter plot that the CF is able to attenuate the measurement noise in the accelerometer and eliminate the offset from the rate integration.

6.4 CF in 3-DOF attitude estimation

As we have seen in the motivating example, rate and angle sensors possess noises in different frequency regimes, thus we are motivated to extend the complementary filtering method to attitude estimation in 3 DOF.

6.4.1 Filter structure

In a first-order complementary filter, we extract the useful low frequency information from the vector measurement (introduced in Chapter 4) by passing it through a low pass filter and the useful high frequency information from the rate measurements by passing it through a high pass filter (Figure 6.4a) where both filters have the same cutoff frequency, ω_c. On Earth, the vector measurements $\mathbf{b}_1$ and $\mathbf{b}_2$ in $\mathcal{F}_b$ could be taken by the on-board accelerometer and magnetometer respectively; in space they could be two vector measurements from

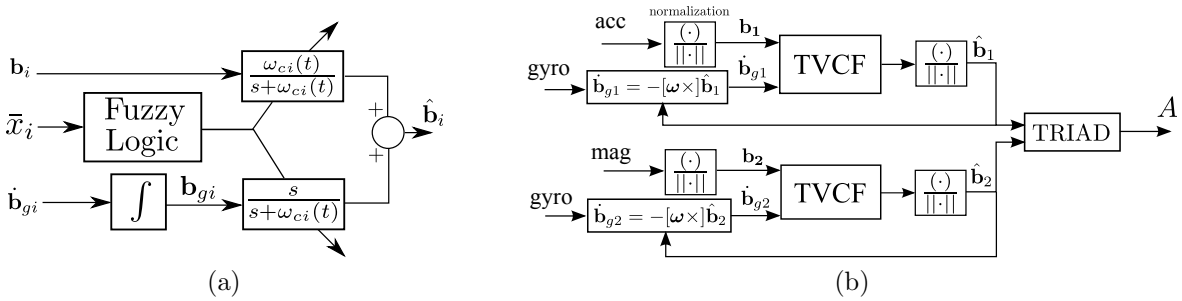


Figure 6.4: (a) Details of the tvcf block. (b) The nonlinear complementary filter used as a prefilter to the triad method used in 3-dof attitude estimation.

a star camera.

In actuality, the components of the rate signal, $\dot{\mathbf{b}}_g$ are passed through a slightly modified high-pass filter (lower channel of Figure 6.4a) that incorporates an integrator,

$$Y(s) = \frac{s}{s+\omega_c}U(s) = \frac{1}{s+\omega_c}\mathcal{L}\{\dot{u}\} \tag{6.13}$$

It is desirable to directly use the measurements from a gyroscope, ω, as the rate signal, however the nonlinearity in the attitude kinematics requires further treatment. We denote $\mathbf{b}_{g1}$ and $\mathbf{b}_{g2}$ as two vector measurements estimated by integrating angular velocity. The rate of change of these two vector measurements can be estimated by the gyroscope measurement using the rotation matrix kinematic relationship

$$\begin{aligned}\dot{\mathbf{b}}_{gi}(k) &= \dot{\mathbf{A}}(k)\mathbf{r}_i \\ &= -[\omega(k)\times]\mathbf{A}(k)\mathbf{r}_i \\ &\approx -[\omega(k)\times]\hat{\mathbf{b}}_i(k), \quad i = 1,2\end{aligned} \tag{6.14}$$

Note that the $\mathbf{r}_i$'s are constant vectors in the reference frame and $\hat{\mathbf{b}}_i$'s are the best estimates of the vectors from the TVCF. A discrete-time complementary filter is obtained by applying the bilinear transformation.

The block diagram of the entire 3-DOF estimation scheme is shown in Figure 6.4b, which can be summarized as follows: the initial measurements of the earth gravitational vector $\mathbf{r}_1$ and the magnetic flux $\mathbf{r}_2$ are recorded during the setup initialization. Namely, the initial attitude is chosen to be the reference frame. Also, the IMU is assumed to be initialized properly so that no bias is present initially. At time step k, the best estimate of attitude is available, or equivalently the $\hat{\mathbf{b}}_i(k)$'s are available. The vectors $\mathbf{b}_1(k+1)$, $\mathbf{b}_2(k+1)$, and $\omega(k+1)$ are measured by the accelerometer and magnetometer or star camera, and gyroscope respectively. Their derivatives $\dot{\mathbf{b}}_{g1}(k+1)$ and $\dot{\mathbf{b}}_{g2}(k+1)$ are calculated from Equation

(6.14). Two parallel time-varying complementary filters then fuse the sensor measurements to obtain the best estimates of the normalized vector measurements

$$\hat{\mathbf{b}}_i(k+1) = \frac{\omega_{ci}(k+1)\Delta t}{2+\omega_{ci}(k+1)\Delta t}(\mathbf{b}_i(k)+\mathbf{b}_i(k+1))+ \\ \frac{2-\omega_{ci}(k+1)\Delta t}{2+\omega_{ci}(k+1)\Delta t}\hat{\mathbf{b}}_i(k) + \frac{\Delta t}{2+\omega_{ci}(k+1)\Delta t}(\dot{\mathbf{b}}_{gi}(k)+\dot{\mathbf{b}}_{gi}(k+1)) \\ i=1,2 \tag{6.15}$$

where Δt denotes the sampling period. With $\hat{\mathbf{b}}_1(k+1)$ and $\hat{\mathbf{b}}_2(k+1)$ obtained from the TVCF, the attitude estimate at time step $k+1$, parameterized by the rotation matrix $\mathbf{A}$, is then calculated by the TRIAD algorithm.

6.5 Fuzzy Logic Based Time-Varying Cutoff Frequency Scheduling

When using an accelerometer and a magnetometer, we can take advantage of their physical properties to adapt the cutoff frequencies between a low and high value based on a fuzzy logic approach proposed by Chang-Siu *et al* [35]. In 1-DOF pitch angle estimation, the magnetometer is not required; for the accelerometer, a fuzzy logic scheme is used to discern the stationary state. In 3-DOF the magnetometer provides true heading information, but the scheme will need to account for disturbances in the magnetic field. The output of the fuzzy logic variable μ, which is valued in the range of $[0, 1]$, signifies how the cutoff frequency shifts between ω_{high} and ω_{low} as

$$\omega_c = \mu\omega_{high} + (1-\mu)\omega_{low} \tag{6.16}$$

Note that if $\mu = 1$ then $\omega_c = \omega_{high}$, which indicates when the accelerometer or magnetometer signal is trusted more than the gyroscope. Conversely, if $\mu = 0$ and $\omega_c = \omega_{low}$, the gyroscope is trusted more. The cutoff frequency decreases when the magnitude of the accelerometer or magnetometer signals deviate from their initial value (indicating rotational or translational acceleration, and magnetic disturbances), and increases when these signals are close to their initial values. The extension from 1-DOF to 3-DOF is simple; just one additional TVCF filter needs to be added in parallel. Therefore, two independent fuzzy logic variables, $\mu_i, i = 1, 2$, need to be computed. The details of calculating μ_1 of the accelerometer channel are the same as the 1-DOF case in [35].

For the magnetometer, the signal $\mathbf{y}_m(t) = [y_{mx}(t), y_{my}(t), y_{mz}(t)]^T$ is considered trustworthy if there are minimal magnetic disturbances. This is operationalized by examining $x_1(t) = ||\mathbf{y}_m(t)|| - ||\mathbf{y}_m(0)||$ and $x_2(t) = \frac{d}{dt}||\mathbf{y}_m(t)||$ and decreasing μ_2 when these signals have a high value, where $||\mathbf{y}_m(t)|| = \sqrt{y_{mx}(t)^2 + y_{my}(t)^2 + y_{mz}(t)^2}$. The IMU is assumed to be

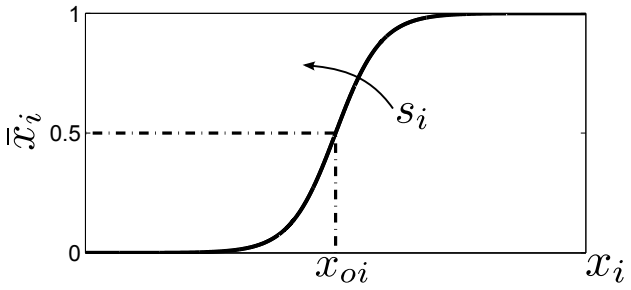


Figure 6.5: The smooth saturation function that computes the fuzzy logic variable. The slope S_I and the 50% threshold x_{oi} can be tuned.

Table 6.1: Fuzzy logic for magnetometer.

Condition	$\bar{x}_1$	$\bar{x}_2$	$\mu_2(t)$
no magnetic disturbance	*low*	*low*	1
constant disturbance	*high*	*low*	0
moving disturbance	*high*	*high*	0

initialized in an environment free of magnetic disturbance. If there is a constant disturbance, such as a large piece of iron, it can be detected by comparing the magnitude of the current magnetometer signal $||\mathbf{y}_m(t)||$ to its initial value $||\mathbf{y}_m(0)||$. Fluctuations in the magnetic field, evidence of a moving disturbance, are detected by examining $\frac{d}{dt}||\mathbf{y}_m(t)||$. This logic is depicted in Tab. 6.1.

More formally, μ_2 in the magnetometer channel is calculated by $\mu_2(t) = \prod_{i=1}^{2}(1 - \bar{x}_i(t))$, where $\bar{x}_1(t) = f_1(||\mathbf{y}_m(t)|| - ||\mathbf{y}_m(0)||)$ and $\bar{x}_2(t) = f_2(\frac{d}{dt}||\mathbf{y}_m(t)||)$. As detailed in [35], f_i is a smooth saturation function shown in Figure 6.5 that converts physical units to logical units bounded by $[0, 1]$ and distinguishes between *high* and *low* based on parameters dependent on the stationary signals.

6.5.1 Performance analysis

The high-frequency noises, present in the accelerometer and magnetometer measurements, are attenuated by the low-pass filters in the algorithm. Consequently, less oscillatory behavior is observed in the attitude estimation.

For the bias present in the gyroscope measurement, we will analytically show that it does not cause the accumulation of error in this estimation scheme unlike the pure rate-integration. However, the filtered vector measurement *does* maintain small steady-state error. To simplify the analysis, we suppose that the body stays steady. Hence, any signals measured by

the gyroscope are just bias β. We also neglect high-frequency noise in the accelerometer and magnetometer in this analysis. Equation (6.15) becomes

$$\alpha\hat{\mathbf{b}}_{ss} = \frac{2\omega_c\Delta t}{2+\omega_c\Delta t}\mathbf{b}_{true} + \frac{2-\omega_c\Delta t}{2+\omega_c\Delta t}\hat{\mathbf{b}}_{ss} - \frac{2\Delta t}{2+\omega_c\Delta t}[\beta\times]\hat{\mathbf{b}}_{ss} \tag{6.17}$$

where $\hat{\mathbf{b}}_{ss}$ is the steady-state estimate from the TVCF. Note that the output from the TVCF does not necessarily have unity norm because of discretization and noises in the accelerometer and magnetometer. However, $\alpha \approx 1$ if fast sampling rate is assumed and noises are neglected. Thus $\hat{\mathbf{b}}_{ss}$ has a closed-form solution

$$\hat{\mathbf{b}}_{ss} \approx (\mathbf{I}_{3\times3} + [\frac{\beta}{\omega_c}\times])^{-1}\mathbf{b}_{true} \tag{6.18}$$

The term in the parenthesis is invertible because its determinant is always greater than or equal to 1. If ω_c is chosen to be much greater than β component-wise, the matrix inverse will be close to the identity. Hence the bias in the gyroscope measurement results in only a small steady-state error in the TVCF estimate. Furthermore, there exists a tradeoff in selecting high or low cut-off frequency. In order to attenuate the high-frequency noise using the low-pass filter, ω_c should be chosen significantly lower than noise frequencies. However, it should be noted that the proposed filter during motion or magnetic disturbances is limited by the quality of the gyroscope.

6.5.2 Comparison with the EKF

Since the above filters only require simple algebraic manipulations, the proposed algorithm does not demand much computational power; the required computations can be handled by low power processors. This is a major reason why the proposed estimation scheme should be found attractive compared with some known optimal Kalman filter formulations when the implementation is on mobile platforms with low computational power. For example, several high-dimensional matrix operations (addition, multiplication and inversion) need to be performed in the multiplicative quaternion EKF. In practice, it takes an 8MHz Arduino Pro Mini processor on average 9 milliseconds to perform either a 6×6 matrix inversion or a multiplication of two 6×6 matrices. Since the EKF requires more than four operations, it cannot be processed within an acceptable sampling rate (i.e. 25Hz in our case).

In terms of filter performances, for terrestrial applications the proposed algorithm significantly simplifies the design of the fuzzy logic law that is used to shift trustworthiness between available sensors, and thus make it possible to easily identify distortions in vector measurements. Although theoretically the EKF perfectly captures the stochastic properties of the system and should produce more accurate estimation, it is extremely difficult in practice to develop some equivalent time-varying noise covariance scheduling in order to account for distortions measured in inertial vector sensors.

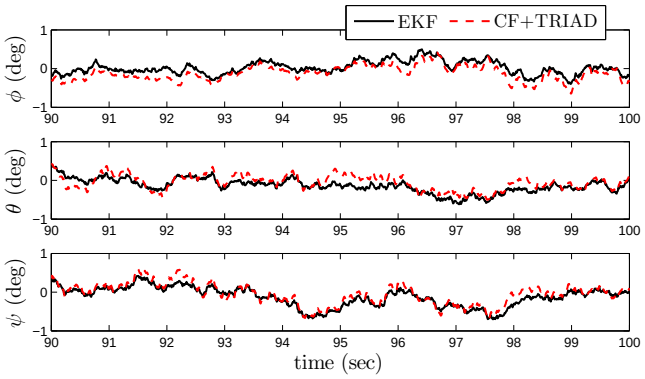


Figure 6.6: Plot of error euler angles from the ekf and cf, showing similar performances between the two methods. ω_c is fixed at $2rad/s$.

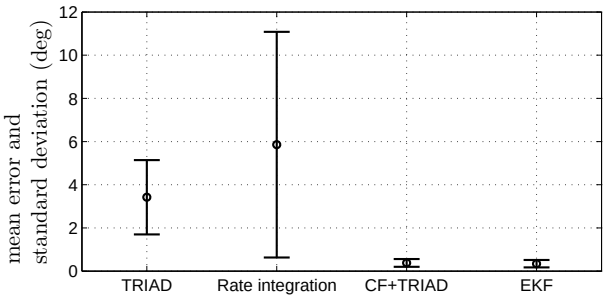
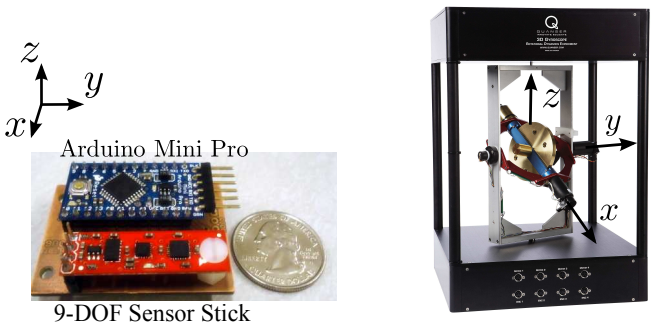


Figure 6.7: Plot of the error and standard deviation by triad, rate integration, cf+triad and ekf. significant improvements are seen using the latter two algorithms.

6.6 Results

6.6.1 Simulation results: 3-DOF spacecraft attitude estimation simulation

Via a simulated space mission, the proposed combination of the complementary filter and the TRIAD algorithm are shown to estimate attitude equally well compared with the multiplicative quaternion EKF formulation, which is detailed in the appendix. The gyroscope model remains the same but the vector measurements $\mathbf{b}_1$ and $\mathbf{b}_2$ are assumed from a star camera that do not possess non-zero bias but only zero-mean Gaussian noises. Hence, a properly tuned *fixed* cutoff frequency is used in the complementary filter approach. An arbitrary true attitude profile is used. Based on the literature [22], the other parameters are set to be: the sampling rate $f = 100$Hz, the gyro noise parameters $\sigma_u = 0.003rad^{1/2}/s, \sigma_v = 0.003(rad/s)^{1/2}$, the star camera measurement noise covariance $\mathbf{R} = 0.03\mathbf{I}_{6\times 6}$, and the initial state error co-



(a) (b)

Figure 6.8: (a) The 9-dof inertial measurement unit and (b) The quanser 3-dof gyroscope (size: $0.7m \times 0.5m \times 0.5m$). Both figures are not the same scale.

variance $\mathbf{P_0} = 0.001\mathbf{I}_{6\times6}$. No initial quaternion and bias estimate errors are present. To evaluate performance, the error Euler angles, $\{\phi, \theta, \psi\}$ are computed by the following equation when they are small

$$\mathbf{A}_{err} \approx \begin{bmatrix} 1 & \psi & -\theta \\ -\psi & 1 & \phi \\ \theta & -\phi & 1 \end{bmatrix} \tag{6.19}$$

From Figure 6.6, it can be seen that the errors from the two methods are quite similar.

The total error angle, regardless of rotation axis, can be calculated from the scalar part of error quaternion as

$$\theta_{tot} = 2\cos^{-1}(\mathbf{q}_{err,4}) \tag{6.20}$$

Figure 6.7 shows the comparison of the mean error and standard deviation from the four different methods. Compared with the sole TRIAD and rate integration methods, the proposed algorithm and the EKF significantly improve the accuracy of the estimation. The proposed algorithm retains good performance while being less demanding in terms of computations.

6.6.2 Experimental results

The performance of the proposed TVCF+TRIAD in 3-DOF attitude estimation is evaluated via experiments. Figure 6.8a shows the 9-DOF IMU developed in the Mechanical System Control Laboratory at UC Berkeley. It has an Arduino Pro Mini microprocessor and a Sparkfun 9-DOF sensor stick (ADXL345 accelerometer, HMC5843 magnetometer, ITG-3200 gyroscope) onboard. The sampling frequency is set to 25Hz. The clock rate of the processor is only 8MHz, but is capable of handling all the required computations including three estimation algorithms in real time. This shows the computational friendliness of the proposed TVCF algorithm. The three estimation schemes are (i) TVCF+TRIAD, (ii) angular rate

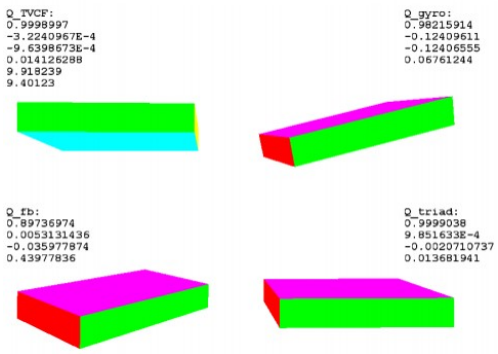


Figure 6.9: Real-time visualization of the estimated attitude. Three benchmarks are displayed at the same time.

integration and (iii) pure TRIAD. In the performance validations shown below, when different cutoff frequencies are compared, the data are collected in real time but the attitude estimates are calculated offline. During the experiment, a graphical user interface (GUI) is used so that the estimated attitude can be visualized in real-time (Figure 6.9).

Effect of the low-pass filter

The low-pass filter attenuates the high-frequency noise in the accelerometer and the magnetometer. Figure 6.10 shows the comparison of the attitude estimates when the IMU is kept steady. The error angles are computed from the quaternions estimated by the two methods using Equation (6.20). The oscillatory behavior is significantly attenuated when the cutoff frequency decreases from infinity (i.e. CF is not utilized) to 0.3 rad/s, which shows the benefit of choosing a small cutoff frequency. Due to the unity magnitude constraint of the quaternion parameterization, noisy measurement signals result in large mean errors. The mean error is thus effectively reduced by the proposed algorithm.

Effect of the high-pass filter

Direct rate integration of the gyroscope measurements is not desirable for applications requiring long period of operation because nonzero bias and noises keep accumulating during the integration. The high-pass filter prevents the accumulation of errors in the gyroscope measurements. The information from angle sensors are utilized to correct the estimate errors. Figure 6.11 shows that the attitude estimate from the rate integration method drifts away, while the CF+TRIAD estimate maintains a bounded error when the IMU is kept steady for a long period of time. This shows the benefit of choosing large cutoff frequencies. Note that the average bias in this particular test is calculated to be $8.3 \times 10^{-4} rad/s$ hence the β/ω_c ratio is of the order -4 when ω_c is selected to be $3 rad/s$. The result confirms our analysis in Equation (6.18).

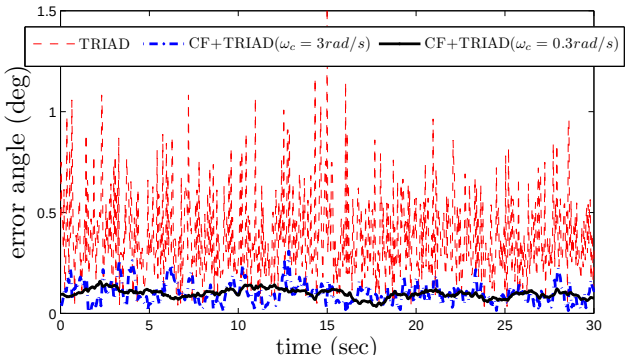


Figure 6.10: Effect of the low-pass filter. The noise attenuation increases after the proposed tvcf prefilter is added to the triad and increases more as the cutoff frequency decreased from $3rad/s$ to $0.3rad/s$.

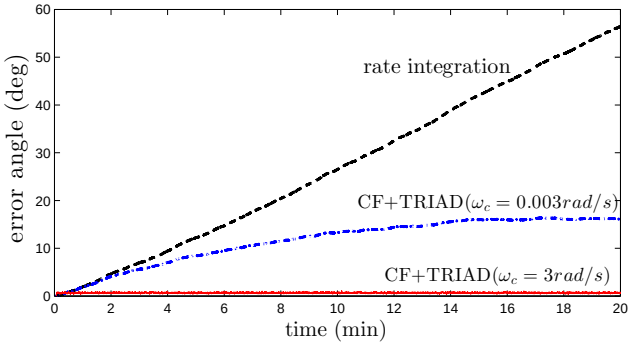


Figure 6.11: Effect of the high-pass filter. As the cutoff frequency in the cf estimate is increased from $0.003rad/s$ to $3rad/s$, the estimate does not drift away when nonzero bias exists.

Selection of the cutoff frequencies

An admissible cutoff frequency range can then be designed. The lower and upper limits of the range are suggested by the test results shown in Figure 6.10 and Figure 6.11 respectively. Namely, the design of the lower value depends on the desired noise attenuation and the upper value depends on the tolerable drift due to the existence of bias. A more rigorous way of choosing cutoff frequency is discussed in [35] where rms estimation errors are experimentally obtained over a grid of cutoff frequencies.

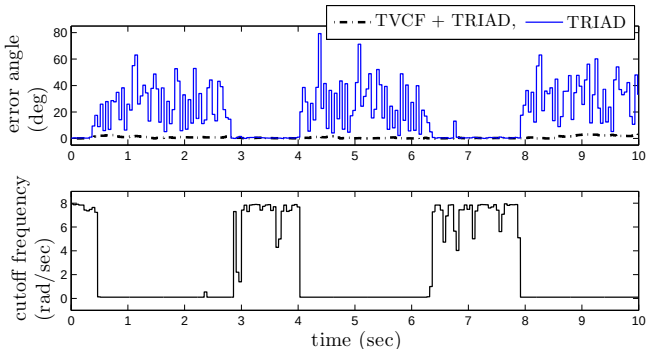


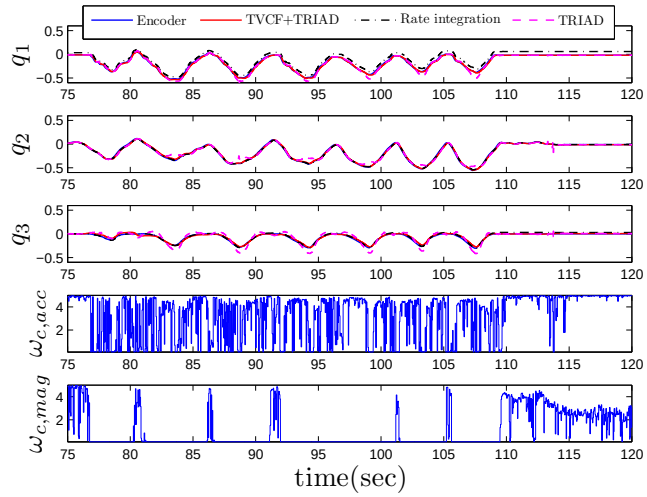
Figure 6.12: The effect of the fuzzy logic to reject motion accelerations. The IMU undergoes pure translational motion in the y-axis. The errors are significantly reduced when the tvcf is added. The upper and lower limits of the cutoff frequency are $8rad/s$ and $0.1rad/s$.

Effect of the fuzzy logic

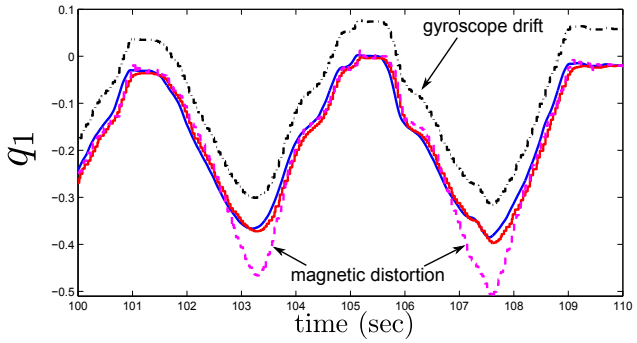
Without the TVCF as a prefilter, the motion acceleration and the magnetic distortion cause the pure TRIAD method to predict incorrect attitude, which is problematic. Figure 6.12 shows that when the IMU undergoes y-axis translational motion, the TVCF correctly identifies the motion acceleration and thus predicts much smaller attitude change compared with the estimate from the sole TRIAD. The lower plot shows the time-varying scheduling of the cutoff frequency in the accelerometer channel. The gyroscope is more trustworthy when the translational motion is detected. Similar behaviors are observed in the presence of temporary magnetic disturbances.

3-DOF attitude estimation experiment

The IMU undergoes a random rotational motion and is attached to the Quanser 3-DOF gyroscope shown in Figure 6.8b, which is equipped with the 5000 lines/rev quadrature optical encoder on each gimbal. The encoder readings can be converted to the quaternion parameterization, providing a true attitude reference for comparison. Magnetic shielding is applied to motors on the testbed to minimize the change of magnetic field due to gimbal movements. However, the magnetic distortion is still large enough to cause problems if the TVCF is not applied. All attitude estimates from the IMU are processed in real time and the quaternions are transmitted to the computer. Figure 6.13a compares the first three components of the quaternions from the encoders, the TVCF+TRIAD algorithm, the rate integration method and the sole TRIAD method. The upper and lower limits of the cutoff frequency are $5rad/s$ and $0.1rad/s$. It can be easily observed that (i) in the q_1 plot, the rate integration drifts in 2 minutes of operation, (ii) in the time intervals where the fuzzy logic shifts the cutoff frequency to its lower limit in the magnetometer channel, the TRIAD method gives inaccurate estimates due to the change of magnetic field (at around 103s and 107s). The



(a)



(b)

Figure 6.13: (a) Plot of the attitude estimates from the different methods and the transitions of the fuzzy logic based cutoff frequencies. (b) Zoomed-in plot of q_1

estimate from the proposed algorithm gives the most accurate attitude estimates among the three methods. Figure 6.14 shows the mean errors calculated from Equation (6.20) and the standard deviation corresponding to the same experiment, where improvements using the proposed algorithm can be concluded. It should also be noted that the effectiveness of the proposed algorithm is underestimated in this experiment because larger errors from the sole TRIAD and the rate integration method will be obtained if the motion accelerations get larger (the centripetal accelerations in this experiment are small due to the small radius of curvature) and if long-term operation is performed, as already demonstrated in Figure 6.11 and 6.12.

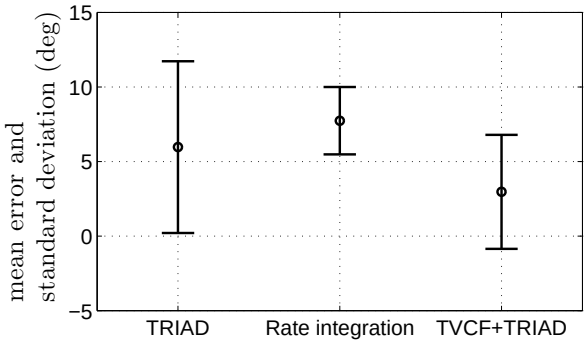


Figure 6.14: Comparison of the estimate errors by the three methods. Among them, the proposed algorithm gives the most accurate attitude estimates.

6.7 Summary

In this chapter, a computationally efficient and tractable method for estimating orientation in 3-DOF is presented. This method can be implemented on many mobile applications and has been shown to be well suited for small, low-cost platforms. The proposed algorithm shows comparable performance to the extended Kalman filter in the simulated space missions and improved performance over alternative methods, such as the rate integration and the TRIAD without TVCF as a prefilter, in inertial measurement applications on Earth. It is capable of not only attenuating noises in different frequency ranges, but also detecting motion acceleration and change of magnetic field, when used in 3-DOF IMUs.

Chapter 7

Conclusions and Future Works

7.1 Concluding Remarks

This dissertation discusses the control algorithms and the estimation algorithms for the attitude control problem.

Adaptive sliding mode attitude control
The adaptive sliding mode attitude control is studied in Chapter 3. The treatment is relatively complete. Both nominal and robust (asymptotic) stability is rigorously proved. Furthermore, the system parameters are adapted in real-time, hence the controller is insensitive to parameter variation.

Marginalized particle filter for attitude estimation
The sequential attitude estimation problem is tackled using Bayesian inference in Chapter 5. Specifically, the marginalized particle filter is found attractive in this case because the system dynamics has a linear sub-structure corresponding to the bias random walk. Furthermore, we show that the bias evolution is decoupled from the quaternion particles. Comparison with the standard particle filter and the extended Kalman filter in numerical simulations validates the superior performance.

Complementary filter for attitude estimation
A computationally efficient and easy-to-tune algorithm is discussed in Chapter 6. Not limited to space missions, this algorithm is desirable for many terrestrial applications since it is implementable on small-scale micro-processors and explicitly deals with noises in inertial sensors (e.g. motion accelerations and magnetic interference). The performance is validated in experiments.

7.2 Topics of Future Research

Parameter convergence
The adaptive control discussed in Chapter 3 guarantees asymptotic convergence of the tracking errors. However, parameter convergence is not guaranteed from the Lyapunov analysis. This can be observed in the simulation results. Recently, Barkana presented a new Invariance Principle for nonlinear nonautonomous systems [46]. The new theorem expands the scope of stability analysis by further mitigating the conditions in the original formulation of the Invariance Principle that are no necessarily needed. It is definitely interesting to explore a relaxed condition for parameter convergence in this new framework, to replace the relatively strong sufficient condition proposed in this dissertation.

Model predictive control
The control algorithm discussed does not explicitly deal with actuator saturation. Instability issue may rise if the computed body torques exceed the actuation authorities. The magnitude of the computed torques may be implicitly controlled via tuning of the controller parameters in practice. Model predictive control becomes a practical methodology for systems of which control saturation needs special attention and minimization of control/power is greatly desirable. The author presented a novel idea of combining model predictive control and sliding mode control [47]. Stability can be rigorously proved for linear systems, but tricky for nonlinear systems. A meaningful extension would be to apply that idea to the attitude control problem, and hopefully design a model predictive controller with stability guarantee.

Appendices

Appendix A

Multiplicative EKF formulation

This appendix presents the derivation of the multiplicative EKF for attitude estimation [22].

If the quaternion estimate is denoted as $\hat{q}$ and the true attitude is q, the *multiplicative* error quaternion in the body frame can be defined as

$$\delta q = q \otimes \hat{q}^{-1}, \quad \text{where } \delta q = [\delta\rho^T,\ \delta q_4]^T \tag{A.1}$$

A.1 Error dynamics

First we take the time derivative of the error quaternion

$$\begin{aligned} \delta\dot{q} &= \dot{q} \otimes \hat{q}^{-1} + q \otimes \frac{d}{dt}(\hat{q}^{-1}) \\ &= \frac{1}{2}\Omega(\omega)\underbrace{q \otimes \hat{q}^{-1}}_{\delta q} + q \otimes \frac{d}{dt}(\hat{q}^{-1}) \\ &= \frac{1}{2}\begin{bmatrix}\omega \\ 0\end{bmatrix} \otimes \delta q + q \otimes \frac{d}{dt}(\hat{q}^{-1}) \end{aligned} \tag{A.2}$$

We now need to determine an expression for $\frac{d}{dt}(\hat{q}^{-1})$, with the short-hand notation $\dot{\hat{q}}^{-1}$. The estimated quaternion follows its kinematics model

$$\dot{\hat{q}} = \frac{1}{2}\Xi(\hat{q})\hat{\omega} = \frac{1}{2}\Omega(\hat{\omega})\hat{q} \tag{A.3}$$

Taking the time derivative of $\hat{q} \otimes \hat{q}^{-1} = [0,\ 0,\ 0,\ 1]^T$ gives

$$\dot{\hat{q}} \otimes \hat{q}^{-1} + \hat{q} \otimes \dot{\hat{q}}^{-1} = 0_{4\times 1} \tag{A.4}$$

Combine the last two equations

$$\frac{1}{2}\Omega(\hat{\omega})\underbrace{\hat{q} \otimes \hat{q}^{-1}}_{[0,\,0,\,0,\,1]^T} + \hat{q} \otimes \dot{\hat{q}}^{-1} = 0_{4\times 1} \tag{A.5}$$

By definition of $\Omega(\hat{\omega})$, this further reduces to

$$\frac{1}{2}\begin{bmatrix}\hat{\omega}\\0\end{bmatrix} + \hat{q} \otimes \dot{\hat{q}}^{-1} = 0_{4\times 1} \tag{A.6}$$

Pre-multiply (in the sense of "$\otimes$") $\hat{q}^{-1}$, we get

$$\dot{\hat{q}}^{-1} = -\frac{1}{2}\hat{q}^{-1} \otimes \begin{bmatrix}\hat{\omega}\\0\end{bmatrix} \tag{A.7}$$

Substituting Equation (A.7) into Equation (A.2), we get

$$\delta\dot{q} = \frac{1}{2}\left\{ \begin{bmatrix}\omega\\0\end{bmatrix} \otimes \delta q - \delta q \otimes \begin{bmatrix}\hat{\omega}\\0\end{bmatrix} \right\} \tag{A.8}$$

We define the error angular velocity

$$\delta\omega = \omega - \hat{\omega} \tag{A.9}$$

Substituting this into Equation (A.8)

$$\begin{aligned}\delta\dot{q} &= \frac{1}{2}\left\{ \begin{bmatrix}\hat{\omega}\\0\end{bmatrix} \otimes \delta q - \delta q \otimes \begin{bmatrix}\hat{\omega}\\0\end{bmatrix} \right\} + \frac{1}{2}\begin{bmatrix}\delta\omega\\0\end{bmatrix} \otimes \delta q \\ &= -\begin{bmatrix}[\hat{\omega}\times]\delta\rho\\0\end{bmatrix} + \frac{1}{2}\begin{bmatrix}\delta\omega\\0\end{bmatrix} \otimes \delta q \\ &\approx \begin{bmatrix}[\hat{\omega}\times]\delta\rho\\0\end{bmatrix} + \frac{1}{2}\begin{bmatrix}\delta\omega\\0\end{bmatrix}\end{aligned} \tag{A.10}$$

The first term is simplified with simple algebraic manipulations, which is exact. The second term is linearized using first-order approximation. This is intuitive because δq is close to $[0,\ 0,\ 0,\ 1]^T$.

The estimated angular velocity is $\hat{\omega} = \tilde{\omega} - \hat{\beta}$, where $\hat{\beta}$ is the estimated bias in the sensor. The bias error is $\Delta\beta = \beta - \hat{\beta}$. The error angular velocity is

$$\delta\omega = \omega - (\tilde{\omega} - \hat{\beta}) = -\Delta\beta - \eta_v \tag{A.11}$$

Hence Equation (A.10) becomes

$$\begin{aligned}\delta\dot{\rho} &= -[\hat{\omega}\times]\delta\rho - \frac{1}{2}(\Delta\beta + \eta_v) \\ \delta\dot{q}_4 &= 0\end{aligned} \tag{A.12}$$

By the small angle approximation $\delta\rho \approx \delta\alpha/2$, where $\delta\alpha$ consists of the small Euler angles (roll-pitch-yaw), we finally reach

$$\delta\dot{\alpha} = -[\hat{\omega}\times]\delta\rho - (\Delta\beta + \eta_v) \tag{A.13}$$

The estimated bias differential equation follows

$$\dot{\hat{\beta}} = 0_{3\times 1} \tag{A.14}$$

Hence

$$\Delta\dot{\beta} = \eta_u \tag{A.15}$$

The EKF error model is thus

$$\Delta\dot{x}(t) = F(\hat{x}(t), t)\Delta x(t) + G(t)w(t) \tag{A.16}$$

where $\Delta x = [\delta\alpha^T,\ \Delta\beta^T]^T, w = [\eta_v^T, \eta_u^T]^T$, and the system matrices are given by

$$F(\hat{x}(t), t) = \begin{bmatrix} -[\hat{\omega}\times] & -I_{3\times 3} \\ 0_{3\times 3} & 0_{3\times 3} \end{bmatrix}, \quad G(t) = \begin{bmatrix} -I_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & I_{3\times 3} \end{bmatrix} \tag{A.17}$$

The process noise covariance matrix is

$$Q(t) = \begin{bmatrix} \sigma_v^2 I_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & \sigma_u^2 I_{3\times 3} \end{bmatrix} \tag{A.18}$$

A.2 Linearized measurement model

N vector measurements can be concatenated to form

$$\begin{gathered} \tilde{y}(t) = h(x(t)) + v(t) = \begin{bmatrix} A(q(t))r_1 \\ A(q(t))r_2 \\ \vdots \\ A(q(t))r_N \end{bmatrix} + v(t) \\ \text{where } \mathbb{E}[v] = 0_{3N\times 1}, \quad \mathbb{E}[vv^T] = \text{diag}[\sigma_1 I_{3\times 3}, \ldots, \sigma_N^2 I_{3\times 3}] \end{gathered} \tag{A.19}$$

The sensitivity matrix $H(\hat{x})$ (i.e. the Jacobian of $h(x(t))$ evaluated at $\hat{x}$) can be derived as follows. Again use the definition of the error quaternion, we get

$$q = \delta q \otimes \hat{q} \quad \Rightarrow \quad A(q) = A(\delta q) \cdot A(\hat{q}) \tag{A.20}$$

When the error is small, the error attitude matrix $A(\delta q)$ and the error Euler angles are related by

$$A(\delta q) \approx I_{3\times 3} - [\delta\alpha\times] \tag{A.21}$$

Thus

$$\begin{aligned} \Delta b &= b - \hat{b} \\ &= A(q)r - A(\hat{q})r \\ &= -[\delta\alpha\times]A(\hat{q})r \\ &= [A(\hat{q})r\times]\delta\alpha + 0_{3N\times 3} \cdot \Delta\beta \end{aligned} \tag{A.22}$$

The sensitivity matrix $H(\hat{x})$ is thus

$$H(\hat{x}) = \begin{bmatrix} [A(\hat{q})r_1\times] & 0_{3\times3} \\ \vdots & \vdots \\ [A(\hat{q})r_N\times] & 0_{3\times3}] \end{bmatrix} \tag{A.23}$$

A.3 Shift posterior information

The derivation of the filter equation is now complete. After taking measurements at each time and run the Kalman filter update, we obtain the a-posteriori estimate

$$\Delta\hat{x}^+ = \begin{bmatrix} \delta\hat{\alpha}^+ \\ \Delta\hat{\beta}^+ \end{bmatrix} \tag{A.24}$$

We need to shift the posterior information to $\hat{q}$ and $\hat{\beta}$

$$\begin{aligned} \hat{q}^+ &= \hat{q}^- + \frac{1}{2}\Xi(\hat{q}^-)\delta\alpha^+ \\ \hat{\beta}^+ &= \hat{\beta}^- + \Delta\hat{\beta}^+ \end{aligned} \tag{A.25}$$

$\Delta\hat{x}^+$ is then set to 0. A brute-force normalization should be performed to insure $\hat{q}^{+T}\hat{q}^+ = 1$.

Bibliography

[1] Wikipedia. *Attitude control*. [Online; accessed 14-April-2015]. 2015. URL: `http://en.wikipedia.org/wiki/Attitude_control`.

[2] Bong Wie and Peter M Barba. "Quaternion feedback for spacecraft large angle maneuvers". In: *Journal of Guidance, Control, and Dynamics* 8.3 (1985), pp. 360–365.

[3] Jean-Jacques E Slotine, Weiping Li, et al. *Applied nonlinear control*. Vol. 199. 1. Prentice-Hall Englewood Cliffs, NJ, 1991.

[4] Kumpati S Narendra and Anuradha M Annaswamy. *Stable adaptive systems*. Courier Dover Publications, 2012.

[5] Thomas AW Dwyer III and Hebertt Sira-Ramirez. "Variable-structure control of spacecraft attitude maneuvers". In: *Journal of Guidance, Control, and Dynamics* 11.3 (1988), pp. 262–270.

[6] Y-P Chen and S-C Lo. "Sliding-mode controller design for spacecraft attitude tracking maneuvers". In: *Aerospace and Electronic Systems, IEEE Transactions on* 29.4 (1993), pp. 1328–1333.

[7] John L Crassidis and F Landis Markley. "Sliding mode control using modified Rodrigues parameters". In: *Journal of Guidance, Control, and Dynamics* 19.6 (1996), pp. 1381–1383.

[8] SR Vadali. "Variable-structure control of spacecraft large-angle maneuvers". In: *Journal of Guidance, Control, and Dynamics* 9.2 (1986), pp. 235–239.

[9] Shih-Che Lo and Yon-Ping Chen. "Smooth sliding-mode control for spacecraft attitude tracking maneuvers". In: *Journal of Guidance, Control, and Dynamics* 18.6 (1995), pp. 1345–1349.

[10] Rush D Robinett and Gordon G Parker. "Spacecraft Euler parameter tracking of large-angle maneuvers via sliding mode control". In: *Journal of guidance, control, and dynamics* 19.3 (1996), pp. 702–703.

[11] Jasim Ahmed, Vincent T Coppola, and Dennis S Bernstein. "Adaptive asymptotic tracking of spacecraft attitude motion with inertia matrix identification". In: *Journal of Guidance, Control, and Dynamics* 21.5 (1998), pp. 684–691.

[12] BT Costic et al. "Quaternion-based adaptive attitude tracking controller without velocity measurements". In: *Journal of Guidance, Control, and Dynamics* 24.6 (2001), pp. 1214–1222.

[13] Amit Sanyal et al. "Inertia-free spacecraft attitude tracking with disturbance rejection and almost global stabilization". In: *Journal of Guidance, Control, and Dynamics* 32.4 (2009), pp. 1167–1178.

[14] Yizhou Wang, Xu Chen, and Masayoshi Tomizuka. "Adaptive Sliding Mode Spacecraft Attitude Control". In: *ASME 2014 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers. 2014, V002T25A001–V002T25A001.

[15] Jean-Jacques E Slotine and Weiping Li. "On the adaptive control of robot manipulators". In: *The International Journal of Robotics Research* 6.3 (1987), pp. 49–59.

[16] JE Slotine and MD Di Benedetto. "Hamiltonian adaptive control of spacecraft". In: *Automatic Control, IEEE Transactions on* 35.7 (1990), pp. 848–852.

[17] John L Crassidis, Srinivas R Vadali, and F Landis Markley. "Optimal variable-structure control tracking of spacecraft maneuvers". In: *Journal of Guidance, Control, and Dynamics* 23.3 (2000), pp. 564–566.

[18] Grace Wahba. "A least squares estimate of satellite attitude". In: *SIAM review* 7.3 (1965), pp. 409–409.

[19] Gerald M Lerner. "Three-axis attitude determination". In: *Spacecraft Attitude Determination and Control* 73 (1978), pp. 420–428.

[20] M. D. Shuster and S. D. Oh. "Three-axis attitude determination from vector observations". In: *Journal of Guidance and Control, vol. 4, Jan.-Feb. 1981, p. 70-77.* 4 (1981), pp. 70–77.

[21] Ern J Lefferts, F Landis Markley, and Malcolm D Shuster. "Kalman filtering for spacecraft attitude estimation". In: *Journal of Guidance, Control, and Dynamics* 5.5 (1982), pp. 417–429.

[22] John L Crassidis and John L Junkins. *Optimal estimation of dynamic systems*. CRC press, 2011.

[23] Eric A Wan and Rudolph Van Der Merwe. "The unscented Kalman filter for nonlinear estimation". In: *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*. IEEE. 2000, pp. 153–158.

[24] John L Crassidis and F Landis Markley. "Unscented filtering for spacecraft attitude estimation". In: *Journal of guidance, control, and dynamics* 26.4 (2003), pp. 536–542.

[25] Arnaud Doucet. *Sequential monte carlo methods*. Wiley Online Library, 2001.

[26] Yang Cheng and John L Crassidis. "Particle filtering for sequential spacecraft attitude estimation". In: *AIAA Guidance, Navigation, and Control Conference and Exhibit*. 2004, pp. 16–19.

[27] Yaakov Oshman and Avishy Carmi. "Attitude Estimation from Vector Observations Using a Genetic-Algorithm-Embedded Quaternion Particle Filter". In: *Journal of Guidance, Control, and Dynamics* 29.4 (2006), pp. 879–891.

[28] Thomas B Schön. "Estimation of nonlinear dynamic systems: Theory and applications". In: (2006).

[29] Yaqiu Liu, Xueyuan Jiang, and Guangfu Ma. "Marginalized particle filter for spacecraft attitude estimation from vector measurements". In: *Journal of Control Theory and Applications* 5.1 (2007), pp. 60–66.

[30] Yizhou Wang, Dennis Wai, and Masayoshi Tomizuka. "Steady-State Marginalized Particle Filter for Attitude Estimation". In: *ASME 2014 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers. 2014, V002T25A002–V002T25A002.

[31] Yizhou Wang et al. "Three Dimensional Attitude Estimation via the Triad Algorithm and a Time-Varying Complementary Filter". In: *ASME 2012 5th Annual Dynamic Systems and Control Conference joint with the JSME 2012 11th Motion and Vibration Conference*. American Society of Mechanical Engineers. 2012, pp. 157–165.

[32] A. J. Baerveldt and R. Klang. "A low-cost and low-weight attitude estimation system for an autonomous helicopter". In: *Intelligent Engineering Systems, 1997. INES '97. Proceedings., 1997 IEEE International Conference on*. 1997, pp. 391–395.

[33] A. El Hadri and A. Benallegue. "Attitude estimation with gyros-bias compensation using low-cost sensors". In: *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*. 2009, pp. 8077–8082.

[34] Robert Grover Brown. *Introduction to random signal analysis and Kalman filtering*. New York: Wiley, 1983.

[35] Evan Chang-Siu, Masayoshi Tomizuka, and Kyoungchul Kong. "Time-varying complementary filtering for attitude estimation". In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. 2011, pp. 2474–2480.

[36] Malcolm D Shuster. "A survey of attitude representations". In: *Navigation* 8.9 (1993).

[37] S.S.Sastry. "Lectures in optimal control and dynamics games". In: *Notes for the course EECS290A, Advanced topics in control theory, University of California Berkeley* (1996).

[38] M. Shuster. "Maximum likelihood estimation of spacecraft attitude". In: *Journal of the Astronautical Sciences* 37 (1989), pp. 79–88.

[39] F Landis Markley et al. "Averaging quaternions". In: *Journal of Guidance, Control, and Dynamics* 30.4 (2007), pp. 1193–1197.

[40] D. Roetenberg, H. Luinge, and P. Slycke. "Xsens MVN: Full 6DOF human motion tracking using miniature inertial sensors". In: *Xsens Motion Technologies BV, Tech. Rep* (2009).

[41] E. R. Bachmann et al. "Design and implementation of MARG sensors for 3-DOF orientation measurement of rigid bodies". In: *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*. Vol. 1. 2003, 1171–1178 vol.1.

[42] W. D. Rencken. "Concurrent localisation and map building for mobile robots using ultrasonic sensors". In: *Intelligent Robots and Systems '93, IROS '93. Proceedings of the 1993 IEEE/RSJ International Conference on*. Vol. 3. 1993, 2192–2197 vol.3.

[43] Hartmut Surmann, Andreas Nüchter, and Joachim Hertzberg. "An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments". In: *Robotics and Autonomous Systems* 45.3-4 (2003). doi: DOI: 10.1016/j.robot.2003.09.004, pp. 181–198.

[44] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan. "Estimation of IMU and MARG orientation using a gradient descent algorithm". In: *Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on*. 2011, pp. 1–7.

[45] J. L. Crassidis, F. L. Markley, and Y. Cheng. "Survey of nonlinear attitude estimation methods". In: *Journal of Guidance Control and Dynamics* 30.1 (2007), p. 12.

[46] Itzhak Barkana. "Defending the beauty of the Invariance Principle". In: *International Journal of Control* 87.1 (2014), pp. 186–206.

[47] Yizhou Wang et al. "Model Predictive Sliding Mode Control for Constraint Satisfaction and Robustness". In: *ASME 2013 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers. 2013, V003T44A005–V003T44A005.