

International Journal of Aerospace Engineering

# Formation Flight Control

Guest Editors: Yu Gu, Giampiero Campa, and Mario Innocenti





---

# **Formation Flight Control**

International Journal of Aerospace Engineering

---

## **Formation Flight Control**

Guest Editors: Yu Gu, Giampiero Campa,  
and Mario Innocenti



---

Copyright © 2011 Hindawi Publishing Corporation. All rights reserved.

This is a special issue published in volume 2011 of “International Journal of Aerospace Engineering.” All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## Editorial Board

Noor Afzal, India  
Brij N. Agrawal, USA  
Saad A. Ahmed, UAE  
C. B. Allen, UK  
N. Ananthkrishnan, Korea  
Hikmat Asadov, Azerbaijan  
Nicolas Avdelidis, Greece  
Hyochoong Bang, Korea  
Wen Bao, Switzerland  
Ronald M. Barrett, USA  
Debes Bhattacharyya, New Zealand  
I. D. Boyd, USA  
Keh-Chin Chang, Taiwan  
Christopher J. Damaren, Canada  
Roger L. Davis, USA  
Gang Ding, China  
Dany Dionne, Canada  
Boris Epstein, Israel  
Jinghong Fan, China  
Jiancheng Fang, China  
Ikkoh Funaki, Japan  
Mohamed Gad-el-Hak, USA  
R. Ganguli, India  
Muhammad R. Hajj, USA  
Kathleen C. Howell, USA  
Hui Hu, USA  
Qinglei Hu, China  
Mikhail S. Ivanov, Russia  
Ratneshwar Jha, USA  
Jacob I. Kleiman, Canada  
Hamid M. Lankarani, USA  
Yunhua Li, China  
T. T. Lim, Singapore  
Rick Lind, USA  
W. W. Liou, USA  
Richard W. Longman, USA  
Enrico C. Lorenzini, Italy  
T. J. Lu, UK  
Siao Chung Luo, Singapore  
Joseph Majdalani, USA  
Pier Marzocca, USA  
Josep Masdemont, Spain  
Philippe Masson, USA  
Franco A. Mastroddi, Italy  
James J. McGuirk, UK  
Giovanni Mengali, Italy  
Achille Messac, USA  
Koorosh Mirfakhraie, USA  
Hong Nie, China  
Christian Oliver Paschereit, Germany  
Chris L. Pettit, USA  
Mark Price, UK  
N. Qin, UK  
Markus Raffel, Germany  
Srinivasan Raghunathan, UK  
Mahmut Reyhanoglu, USA  
Corin Segal, USA  
Kenneth M. Sobel, USA  
K. Sudhakar, India  
Martin Tajmar, Austria  
Paolo Tortora, Italy  
Jean-Yves Trépanier, Canada  
Srinivas R. Vadali, USA  
Linda L. Vahala, USA  
Mehdi Vahdati, UK  
George Vukovich, Canada  
Shaoping Wang, China  
Paul Williams, The Netherlands  
R. K. Yedavalli, USA  
Youngbin Yoon, Korea  
Gecheng Zha, USA  
Youmin Zhang, Canada  
Xiaolin Zhong, USA  
Mei Zhuang, USA

# Contents

---

**Formation Flight Control**, Yu Gu, Giampiero Campa, and Mario Innocenti  
Volume 2011, Article ID 798981, 2 pages

**Guidance Navigation and Control for Autonomous Multiple Spacecraft Assembly: Analysis and Experimentation**, Riccardo Bevilacqua, Marcello Romano, Fabio Curti, Andrew P. Caprari, and Veronica Pellegrini  
Volume 2011, Article ID 308245, 18 pages

**Cascade-Based Controlled Attitude Synchronization and Tracking of Spacecraft in Leader-Follower Formation**, Rune Schlanbusch, Antonio Lori'a, and Per Johan Nicklasson  
Volume 2011, Article ID 151262, 12 pages

**Design of an Extended Interacting Multiple Models Adaptive Estimator for Attitude Determination of a Stereoimagery Satellite**, Hossein Bolandi, Farhad Fani Saberi, and Amir Mehrjardi Eslami  
Volume 2011, Article ID 890502, 19 pages

**Decentralized Model Predictive Control for Cooperative Multiple Vehicles Subject to Communication Loss**, Hojjat A. Izadi, Brandon W. Gordon, and Youmin Zhang  
Volume 2011, Article ID 198308, 13 pages

**Vision-Based Tracking of Uncooperative Targets**, Suresh K. Kannan, Eric N. Johnson, Yoko Watanabe, and Ramachandra Sattigeri  
Volume 2011, Article ID 243268, 17 pages

## Editorial

# Formation Flight Control

**Yu Gu,<sup>1</sup> Giampiero Campa,<sup>2</sup> and Mario Innocenti<sup>3</sup>**

<sup>1</sup> Department of Mechanical and Aerospace Engineering, West Virginia University, Morgantown, WV 26506, USA

<sup>2</sup> The Mathworks, El Segundo, CA 90245, USA

<sup>3</sup> Department of Electrical Systems and Automation, University of Pisa, 56126 Pisa, Italy

Correspondence should be addressed to Yu Gu, [yu.gu@mail.wvu.edu](mailto:yu.gu@mail.wvu.edu)

Received 27 June 2011; Accepted 27 June 2011

Copyright © 2011 Yu Gu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The formation flight control problem has been extensively discussed in recent years with numerous applications on aircraft and spacecraft systems. For aerial vehicles, advantages of performing formation flight have been well documented, including fuel saving, improved efficiency in air traffic control, and cooperative task allocation. For space vehicles, precise control of formation flight will enable future large-aperture space telescopes, variable baseline space interferometers, autonomous rendezvous and docking, and robotic assembly of space structures.

To facilitate further development in this emerging area of research, this special issue includes several papers discussing the most recent developments and ideas in the field, with emphasis on both the theoretical and experimental results. Of the five papers selected, three are in the area of spacecraft formation flight control and two are related to autonomous aerial vehicles. Contents of these papers include modeling and estimation, guidance, navigation, and control (GNC), and experimental techniques related to autonomous formation flight.

The first paper of this special issue “*Guidance navigation and control for autonomous multiple spacecraft assembly: analysis and experimentation*” explores the GNC problem for the in-plane orbital assembly of autonomous multiple spacecraft. The guidance and control strategies are designed to take into account the evolving shape and mass properties of the assembling spacecraft. The proposed approaches were validated via hardware-in-the-loop experiments, using four autonomous 3-degree-of-freedom robotic spacecraft simulators.

The second paper “*Cascades-based controlled attitude synchronization and tracking of spacecraft in leader-follower formation*” presents control strategies for leader-follower

attitude synchronization of spacecraft formations in the presence of disturbances. In a leader-follower formation configuration, the leader spacecraft is controlled to follow a given reference, while a follower spacecraft is controlled to synchronize its motion with the leader. A stability analysis is provided in the paper for both known and unknown but bounded disturbances.

The third paper “*Design of an extended interacting multiple models adaptive estimator (EIMMAE) for attitude determination of a stereo-imagery satellite*” deals with the attitude determination issue for a pair of satellites used in a stereoimaging scenario. For this type of operations, highly accurate and stable pointing maneuvers are needed to be accomplished in a few seconds, which requires the satellites to rotate along a relatively large-angle attitude very quickly. Therefore, different estimation strategies are discussed for two different modes: “maneuvering motion” mode and “uniform motion” mode.

The fourth paper “*Decentralized model predictive control for cooperative multiple vehicles subject to communication loss*” investigates the control of multiple cooperative vehicles with the possibility of communication loss/delay. Such communication issues could lead to poor cooperation performance and unsafe behaviors such as collisions. A decentralized model predictive control (DMPC) architecture is proposed to estimate the tail of neighbor’s trajectory which is not available due to the large communication delays. The concept of the tube MPC is also employed to improve the safety of the fleet against collisions, in the presence of large intervehicle communication delays.

The final paper of this special issue “*Vision-based tracking of uncooperative targets*” presents both the flight test and simulation results of a follower aircraft tracking an

uncooperative leader, using only monocular vision information. The situations with and without the subtended angle information for range estimation are both discussed with different approaches presented.

*Yu Gu*  
*Giampiero Campa*  
*Mario Innocenti*

## Research Article

# Guidance Navigation and Control for Autonomous Multiple Spacecraft Assembly: Analysis and Experimentation

**Riccardo Bevilacqua,<sup>1</sup> Marcello Romano,<sup>2</sup> Fabio Curti,<sup>3</sup> Andrew P. Caprari,<sup>4</sup>  
and Veronica Pellegrini<sup>5</sup>**

<sup>1</sup> Department of Mechanical, Aerospace & Nuclear Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180, USA

<sup>2</sup> Department of Mechanical and Aerospace Engineering and Space Systems Academic Group, Naval Postgraduate School, Monterey, CA 93943-5100, USA

<sup>3</sup> Dipartimento di Ingegneria Aerospaziale e Astronautica, Scuola di Ingegneria Aerospaziale, Università di Roma "La Sapienza", 00138 Roma, Italy

<sup>4</sup> Department of Mechanical and Aerospace Engineering, Naval Postgraduate School, Monterey, CA 93943-5100, USA

<sup>5</sup> Department of Applied Mathematics and Statistics, University of California, Santa Cruz, CA 95064, USA

Correspondence should be addressed to Riccardo Bevilacqua, bevilr@rpi.edu

Received 31 August 2010; Revised 22 October 2010; Accepted 23 December 2010

Academic Editor: Giampiero Campa

Copyright © 2011 Riccardo Bevilacqua et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This work introduces theoretical developments and experimental verification for Guidance, Navigation, and Control of autonomous multiple spacecraft assembly. We here address the in-plane orbital assembly case, where two translational and one rotational degrees of freedom are considered. Each spacecraft involved in the assembly is both chaser and target at the same time. The guidance and control strategies are LQR-based, designed to take into account the evolving shape and mass properties of the assembling spacecraft. Each spacecraft runs symmetric algorithms. The relative navigation is based on augmenting the target's state vector by introducing, as extra state components, the target's control inputs. By using the proposed navigation method, a chaser spacecraft can estimate the relative position, the attitude and the control inputs of a target spacecraft, flying in its proximity. The proposed approaches are successfully validated via hardware-in-the-loop experimentation, using four autonomous three-degree-of-freedom robotic spacecraft simulators, floating on a flat floor.

## 1. Introduction

The technical difficulties presented by the autonomous multiple spacecraft assembly problem relate to the development of robust and reliable guidance, navigation, and control techniques for on-orbit evolving systems. The main open challenges are: (1) propellant-efficient control of an assembling (also known as evolving system), the evolution occurring both in its mass and inertia properties, as well as in its sensors and actuators configuration and (2) accurate relative navigation among the spacecraft, especially in the event of low frequency measurements update and interruptions of measurements due, for example, to relative sensors' view's obstruction by other spacecraft. The works of [1–4]

address specifically the problem of a system's evolution and its control. In [5], more emphasis is given to a potential solution for the wireless connectivity of different parts intended for the assembly of a bigger spacecraft, where a Wi-Fi bridge acts as the only real "assembly." Furthermore, wireless capability is becoming a more relevant option for exchanging data amongst close proximity spacecraft which eventually dock to each other (see [6]). Also, the high-risk situation of an assembly maneuver in space does not leave room for computationally intensive logics, such as optimal controllers (see [7]). Onboard CPUs must allocate most of their performance capabilities to platform safety issues.

The use of Commercial Off-The-Shelf (COTS) relative sensors, such as low-cost cameras, justify the need for

robust relative navigation schemes. Many different filters for tracking a maneuvering target have been considered in the literature.

Approaches based on Kalman filter include the work of Singer [8], in which the target acceleration is modeled as a random process with known exponential autocorrelation. The input estimation approach for tracking a maneuvering target is proposed by Chan and Couture [9]. In this approach, the magnitude of the acceleration is identified by the least-squares estimation when a maneuver is detected. The estimated acceleration is then used in conjunction with a standard Kalman filter to compensate the state estimate of the target. The standard filter alone is used during periods when no maneuver takes place. The augmented filtering approach is proposed by Bar-Shalom et al. [10, 11]. In this approach, the state model for the target is changed by introducing extra state components, the target's accelerations. The maneuver, modeled as an acceleration, is estimated recursively along with other states associated with position and velocity while a target maneuvers. Bogler [12] used this method as an implementation on high maneuver target tracking with maneuver detection.

The input estimation filter and the augmented-dimension filter are commonly used in view of their computational efficiency and tracking performance. Among input estimation techniques, the Augmented State estimation approach yields reasonable performance without constant acceleration or small sampling time assumptions. Furthermore, it not only provides fast initial convergence rate, but it can also track a maneuvering target with fairly good accuracy as mentioned by Khaloozadeh and Karsaz [13]. Bahari et al. [14] and Bahari and Pariz [15] propose an intelligent error covariance matrix resetting, by a fuzzy logic approach, necessary for high maneuvering target tracking, to improve the estimation of the target state.

In space applications, particularly in the spacecraft relative navigation for the autonomous rendezvous and assembly, each vehicle is both the target and the chaser for the other spacecraft. Here, an additional challenge is considered: the frequent loss of communications for the data exchange when the application involves more than one spacecraft. Alternatives means to perform relative navigation may include a vision-based system. These types of sensors require the image processing and may result in low frequency measurement updates, especially for small spacecraft with limited computation capabilities. Such sensors suffer of problems such as limitations on the field of view and/or other spacecraft obstructing the view. Furthermore, each vehicle does not usually know the other vehicles inputs, that is, it does not possess the information about the maneuvers performed by its fellow spacecraft. This missing information needs to be reconstructed in the estimation scheme that would otherwise diverge quickly.

We here focus on the utilization of low frequency update and low-cost sensors, such as COTS devices. In particular, the spacecraft are envisioned to have range and line of sight measurements, and relative attitude measurements. The navigation algorithm here presented build upon our preliminary work of [16].



FIGURE 1: Multispacecraft testbed at the Spacecraft Robotics Laboratory of the Naval Postgraduate School.

In this work, we build upon known techniques in order to develop guidance, navigation, and control approaches to perform three-degree-of-freedom spacecraft assembly maneuvers. Furthermore, the suggested methodologies are validated via hardware-in-the-loop testing, using four robotics spacecraft simulators.

In particular, the guidance and control problems are tackled by continuously linearizing the dynamics about the current relative state vector between two spacecraft, and employing a Linear Quadratic Regulator to suboptimize propellant consumption. The LQR weighting matrices are computed in real time, depending on the relative state vector, acting as a feedback control. The LQR real-time solver developed for this research is an extension of what used during a real on-orbit spacecraft test inside the International Space Station [17], where a simplified problem-targeted LQR was executed (a version of the LQR Simulink solver for both RTAI Linux and xPC Target is available for download) (see [18]). While the system evolves, changing its mass properties and actuators' configuration, the LQR-based approach remains unaltered, controlling the growing structure by the simple online modification of a few parameters when a new spacecraft is docked.

As for the relative navigation, we here propose a design based on the augmented state estimation technique. Robustness to frequent signal loss and/or darkening of the sensors is achieved. Furthermore, the suggested approach reconstructs the information of the other vehicles' maneuvers. A spacecraft is envisioned to run a copy of the augmented state estimation technique algorithm for each other spacecraft in the bunch, every vehicle being chaser and target at the same time.

For the experimental part of this work, two dynamic models for the relative navigation filter are considered: (1) the classical Kalman filtering technique, [19], in which the unknown input (the maneuver command) is modeled as a random process and (2) the augmented state estimation technique, where the maneuver is estimated, using a Kalman filter scheme [19], in real time, as an additional variable in an augmented state vector.

Between the two approaches, the second one proves to be the most successful. It yields satisfactory performances

without constant accelerations or small sampling time assumptions. Furthermore, it does not only provide fast initial convergence rate, but it can also track a maneuvering target with a good accuracy under unpredictable loss of the data link and slow data rate, allowing the spacecraft to perform critical maneuvers such as the docking and the multivehicle assembly.

The successful results of the study here presented pave the way for further research and implementation of the new GNC techniques for the full six degrees of freedom spacecraft relative motion.

Main contributions of this work to the state-of-the-art for multiple spacecraft assembly GNC are as follows.

- (1) Development of a guidance and control approach flexible to mass and actuators' configuration changes during the assembly. The methodology is based on a suboptimal LQR for propellant-efficient rendezvous and docking maneuvers.
- (2) Implementation of a spacecraft relative navigation scheme based on augmented state estimation, robust to low frequency measurements updates. In particular, the spacecraft are envisioned to have the availability of range and line of sight measurements, and relative attitude measurements. No relative velocities measurements are available. This is the first time, to our knowledge, that augmented state vector estimation is used for spacecraft relative navigation.
- (3) The first (to the best of authors' knowledge) hardware-in-the-loop laboratory experiment involving four spacecraft simulators in a completely autonomous assembly maneuver.

The paper is organized as follows. Section 2 presents the robotic spacecraft simulators employed for the experiments. Section 3 presents the equations of the three-degree-of-freedom motion for the spacecraft relative maneuvering. Section 4 presents the augmented estimation approach and demonstrates the observability of the augmented state. Section 5 illustrates the guidance and control. Section 6 describes how navigation and control are performed once more spacecraft are assembled. Section 7 is dedicated to the experimental validation of the proposed methodologies. Section 8 concludes the paper.

## 2. Third Generation Spacecraft Simulators at the Spacecraft Robotics Laboratory

This section introduces the third generation of spacecraft simulators developed at the Spacecraft Robotics Laboratory of the US Naval Postgraduate School. Figure 1 shows the fleet of operational spacecraft simulators. The simulators float using air bearings over a very smooth epoxy floor, reproducing a nearly frictionless and weightless environment in two dimensions and three degrees of freedom, that is, two degrees of freedom for the translation and one for the rotation. This experimental testbed allows for the partial verification of guidance, navigation, and control algorithms in a simulated in-plane close proximity flight condition

[20]. For more details on the different families of spacecraft simulators employed throughout the world, we address the reader to [6, 16, 20–23].

In order to perform docking experimentations, two separate custom designed docking interfaces have been developed and each is currently undergoing experimental testing (see Figure 2).

The type 1 docking interfaces are designed in order to passively connect the spacecraft through electromagnetic mechanisms, and their design will allow data/power/fluids exchange (see Figure 3). Conversely, the type 2 design lacks the afore mentioned characteristics but enhances the robustness on the docking concept by correcting residual translational and rotational errors developed during the final docking phase of the spacecraft assembly for experimentation. This second design hosts two small permanent magnets to provide a final docking force and to keep the robots physically connected.

Other key features of the spacecraft simulators include the following.

- (1) Ad-hoc wireless communication. Continuous data exchange amongst each simulator and the external environment over the wireless network provides for in situ communication. This greatly increases the robustness of data collection in the event of communication loss with one of the simulators.
- (2) Modularity. The simulators are divided into two modules where the payload can be disconnected from the consumables, thus allowing for a wide range of applications with virtually any kind of different payload (Figure 4).
- (3) Small footprint. The .19 m length  $\times$  .19 m width of each simulator allows for the working area ( $\sim 5 \text{ m} \times 5 \text{ m}$ ) on the epoxy floor to be optimally exploited.
- (4) Light weight.  $\sim 10 \text{ kg}$ .
- (5) Rapid Prototyping. The capability to rapidly reproduce further generations of simulators and improve existing designs via computer aided design (CAD) with the in house STRATASYS 3D printing machine.

Most notably, point 1 of the previous list has provided an invaluable contribution to the success of our ongoing experimentation. The ad-hoc wireless communication system, currently employed onboard the simulators, was experimentally verified by a distributed computing test, which demonstrated the wireless communication real-time capability for the SRL (see [6]).

Table 1 illustrates the characteristics of the electronics used onboard each spacecraft simulator. The PC104 (onboard computer), the sensors, and the actuators are described below (see [6]).

Each robot performs absolute navigation in the laboratory environment employing indoor pseudo-GPS for position, and magnetometer and gyroscope for attitude (Table 1). The measurements from these sensors are processed by two separate Linear Digital Kalman Filters, estimating position and velocity of the center of mass with respect to the

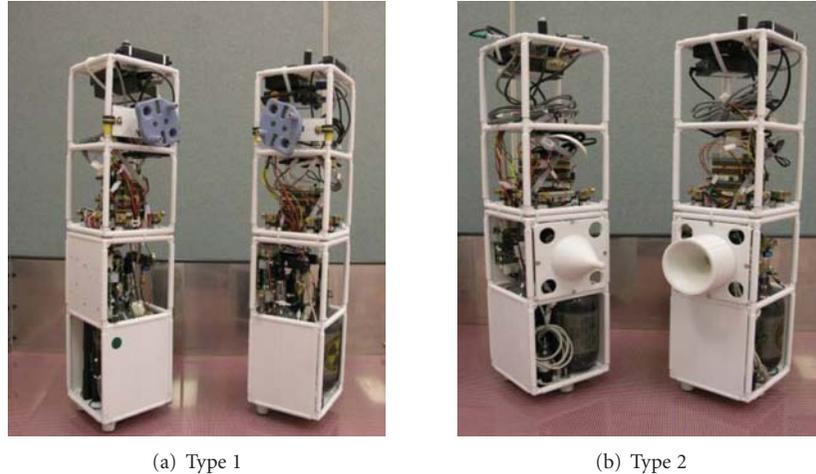


FIGURE 2: (a) Patent pending docking interface design (electromagnet and fluid transfer capability). (b) Concept (male/female) docking interface used for the experiments in this paper.

TABLE 1: Electronics hardware description.

Part's name and manufacturer	Details	Description
PC104 (plus) Motherboard (Advanced Digital Logic)	Processor RAM	SmartCoreT3-400, 400 Mhz CPU SDRAM256-PS
Compact Flash (SanDisk Extreme IV)	—	8 Gbyte capacity
20 Relays Board (IR-104-PBF) (Diamond Systems)	—	High-density optoisolated input + relay output
8 Serial Ports Board (MSMX104+) (Advanced Digital Logic)	—	—
Firewire PC104 board (Embedded Designs Plus)	—	IEEE1394 card with 16 Bit PC104
Compact Wireless-G USB Adapter (Linksys)	—	54 Mbps 802.11 b/g wireless USB network interface adapter
Wireless Pocket Router/AP DWL-G730AP (D-Link)	—	2.4 Ghz 802.11 g, ethernet to wireless converter
Solenoid Valves (Predyne)	—	2 way, 24VDC, 2 Watt
Fiber Optic Gyro DSP3000 (KVH)	—	Single axis rate, 100 Hz, Asynchronous, RS-232
Magnetometer, MicroMag-3Axis (evaluation kit with RS232 board) (PNI)	—	Asynchronous, RS-232 (the evaluation kit is still a development version)
DC/DC converters: EK-05 Battery Controller and Regulator + DC1U-1VR 24V DC/DC Converter (Ocean Server)	—	3.3, 5, 12, 24 Volts outputs. The main board is equipped with a batteries' status controller.
Battery (Inspired Energy)	—	Lithium Ion Rechargeable battery (95 Whr)
Metris iGPS pseudo-GPS indoor system	—	—

laboratory reference frame, and heading and heading rate of the robot with respect to the laboratory frame. The details on the robots' absolute navigation are beyond the scope of this work, and they will not be discussed here; for additional information, the reader can refer to [20].

The maximum computational power of 400 Mhz listed in Table 1 is not required for real-time recomputation of the LQR solution. In the SPHERES satellites [17], Texas Instruments C6701 Digital Signal Processor is employed to solve a very comparable problem.

Figure 5 depicts the main concept of the testbed at the SRL. The main components and their interfaces are illustrated onboard the robot at the bottom of the sketch. Furthermore, the figure emphasizes the fact that the configuration is scalable to an arbitrary number of robots depending on the application or mission.

The Wi-Fi capability of each robot is not only used to communicate with other robots, but it is also necessary for receiving its own absolute position within the laboratory, as sensed by the pseudo-GPS indoor system.

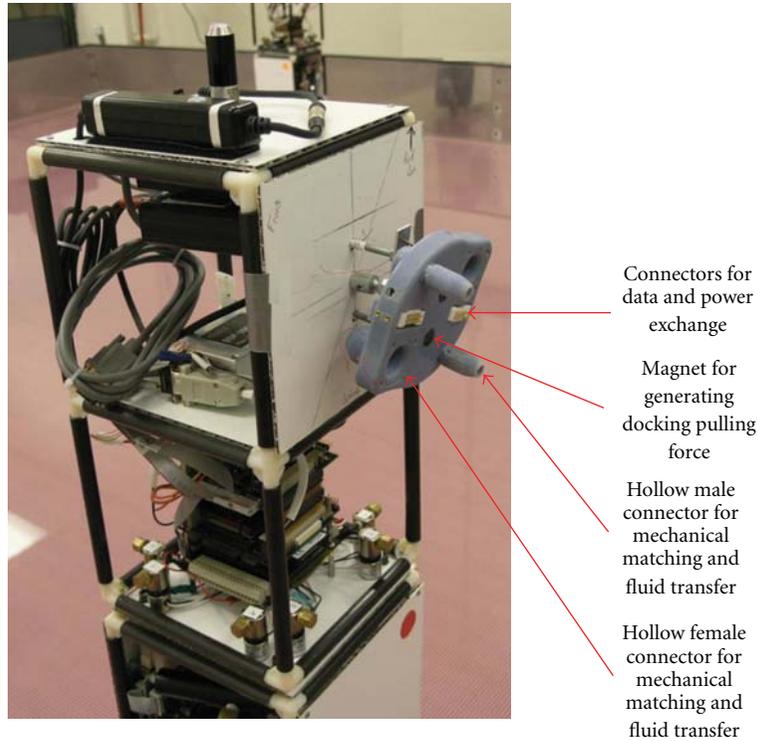


FIGURE 3: Main components of the patent pending docking interface.

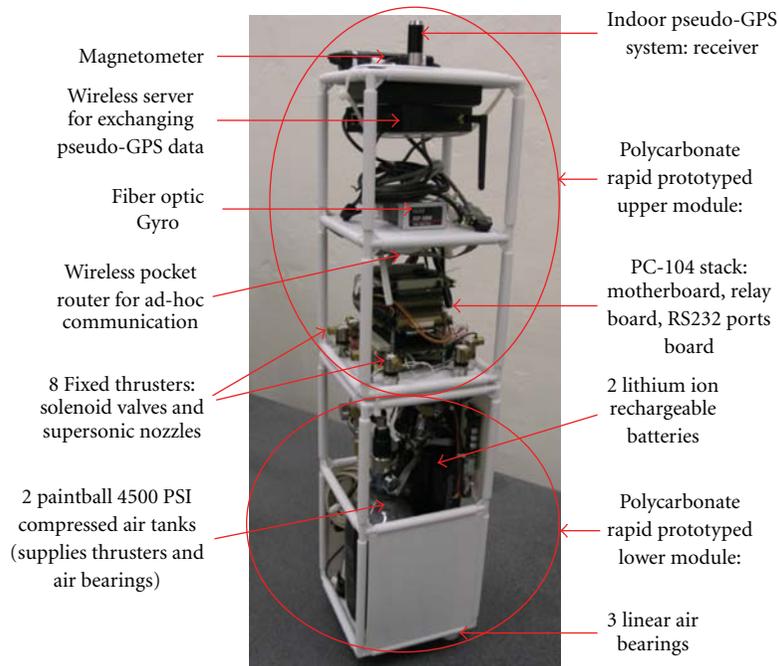


FIGURE 4: Detailed collocation of the hardware on the spacecraft simulators.

The onboard real-time operating system is RTAI patched Linux (see [24]), in the Debian 2.6.19 flavor. The classical use of xPC Target by MathWorks as a real-time operating system (OS) is common in academic research (see [25]). A key advantage of xPC Target is its seamless integration between

Simulink via Real-Time Workshop which allows for rapid prototyping of navigation and control algorithms for real-time requirements. Real-Time Workshop automatically generates C code from a Simulink model and the corresponding executable file for an xPC Target-based computer. On the

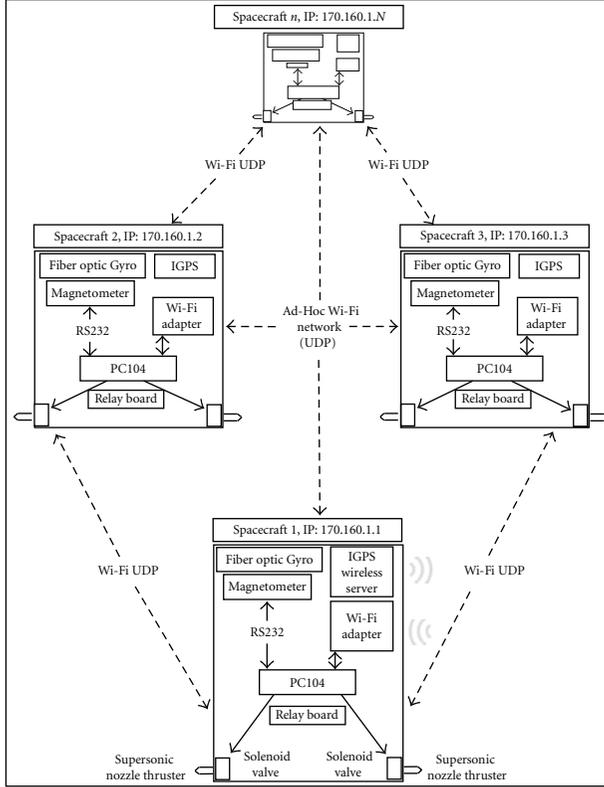


FIGURE 5: Ad-hoc wireless network at the SRL test-bed.

other hand, xPC Target has some disadvantages that include support for a limited number of hardware components and no support for USB or Firewire devices. Furthermore, the inaccessibility of its source code, due to its proprietary commercial nature, makes it challenging to add or modify drivers for unsupported hardware.

RTAI Linux has been successfully used as an onboard real-time OS. RTAI is a patch to the Linux kernel that allows for the execution of real-time tasks in Linux (see [26, 27]). The RTAI Linux solution is being widely exploited in several engineering areas (see [28–31]). In this work, we use RTAI Linux with a wide variety of hardware interfaces to include wireless ad-hoc radio communication using UDP, RS232 interface with the sensor suite and power system and a PC/104 relay board for actuating compressed air nozzles. RTAI Linux also allows for automatic generation of C code from Simulink models through Real-Time Workshop with the executable file for the onboard computers being created outside MATLAB by simple compilation of the C code.

The details on the ad-hoc wireless network and hardware-software interfaces developed for the Spacecraft Simulators are available in [6].

### 3. S/C Relative Motion Dynamics and Problem Statement

In this section, we provide the dynamics of spacecraft relative motion in the three degrees of freedom case. The dynamics

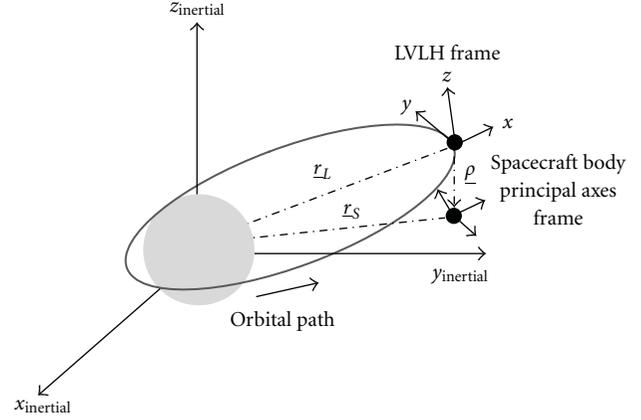


FIGURE 6: Local vertical local horizontal and inertial frames.

encompasses both the relative translation (two degrees of freedom) and rotation (one degree of freedom). We will refer in the following to a Local Vertical Local Horizontal (LVLH) reference frame (Figure 6) that rotates with the orbital angular velocity  $\omega_{LVLH}$ . The origin of LVLH moves on a virtual orbit, conveniently chosen to remain in the vicinity of the maneuvering spacecraft. This point can also be chosen as coincident with one of the spacecraft. The  $x$ -axis points from the center of the Earth to the center of LVLH, while the  $y$ -axis is in the orbital plane in the direction of the motion along the orbit and perpendicular to the  $x$ -axis. The  $z$ -axis completes the right-handed LVLH frame.

The dynamics of such motion can be represented in the compact form as

$$\dot{\mathbf{X}}_{\text{rel}} = f(\mathbf{X}_{\text{rel}}) + B(\mathbf{X}_{\text{rel}})\mathbf{u}. \quad (1)$$

From now on, we will consider the specific application of hardware-in-the-loop testing using the three-degree-of-freedom spacecraft simulators at the Naval Postgraduate School. For the experimental setup, the state vector becomes

$$\begin{aligned} \mathbf{X}_{\text{rel}} &= [x \ y \ \theta \ \dot{x} \ \dot{y} \ \omega]^T \\ &= [x_T - x_C \ y_T - y_C \ \theta_T - \theta_C \ \dot{x}_T - \dot{x}_C \ \dot{y}_T - \dot{y}_C \ \omega_{zT} - \omega_{zC}]^T \\ \mathbf{u} &= [u_{xT} - u_{xC} \ u_{yT} - u_{yC} \ M_T - M_C]^T. \end{aligned} \quad (2)$$

Being  $u_{ij}$ ,  $i = x, y, j = C, T$  the control forces components of chaser and target, and  $M_j$ ,  $j = C, T$  the control torque of chaser and target about the  $z$  axis.

It is common use in the literature to linearize the relative motion dynamics and use the Clohessy and Wiltshire linear equation [32]

$$\begin{aligned} \ddot{x} - 2\omega_{LVLH}\dot{y} - 3\omega_{LVLH}^2x &= \frac{1}{m}(u_{xT} - u_{xC}), \\ \ddot{y} - 2\omega_{LVLH}\dot{x} &= \frac{1}{m}(u_{yT} - u_{yC}), \end{aligned} \quad (3)$$

with the assumption that the spacecraft have the same mass  $m$ .

For maneuvers confined in the vicinity of the LVLH origin, elapsing a short time in comparison to the orbital period (3) can be further simplified into a double integrator for both  $x$  and  $y$ . A double integrator dynamics also represents the dynamics of the spacecraft simulators in the laboratory inertial reference frame. For the above-mentioned reasons, (4) will be used for the remaining of the paper;

$$\begin{aligned}\ddot{x} &= \frac{1}{m}(u_{xT} - u_{xC}), \\ \ddot{y} &= \frac{1}{m}(u_{yT} - u_{yC}).\end{aligned}\quad (4)$$

Assuming the spacecraft having the same moment of inertia about the  $z$  axis, the attitude dynamics is also represented by a double integrator

$$\ddot{\theta} = \ddot{\theta}_T - \ddot{\theta}_C = \dot{\omega}_{zT} - \dot{\omega}_{zC} = \frac{1}{J_z}(M_T - M_C). \quad (5)$$

The goal of this work is to develop a GNC approach for driving the state  $\mathbf{X}_{\text{rel}}$  to perform assembly maneuvers. This requires accurate guidance, especially in the last phases of docking, optimized or suboptimized control, to minimize propellant consumption, and a robust relative navigation scheme. These requirements are addressed in the following sections.

#### 4. Relative Navigation: The Augmented State Estimation Method

In this section, the theory for the three-degree-of-freedom augmented state relative navigation is presented. The controls of another vehicle (target) are treated as additional terms in the corresponding state equation, so that the model provides an augmented state vector. The measurements available on each spacecraft are relative positions (from range and line of sight) and relative attitude, and we assume the knowledge of the controls of the chaser, onboard the chaser itself. No relative velocities measurements are available. Observability proof of the vector  $[x \ y \ \theta \ \dot{x} \ \dot{y} \ \omega \ u_{xT} \ u_{yT} \ M_{zT}]^T$  from the measurements  $[x \ y \ \theta]^T$  is provided for the proposed estimation technique, demonstrating how the augmented state technique can reconstruct relative velocities and controls of the target.

In the following developments, the estimated target's controls are considered constants within every sample time interval. It is worth underlying that the control variables  $u_{xT}$ ,  $u_{yT}$ , and  $M_{zT}$  do not represent the actual actuators' control variables onboard the spacecraft simulator. The way  $u_{xT}$ ,  $u_{yT}$ , and  $M_{zT}$  are obtained from the target does not matter from the augmented state filter's point of view. These control variables are estimated in order to add robustness to the filtering technique; they represent the target's maneuvers, but not the specific way they are performed by the target's actuation subsystem.

The same assumption will be used for the observability demonstration. The navigation algorithm is developed using the Kalman filter approach.

The augmented state estimation approach presents numerical efficiency comparable to the standard Kalman Filter applied on the state only. In fact, the augmented state approach introduces a few more variables in the Kalman Filter, without a significant increase on the numerical burden. Additional references with regards to the implementation of Extended Kalman Filters onboard real space missions can be found in [33, 34].

**4.1. Relative Motion Estimation.** The assumption is made of independent estimation and control for the attitude and the position, so that we can proceed as follows. For the relative position, the state vector can be written as (see (2))

$$\mathbf{X}_{T_i}^{C_j} = \mathbf{X}_{T_i}^{C_j} [x \ y \ \dot{x} \ \dot{y}]^T. \quad (6)$$

The discrete dynamics for the problem is the following:

$$\begin{aligned}\mathbf{X}_{T_i}^{C_j}(k+1) &= \mathbf{X}_{T_i}^{C_j}\Psi(k) \mathbf{X}_{T_i}^{C_j}(k) \\ &+ \mathbf{X}_{T_i}^{C_j}\mathbf{B}(k)(\mathbf{u}_T(k) - \mathbf{u}_C(k)) + \mathbf{G}\mathbf{W}(k), \\ \mathbf{Z}(k) &= \mathbf{H}_{T_i}^{C_j}\mathbf{X}(k) + \mathbf{V}(k).\end{aligned}\quad (7)$$

The expressions of the matrices:  $G$ ,  $H$ ,  $\mathbf{X}_{T_i}^{C_j}\mathbf{B}(k)$ , and  $\Psi$  as functions of the measurement update time  $T_s$  for this planar case can be written as

$$\begin{aligned}\mathbf{G} &= \mathbf{I}_{4 \times 4}, \quad \mathbf{H} = \begin{pmatrix} \mathbf{I}_{2 \times 2} & \mathbf{0}_{2 \times 2} \end{pmatrix}, \\ \mathbf{X}_{T_i}^{C_j}\mathbf{B}(k) &= \int_0^{T_s} \left( \mathbf{X}_{T_i}^{C_j}\Phi(t) \cdot \mathbf{X}_{T_i}^{C_j}\mathbf{B} \right) dt = \begin{pmatrix} \frac{T_s^2}{2} & 0 \\ 0 & \frac{T_s^2}{2} \\ T_s & 0 \\ 0 & T_s \end{pmatrix} / m, \\ \mathbf{X}_{T_i}^{C_j}\Psi(k) &= \mathbf{X}_{T_i}^{C_j}\Phi(T_s) = e^{\mathbf{X}_{T_i}^{C_j}\mathbf{F}T_s} \\ &= \mathbf{I}_{4 \times 4} + \mathbf{X}_{T_i}^{C_j}\mathbf{F}T_s = \begin{pmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},\end{aligned}\quad (8)$$

being

$$\mathbf{X}_{T_i}^{C_j}\mathbf{F} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{X}_{T_i}^{C_j}\mathbf{B} = \begin{pmatrix} \mathbf{0}_{2 \times 2} \\ \mathbf{I}_{2 \times 2} \end{pmatrix} / m. \quad (9)$$

The augmented dynamics adds the estimation of  $u_{xT}$  and  $u_{yT}$ , assuming knowledge of the chaser's controls  $u_{xc}$  and  $u_{yc}$ . The related state equation matrices can be written as

$$\begin{aligned} \frac{C_j}{T_i} \mathbf{X}_A &= \begin{bmatrix} \frac{C_j}{T_i} \mathbf{X}(K+1) \\ \mathbf{u}_T(K+1) \end{bmatrix} \\ &= \frac{C_j}{T_i} \Psi_A(k) \begin{bmatrix} \frac{C_j}{T_i} \mathbf{X}(K) \\ \mathbf{u}_T(K) \end{bmatrix} + \mathbf{G}_A \mathbf{W}(k) + \frac{C_j}{T_i} \mathbf{B}_C(k) \mathbf{u}_C(k), \end{aligned}$$

$$\mathbf{Z}_A(K) = \mathbf{H}_A \begin{bmatrix} \frac{C_j}{T_i} \mathbf{X}(K) \\ \mathbf{u}_T(K) \end{bmatrix} + \mathbf{V}(K),$$

$$\mathbf{G}_A = \begin{pmatrix} \mathbf{G} \\ \mathbf{0}_{2 \times 4} \end{pmatrix}, \quad \mathbf{H}_A = (\mathbf{I}_{2 \times 2} \quad \mathbf{0}_{2 \times 4}), \quad \mathbf{u}_C = \begin{bmatrix} u_{Cx} \\ u_{Cy} \end{bmatrix},$$

$$\frac{C_j}{T_i} \Psi_A(k) = \begin{pmatrix} \frac{C_j}{T_i} \Psi(k) & \frac{C_j}{T_i} \mathbf{B}(k) \\ \mathbf{0}_{2 \times 4} & \mathbf{I}_{2 \times 2} \end{pmatrix} = \begin{pmatrix} 1 & 0 & T_s & 0 & \frac{T_s^2}{2m} & 0 \\ 0 & 1 & 0 & T_s & 0 & \frac{T_s^2}{2m} \\ 0 & 0 & 1 & 0 & \frac{T_s}{m} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{T_s}{m} \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\frac{C_j}{T_i} \mathbf{B}_C(k) = \begin{pmatrix} -\frac{C_j}{T_i} \mathbf{B}(k) \\ \mathbf{0}_{2 \times 2} \end{pmatrix} = \begin{pmatrix} -\frac{T_s^2}{2} & 0 \\ 0 & -\frac{T_s^2}{2} \\ -T_s & 0 \\ 0 & -T_s \\ 0 & 0 \\ 0 & 0 \end{pmatrix} / m. \quad (10)$$

**4.2. Relative Attitude Estimation.** The same algorithm is implemented for target's attitude and control torque estimation. Assuming the target rotating only about the vertical axis ( $z$ -axis), the  $i$ th Target attitude state vector, with respect to the  $j$ th Chaser spacecraft, is chosen to be

$$\frac{C_j}{T_i} \Theta = \frac{C_j}{T_i} [\theta \quad \dot{\theta}]^T. \quad (11)$$

The discrete dynamics for the attitude problem is

$$\begin{aligned} \frac{C_j}{T_i} \Theta(k+1) &= \frac{C_j}{T_i} \Psi(k) \frac{C_j}{T_i} \Theta(k) \\ &+ \frac{C_j}{T_i} \mathbf{B}(k) (M_T(k) - M_C(k)) + \mathbf{G} \mathbf{W}(k), \end{aligned} \quad (12)$$

$$\mathbf{Z}(k) = \mathbf{H} \frac{C_j}{T_i} \Theta(k) + \mathbf{V}(k),$$

and the principal dynamics matrices, as function of the time sampling  $T_s$ , are

$$\mathbf{G} = \mathbf{I}_{2 \times 2}, \quad \mathbf{H} = (1 \quad 0),$$

$$\frac{C_j}{T_i} \mathbf{B}(k) = \int_0^{T_s} \left( \frac{C_j}{T_i} \Phi(t) \cdot \frac{C_j}{T_i} \mathbf{B} \right) dt = \begin{pmatrix} \frac{T_s^2}{2J_z} \\ \frac{T_s}{J_z} \end{pmatrix},$$

$$\frac{C_j}{T_i} \Psi(k) = \frac{C_j}{T_i} \Phi(T_s) = e^{C_j \mathbf{F} T_s} = \mathbf{I}_{2 \times 2} + \frac{C_j}{T_i} \mathbf{F} T_s = \begin{pmatrix} 1 & T_s \\ 0 & 1 \end{pmatrix}, \quad (13)$$

being

$$\frac{C_j}{T_i} \mathbf{F} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad \frac{C_j}{T_i} \mathbf{B} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} / J_z. \quad (14)$$

The formulation of the augmentation of the state dynamics adds the estimation of  $M_T$ , assuming the knowledge of the chaser's control torque  $M_C$ . The related state equation matrices can be written as

$$\begin{aligned} \frac{C_j}{T_i} \Theta_A(k) &= \begin{bmatrix} \frac{C_j}{T_i} \Theta(k+1) \\ \mathbf{M}_T(k+1) \end{bmatrix} \\ &= \frac{C_j}{T_i} \Psi_A(k) \begin{bmatrix} \frac{C_j}{T_i} \Theta(k) \\ \mathbf{M}_T(k) \end{bmatrix} + \frac{C_j}{T_i} \mathbf{B}_C(k) M_C + \mathbf{G}_A \mathbf{W}(k), \end{aligned}$$

$$\mathbf{Z}_A(k) = \mathbf{H}_A \begin{bmatrix} \frac{C_j}{T_i} \Theta(k) \\ \mathbf{M}_T(k) \end{bmatrix} + \mathbf{V}(k),$$

$$\mathbf{G}_A = \begin{pmatrix} \mathbf{G} \\ \mathbf{0}_{2 \times 2} \end{pmatrix}, \quad \mathbf{H}_A = (1 \quad 0 \quad 0), \quad M_C = [M_z],$$

$$\frac{C_j}{T_i} \mathbf{B}_C(k) = \begin{pmatrix} -\frac{C_j}{T_i} \mathbf{B}(k) \\ 0 \end{pmatrix} = \begin{pmatrix} -\frac{T_s^2}{2} \\ -T_s \\ 0 \end{pmatrix} / J_z,$$

$$\frac{C_j}{T_i} \Psi_A(k) = \begin{pmatrix} \frac{C_j}{T_i} \Psi(k) & \frac{C_j}{T_i} \mathbf{B}(k) \\ \mathbf{0}_{1 \times 2} & 1 \end{pmatrix} = \begin{pmatrix} 1 & T_s & \frac{T_s^2}{2J_z} \\ 0 & 1 & \frac{T_s}{J_z} \\ 0 & 0 & 1 \end{pmatrix}. \quad (15)$$

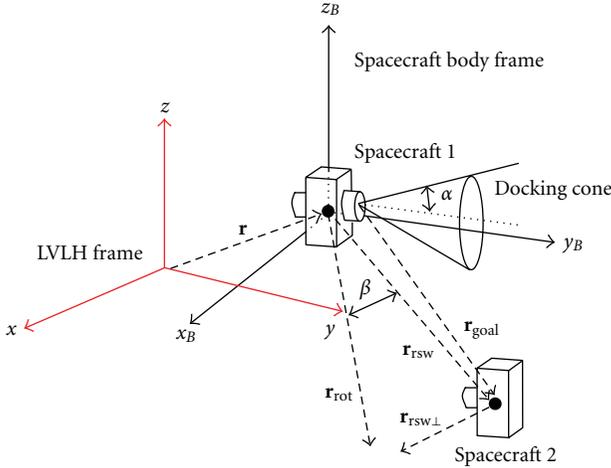


FIGURE 7: Relative vectors used in the alignment and assembly logic. All vectors are in the LVLH  $xy$  plane.

**4.3. Observability of the Augmented Dynamics.** For sake of simplicity, considering that the controls are constant in each sample time, we provide, for the planar case, the proof of the observability for the continuous models of the relative dynamics. The observability property holds for the discrete models [35]. The augmented relative motion dynamics can be expressed as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \\ \dot{u}_{xT} \\ \dot{u}_{yT} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{m} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{m} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \\ u_{xT} \\ u_{yT} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -\frac{1}{m} & 0 \\ 0 & -\frac{1}{m} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_{xC} \\ u_{yC} \end{bmatrix}. \quad (16)$$

(A) (B)

The measurements are related to the state as follows:

$$Z = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \\ u_{xT} \\ u_{yT} \end{bmatrix}. \quad (17)$$

(C)

It is of immediate demonstration that the following observability matrix has full rank:

$$O = \begin{bmatrix} C \\ CA \\ \cdot \\ \cdot \\ CA^5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{m} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{m} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (18)$$

Similar developments lead to observability for the relative attitude motion. The dynamics can be expressed as

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{M}_{zT} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & \frac{1}{J_z} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ M_{zT} \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{1}{J_z} \\ 0 \end{bmatrix} M_{zC}. \quad (19)$$

(A) (B)

The measurements are related to the state as follows:

$$Z = \begin{bmatrix} 1 & 0 & 0 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ M_{zT} \end{bmatrix}. \quad (20)$$

(C)

The observability matrix is as following:

$$O = \begin{bmatrix} C \\ CA \\ CA^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{J_z} \end{bmatrix}, \quad (21)$$

that has full rank.

## 5. Guidance and Control for the Assembly Maneuver

This section describes guidance and control for the autonomous assembly.

**5.1. Guidance.** Figure 7 shows the principal vectors used by the guidance algorithm. It is worth to underline that Figure 7

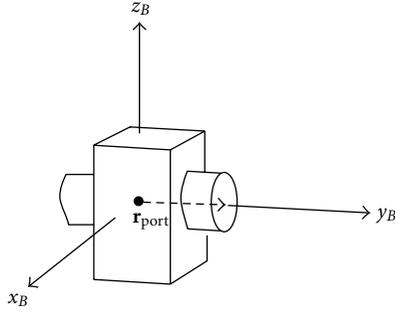


FIGURE 8: Body fixed docking port vector.

only represents one possible configuration and that the docking ports do not have to be aligned with any particular body axis.  $\mathbf{r}_{\text{goal}}$  is the vector originating from the center of a docking interface, terminating at the center of mass of the other spacecraft going to dock to it.  $\mathbf{r}_{\text{rsw}}$  is the vector originating from a spacecraft's center of mass, terminating at the other spacecraft's center of mass.

The guidance problem is here expressed in terms of desired state vector for each spacecraft, defined dynamically during the maneuver. The state vector error to minimize is

$$\mathbf{x}_{\text{err}} = \begin{bmatrix} x - x_{\text{des}} & y - y_{\text{des}} & \theta - \theta_{\text{des}} & \dot{x} - \dot{x}_{\text{des}} & \dot{y} - \dot{y}_{\text{des}} & \omega - \omega_{\text{des}} \end{bmatrix}^T. \quad (22)$$

The subscript “des” indicates a desired relative state vector component. The desired state is dynamically changed throughout the assembly maneuver according to the following two-phase guidance logic.

The center of mass trajectory is unconstrained, free to be optimized, unless in the vicinity of the docking phase. As for the attitude, we reproduce a realistic condition in which the spacecraft has to show one particular side (usually the one with the docking port) towards the current target spacecraft. In other words, the docking port side is commanded to be perpendicular to either the  $\mathbf{r}_{\text{rsw}}$  or the  $\mathbf{r}_{\text{goal}}$  vector (Figure 7), depending on the phase. Each spacecraft in Figure 7 can be considered either a single agent or an already assembled structure; the following, description applies to both scenarios. In the following the vectors are always intended to be parallel to the  $xy$  plane.  $r_{\text{dock}}$  is a user defined distance threshold, specifying when the docking phase begins.

(1)  $|\mathbf{r}_{\text{rsw}}| > r_{\text{dock}}$ , RENDEZVOUS: the spacecraft is at a far away distance from its target docking port. The state vector error is  $\mathbf{x}_{\text{err}} = [\mathbf{r}_{\text{goal}_x} \ \mathbf{r}_{\text{goal}_y} \ \theta - \theta_{\text{des}} \ \dot{x} \ \dot{y} \ \omega]^T$ . The desired attitude  $\theta_{\text{des}}$  is such to align  $\mathbf{r}_{\text{port}}$  to  $\mathbf{r}_{\text{rsw}}$  (Figures 7 and 8).

(2)  $|\mathbf{r}_{\text{rsw}}| \leq r_{\text{dock}}$ , DOCKING APPROACH: the spacecraft is close to its target docking port. The desired state vector to minimize is:

- (a) If  $\cos^{-1}((\mathbf{r}_{\text{goal}} \cdot \mathbf{r}_{\text{port}})/|\mathbf{r}_{\text{goal}}||\mathbf{r}_{\text{port}}|) < \alpha$ , that is, the spacecraft is within the security docking cone, there are two subcases.

SUBCASE 1. The distance between the spacecraft is greater than the chosen impingement stand-off range, then  $\mathbf{x}_{\text{err}} = [\mathbf{r}_{\text{goal}_x} \ \mathbf{r}_{\text{goal}_y} \ \theta - \theta_{\text{des}} \ \dot{x} \ \dot{y} \ \omega]^T$ . The desired attitude  $\theta_{\text{des}}$  is such to align  $\mathbf{r}_{\text{port}}$  to  $\mathbf{r}_{\text{rsw}}$  (Figures 7 and 8).

SUBCASE 2. The distance between the spacecraft is less than the chosen impingement stand-off range, then any thrusters causing plume impingement on the other spacecraft are shut off, and only used if an emergency brake is needed, in the event of docking occurring at high velocity (above a chosen threshold). For the NPS spacecraft simulators, this will mean shutting off two thrusters, as it will be clear later on. The remaining actuators will compensate for attitude alignment in the last phase of docking and will provide required forces to push the spacecraft together.

- (b) If  $\cos^{-1}((\mathbf{r}_{\text{goal}} \cdot \mathbf{r}_{\text{port}})/|\mathbf{r}_{\text{goal}}||\mathbf{r}_{\text{port}}|) \geq \alpha$ , that is, the spacecraft is outside the security docking cone. In this case, referring to spacecraft 2 of Figure 7, the vehicle maneuvers orbiting around the one hosting its target docking port, moving along the direction perpendicular to the  $\mathbf{r}_{\text{rsw}}$  vector, towards the way that is the shortest in order to reach the safety corridor. The amount of commanded rotation at each time step, around the target docking port, is a chosen parameter  $\beta = \text{const}$ . In terms of state vector error to minimize, defining a reference frame which has as a basis the unit vectors  $\hat{\mathbf{r}}_{\text{rsw}\perp}$ ,  $\hat{\mathbf{r}}_{\text{rsw}}$ , the  $\mathbf{r}_{\text{rsw}}$  can be rotated of an angle  $\beta$  into  $\mathbf{r}_{\text{rot}}$  and easily expressed as function of the basis  $\mathbf{r}_{\text{rot}} = |\mathbf{r}_{\text{rsw}}|(\cos\beta\hat{\mathbf{r}}_{\text{rsw}} + \sin\beta\hat{\mathbf{r}}_{\text{rsw}\perp})$ . The state error to minimize is  $\mathbf{x}_{\text{err}} = [\mathbf{r}_{\text{rot}_x} - \mathbf{r}_{\text{goal}_x} \ \mathbf{r}_{\text{rot}_y} - \mathbf{r}_{\text{goal}_y} \ \theta - \theta_{\text{des}} \ \dot{x} \ \dot{y} \ \omega]^T$ . The desired attitude  $\theta_{\text{des}}$  is such to show the chosen side to the target docking spacecraft, that is,  $\perp \mathbf{r}_{\text{goal}}$  (Figure 7). In simple terms, the satellites circle around each other, in the direction of shortest angular displacement, to allow the docking interfaces to be in the mutual fields of view. Each spacecraft needs to be in the safety corridor one of the other, with the respective docking interfaces'  $\mathbf{r}_{\text{port}}$  vectors and  $\mathbf{r}_{\text{goal}}$  vectors aligned. The respective  $\mathbf{r}_{\text{port}}$  vectors of two satellites will need to be at 180 degrees (plus-minus the tolerance); the same applies, as a consequence, to the  $\mathbf{r}_{\text{goal}}$  vectors.

5.2. LQR Control. The LQR problem (23) is solved at each time step, with dynamically sized weighting matrices  $\mathbf{Q}$  and  $\mathbf{R}$ , adapting to the current situation, avoiding high control values when the state vector error is relevant, and vice versa. This choice results in a smoother behavior, in terms of requested control actions, with respect to classical fixed gain matrices LQR;

$$J = \int_0^{\infty} (\mathbf{x}_{\text{err}}^T \mathbf{Q} \mathbf{x}_{\text{err}} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt. \quad (23)$$

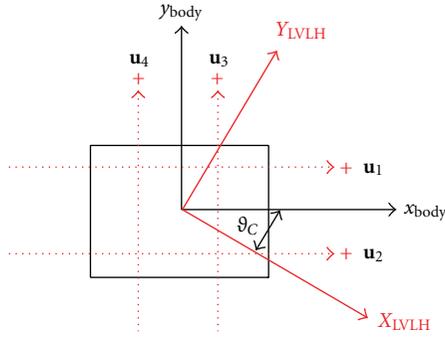


FIGURE 9: Locations of controls for the planar assembly.

TABLE 2: Main simulation parameters.

Mass of each simulator	10,5	kg
Inertia of simulator	0.063	kg m <sup>2</sup>
Inertia of the two simulators assembled	0.18	kg m <sup>2</sup>
Single thruster estimated force [36]	0.16	N
Docking cone semiaperture	0.75	degrees
Force arms (for torque generation)	5, 10, 21	cm
Limit distance for switching off the thrusters	0.7	m
iGPS accuracy	1	mm
Gyroscope accuracy	0.003	deg/sec
Thrusters minimum actuation time	$1.5 \cdot 10^{-3}$	sec

The cost function in (23) aims to minimize control effort, while reducing the relative state vector between two satellites. The mutual relevance between state vector error and control effort is dictated by the relative values of the weighting matrices  $\mathbf{Q}$  and  $\mathbf{R}$ .

The control vector  $\mathbf{u}$  is chosen as a four-component vector of forces, expressed in the spacecraft body frame (Figure 9). The choice of  $\mathbf{u}$  in the spacecraft body frame, removes the need for thruster mapping [22].

For the phases described in the previous section the weighting matrices for the LQR are chosen as

$$\mathbf{Q} = \begin{bmatrix} \frac{1}{|\mathbf{r}_{\text{goal}}|} \cdot \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & |\mathbf{r}_{\text{goal}}|^3 \cdot V \cdot \mathbf{I}_3 \end{bmatrix}, \quad (24)$$

$$\mathbf{R} = \frac{|\mathbf{r}_{\text{goal}}|}{a^2} \mathbf{I}_4.$$

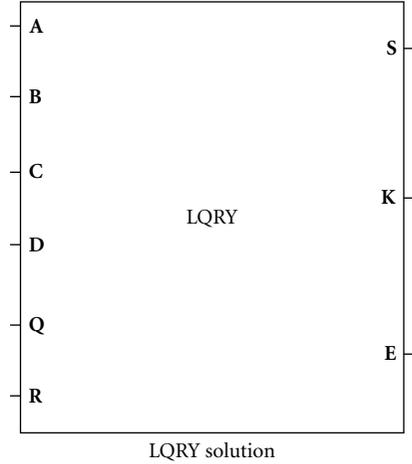


FIGURE 10: LQR Solver Simulink Block [18]. This routine solves the complete algebraic Riccati equation accepting the input matrices:  $\mathbf{A}$  (dynamics matrix),  $\mathbf{B}$  (control matrix),  $\mathbf{C}$  (state-output mapping matrix),  $\mathbf{D}$  (control-output mapping matrix), and  $\mathbf{Q}$  and  $\mathbf{R}$  weighting matrices. The outputs are the LQR gain matrix,  $\mathbf{K}$ , which is the solution to the associated algebraic Riccati equation, the matrix  $\mathbf{S}$ , and a two-dimensional vector,  $\mathbf{E}$ , whose first element indicates an error when it is greater than zero or a somewhat unreliable result when it is negative. The second element of  $\mathbf{E}$  is the condition number of the  $\mathbf{R}$  matrix.

Each time step solution of the LQR generates a gain matrix  $\mathbf{K}_{\text{LQR}}$ , used to implement the required suboptimal control vector

$$\mathbf{u}_{\text{LQR}} = -\mathbf{K}_{\text{LQR}} \cdot \mathbf{x}_{\text{err}}. \quad (25)$$

The values of the constants  $a = 3.05 \cdot 10^{-2}$  and  $V = 0.06$  are chosen as in [23]. In particular, their values are chosen looking at variables with physical meaning, but we do not assign dimensions to them here, being their dimensions the appropriate ones for consistency in the cost function (23). The  $a$  constant weighs the terms in the matrix  $\mathbf{R}$  with respect to the maximum translational acceleration achievable on the spacecraft simulator. This value is computed considering two thrusters simultaneously activated on the same side of the vehicle. The thrust values and mass of the simulator can be found in Table 2, and the interested reader can find more details on the thrusters in [36]. The originating idea for scaling the  $\mathbf{R}$  matrix as in (24) is the desire to maintain the controls required by the LQR solution below the maximum hardware-achievable controls. With regard to the constant  $V$  in the  $\mathbf{Q}$  matrix, it is set to be the maximum translation speed allowed for the simulator.  $V$  weighs the part of the state vector which is related to linear velocities. The choice to introduce the above mentioned parameters does not a priori guarantee controls below the maximum onboard control authority and a limited translational velocity, but the scaling in (24) has been proven very effective in mitigating high requests on the control and undesired fast maneuvers on the testbed. This result was previously found in numerical simulations, and then experimentally verified [23].

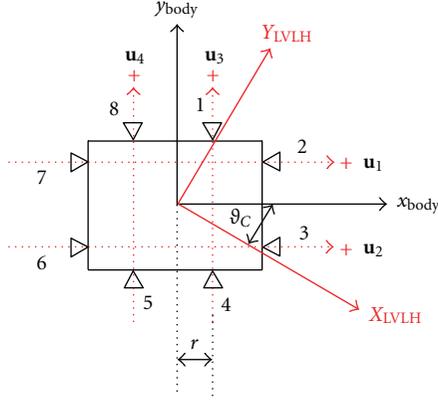


FIGURE 11: Thruster coupling on the spacecraft simulators.

Figure 10 shows the required inputs to the LQR solver, implemented in Simulink.

The LQR solver employed for developing the proposed approach was downloaded from [18], adapted for automatic generation of code through Real-Time-Workshop for RTAI Linux (it was originally only compatible with Windows Operating Systems), and uploaded again on the MathWorks file exchange website [18].

In specializing the design of Figure 9 to the SRL spacecraft simulators, we treat the eight body fixed thrusters in couples, so that symmetric thrusters are reduced to one control variable, which can be either  $u_{\max}$ ,  $-u_{\max}$ , 0. Figure 11 shows the thruster couplings: 1–4, 2–7, 3–6, and 5–8. The control vector is  $\mathbf{u} = [u_1 \ u_2 \ u_3 \ u_4]^T$ . The red arrows along the couples in Figure 11 show the positive directions assumed for the controls. Ultimately, thrusters

coupling allows the LQR to solve a reduced problem in which the control vector has four components instead of eight.

Given the choice for the control vector, the control distribution matrix becomes nonlinear, as in (25). Equation (25) also shows the system dynamics matrices, when the expression  $\dot{\mathbf{X}} = \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{U}$ ,  $\mathbf{Y} = \mathbf{C}\mathbf{X} + \mathbf{D}\mathbf{U}$  is used. The spacecraft orientation  $\theta_C$  is replaced with  $\theta$  for simplicity

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\mathbf{B} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{\cos(\theta)}{m} & \frac{\cos(\theta)}{m} & -\frac{\sin(\theta)}{m} & -\frac{\sin(\theta)}{m} \\ \frac{\sin(\theta)}{m} & \frac{\sin(\theta)}{m} & \frac{\cos(\theta)}{m} & \frac{\cos(\theta)}{m} \\ -\frac{r}{J_z} & \frac{r}{J_z} & \frac{r}{J_z} & -\frac{r}{J_z} \end{pmatrix}, \quad (26)$$

$$\mathbf{C} = \mathbf{I}_{6 \times 6}, \quad \mathbf{D} = \mathbf{0}_{6 \times 4}.$$

In order to employ the LQR approach, the control distribution matrix is linearized at each time step, in the vicinity of the desired attitude.

$$\mathbf{B}_{\text{LIN}} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{\cos(\theta_{\text{des}}) - \sin(\theta_{\text{des}})(\theta - \theta_{\text{des}})}{m} & \frac{\cos(\vartheta_{\text{des}}) - \sin(\theta_{\text{des}})(\theta - \theta_{\text{des}})}{m} & -\frac{\sin(\vartheta_{\text{des}}) + (\cos(\theta_{\text{des}}))(\theta - \theta_{\text{des}})}{m} & -\frac{\sin(\theta_{\text{des}}) + (\cos(\theta_{\text{des}}))(\theta - \theta_{\text{des}})}{m} \\ \frac{\sin(\theta_{\text{des}}) + (\cos(\theta_{\text{des}}))(\theta - \theta_{\text{des}})}{m} & \frac{\sin(\theta_{\text{des}}) + (\cos(\theta_{\text{des}}))(\theta - \theta_{\text{des}})}{m} & \frac{\cos(\vartheta_{\text{des}}) - \sin(\theta_{\text{des}})(\theta - \theta_{\text{des}})}{m} & \frac{\cos(\theta_{\text{des}}) - \sin(\theta_{\text{des}})(\theta - \theta_{\text{des}})}{m} \\ -\frac{r}{J_z} & \frac{r}{J_z} & \frac{r}{J_z} & -\frac{r}{J_z} \end{pmatrix} \quad (27)$$

By inserting the matrices defined in (26) and (27) and the weighting matrices described in (24) into the LQR solver of Figure 10, the optimal four components vector of forces is obtained, at each time step during the maneuver. The obtained control vector will be a continuous signal. In order to drive the on/off thrusters from the continuous signal Pulse Width Modulation is used; the PWM collects commanded controls over 10 sample times before actuating. Furthermore, a Schmitt Trigger is implemented, to filter out low commanded controls and reduce the amount of chattering.

## 6. Navigation and Control of the Assembled Structure

Once the S/Cs are assembled, the mass and inertia properties along with the thrusters configuration change. Figure 12 shows an example, applied to the SRL testbed, in which thrusters six and seven of both spacecraft cannot be used anymore. The assembled new spacecraft has doubled mass, different moment of inertia and four more thrusters, differently allocated with respect to the single spacecraft. In assembled configuration, one of the robots acts as master

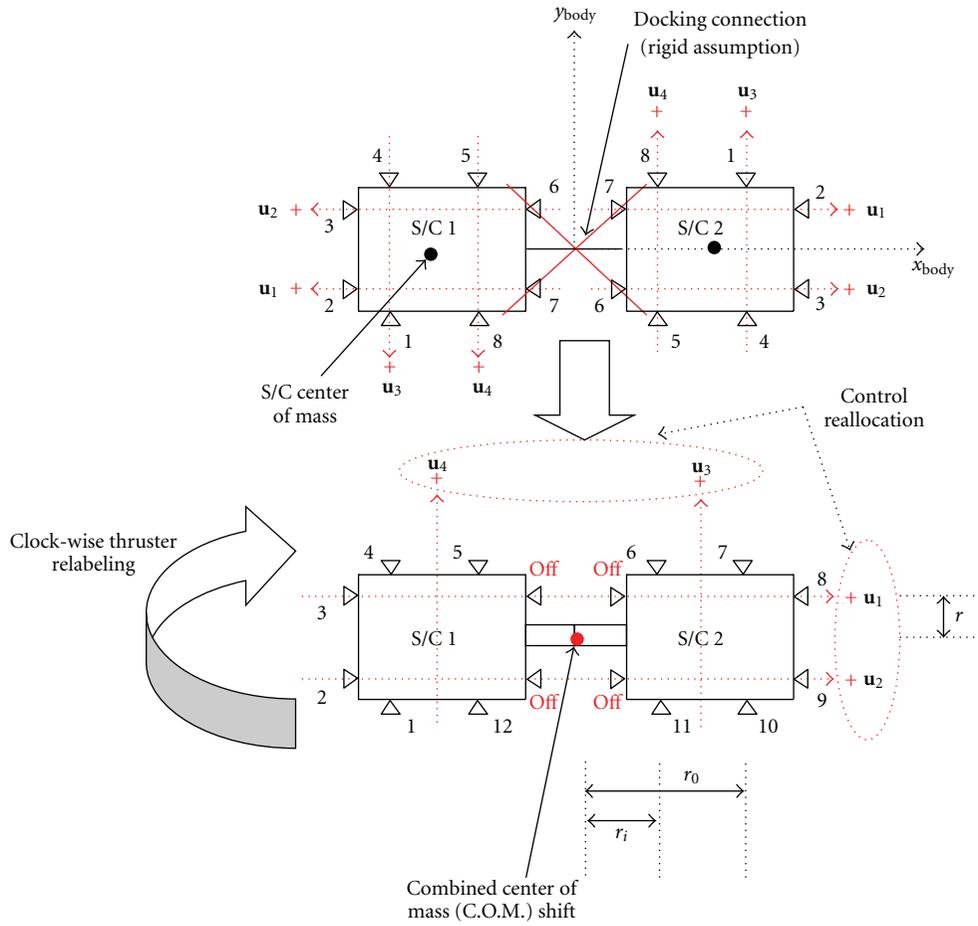


FIGURE 12: Assembled configurations with reallocated thruster coupling and COM shift.

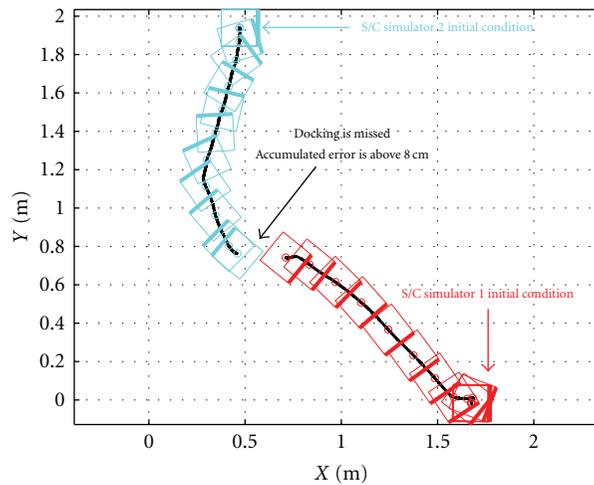


FIGURE 13: Experimental Result: bird's eye view for two spacecraft simulator failed assembly maneuver. The relative navigation is performed via classical Kalman Filtering, no Augmented State Estimation. The bolded lines are employed to help visualize the simulator's orientation.

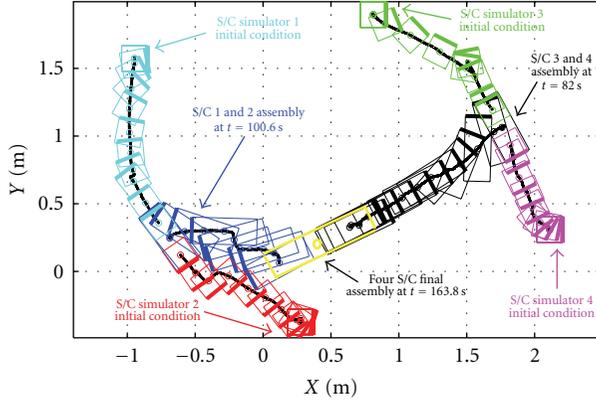


FIGURE 14: Experimental Result: bird's eye view for four spacecraft simulator assembly maneuver. The relative navigation is performed via augmented state estimation. The bolded lines are employed to help visualize the simulator's orientation.

and performs both navigation and control of the new bigger robot. In order to keep using the same logic employed for controlling a single simulator, the twelve thrusters of the new assembled spacecraft are associated according to the following sets:

- (1)  $u_1$  is generated by firing either thruster 8 ( $u_1 < 0$ ) or 3 ( $u_1 > 0$ ),
- (2)  $u_2$  is generated by firing either thruster 9 ( $u_2 < 0$ ) or 2 ( $u_2 > 0$ ),
- (3)  $u_3$  is generated by firing either thrusters 6 and 7 synchronously ( $u_3 < 0$ ) or 11 and 10 synchronously ( $u_3 > 0$ ),
- (4)  $u_4$  is generated by firing either thrusters 4 and 5 synchronously ( $u_4 < 0$ ) or 1 and 12 synchronously ( $u_4 > 0$ ).

The input matrices to the LQR solver will be changed once an additional portion of the structure is connected.

Also, the new control vector will have maximum and minimum values reduced, due to the increase of mass. For instance, the case represented in Figure 12 leads to the new matrices

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\mathbf{B} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{\cos(\theta)}{m_{\text{comb}}} & \frac{\cos(\theta)}{m_{\text{comb}}} & \frac{-\sin(\theta)}{m_{\text{comb}}} & \frac{-\sin(\theta)}{m_{\text{comb}}} \\ \frac{\sin(\theta)}{m_{\text{comb}}} & \frac{\sin(\theta)}{m_{\text{comb}}} & \frac{\cos(\theta)}{m_{\text{comb}}} & \frac{\cos(\theta)}{m_{\text{comb}}} \\ \frac{-r}{J_{z,\text{comb}}} & \frac{r}{J_{z,\text{comb}}} & \frac{r_o + r_i}{J_{z,\text{comb}}} & \frac{-(r_o + r_i)}{J_{z,\text{comb}}} \end{pmatrix}, \quad (28)$$

$$\mathbf{C} = \mathbf{I}_{6 \times 6}, \quad \mathbf{D} = \mathbf{0}_{6 \times 4},$$

where  $J_{z,\text{comb}}$  is the inertia of the assembled system about the vertical axis and  $m_{\text{comb}}$  is the new mass. Linearization of the new control distribution matrix leads to

$$\mathbf{B}_{\text{LIN}} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{\cos(\theta_{\text{des}}) - \sin(\theta_{\text{des}})(\theta - \theta_{\text{des}})}{m_{\text{comb}}} & \frac{\cos(\theta_{\text{des}}) - \sin(\theta_{\text{des}})(\theta - \theta_{\text{des}})}{m_{\text{comb}}} & \frac{-\sin(\theta_{\text{des}}) + \cos(\theta_{\text{des}})(\theta - \theta_{\text{des}})}{m_{\text{comb}}} & \frac{-\sin(\theta_{\text{des}}) + \cos(\theta_{\text{des}})(\theta - \theta_{\text{des}})}{m_{\text{comb}}} \\ \frac{\sin(\theta_{\text{des}}) + \cos(\theta_{\text{des}})(\theta - \theta_{\text{des}})}{m_{\text{comb}}} & \frac{\sin(\theta_{\text{des}}) + \cos(\theta_{\text{des}})(\theta - \theta_{\text{des}})}{m_{\text{comb}}} & \frac{\cos(\theta_{\text{des}}) - \sin(\theta_{\text{des}})(\theta - \theta_{\text{des}})}{m_{\text{comb}}} & \frac{\cos(\theta_{\text{des}}) - \sin(\theta_{\text{des}})(\theta - \theta_{\text{des}})}{m_{\text{comb}}} \\ \frac{-r}{J_{z,\text{comb}}} & \frac{r}{J_{z,\text{comb}}} & \frac{r_o + r_i}{J_{z,\text{comb}}} & \frac{-(r_o + r_i)}{J_{z,\text{comb}}} \end{pmatrix} \quad (29)$$

The thrusters remained available after docking will be commanded by either spacecraft one or two, thanks to the

real-time wireless link (see [6]). Navigation for the assembled structure is performed onboard the robot acting as the

TABLE 3: Position, augmented state filter parameters.

$\bar{\mathbf{Q}}$	$(1 \times 10^{-6}) \cdot \mathbf{I}_{6 \times 6}$	Process covariance matrix, for the adapted augmented state filter
$\bar{\mathbf{R}}$	$(1 \times 10^{-4}) \cdot \mathbf{I}_{2 \times 2}$	Measurements covariance matrix, for the adapted augmented state filter
$\mathbf{P}_0$	$R_k \cdot \mathbf{I}_{6 \times 6}$	Initial covariance matrix, for the adapted augmented state filter
$T_s$	0.02 sec	Simulation sampling time

TABLE 4: Attitude, augmented state filter parameters.

$\bar{\mathbf{Q}}$	$(1 \times 10^{-12}) \cdot \mathbf{I}_{3 \times 3}$	Process covariance matrix, for the adapted augmented state filter
$\bar{\mathbf{R}}$	$(1 \times 10^{-2})$	Measurements covariance matrix, for the adapted augmented state filter
$\mathbf{P}_0$	$R_k \cdot \mathbf{I}_{3 \times 3}$	Initial covariance matrix, for the adapted augmented state filter
$T_s$	0.02 sec	Simulation sampling time

master. For the two-robot configuration in Figure 12, it follows the rigid body equations, being the attitude and attitude rate of the new spacecraft the same of the master, and its center of mass position and velocity deduced by those of the master (see also Figure 12).

## 7. Experimental Results: Four Spacecraft Simulators Assembly

In this section, assembly maneuvers are employed to experimentally test the suggested guidance, navigation, and control schemes. For our experiment, we do not implement any collision avoidance algorithm, which has been, however, previously successfully tested [23]. At the time of writing this paper, the simulators do not have hardware dedicated to relative measurements. Relative measurements are assumed to be the range, line of sight, and relative attitude. This information is obtained via software, by having the robots exchanging data over the ad-hoc wireless channel [6]. This feature has the benefit of flexibility in imposing the desired frequency of measurement update, by simple modification of the software. Furthermore, for the following experiments, we do not assume any particular noise or bias characteristics for the measurements, that is, the filter does not have that information. Noise is present, and it comes from the wireless communication. This assumption does not conflict with the previously stated contribution of our work in designing an estimation technique more robust than standard Kalman Filtering in the presence of low frequency updates, as demonstrated in the following.

Two experimental runs are presented. The first one demonstrates the unsuccessful relative navigation when classical Kalman Filtering is employed, considering the other S/C's maneuvers as a random process. Only two simulators are involved. The second experiment involves the four vehicles, showing how augmented state estimation can handle low measurement updates and unpredictable interruptions of updates, and still perform correct relative navigation, driving the mission to success. In particular, we are here imposing, via the wireless network, an update of 2 seconds. Once the two couples of robots are docked, each of the assembled structure is considered to be a new vehicle

with new mass and geometry. For this reason, the augmented state estimator is reinitialized for the new structure with different mass and inertia as in Table 2. For this part of the experimentation, the software is running only onboard the master S/C, that is, one pre-chosen unit for each couple.

The time step, or simulation sampling time, was chosen to be: (1) higher than the thrusters minimum actuation time (Tables 3 and 4), (2) in compatibility with CPU computational power, and (3) so to maintain the dynamics within the linearity range. The choice was also justified by previous experience with the employed hardware and by prior computer numerical simulations. In fact, the experimental activities at the Spacecraft Robotics Laboratory are always anticipated by high fidelity numerical simulations of the test-bed dynamics in Simulink, by visualizing on the computer how the experiment will develop. This prototyping approach reduces the time-to-market and trouble-shooting costs of newly developed GNC methodologies, by significantly cutting down the need for intermediate hardware prototypes and the number of experimental tests.

*7.1. The Classical Kalman Filter Technique.* Figure 13 is the bird's eye view of the experiment, demonstrating the unfeasibility of classical Kalman filter for spacecraft relative navigation, when relative measurements updates occur at low frequency. Two spacecraft simulators start maneuvering, with the goal of docking, from a short distance. The sides opposite to the bolded lines are the designated docking sides. After approximately 1 minute of maneuver, the accumulated error in relative state vector (position) exceeds the tolerance of the docking interfaces (Figure 2), driving the vehicles into a failed docking maneuver. A video of the experiment can be found online at [37].

*7.2. The Augmented State Estimation Technique.* Figure 14 is the bird's eye view of the experiment, demonstrating the feasibility of the augmented state estimation for spacecraft relative navigation. The main data for the filters are presented in Tables 3 and 4. Four spacecraft simulators start maneuvering, with the goal of assembling into a line-shaped structure, from short distances. The sides opposite to the bolded lines are the designated docking sides. After less than 3 minutes of

maneuver, the four vehicles successfully complete the given mission. The rectangular black and blue vehicles represent two spacecraft simulators docked and maneuvering as a single bigger unit. A video of the experiment can be found online at [38].

Once the simulators are assembled in couples, they maneuver as a single bigger unit. In particular, the augmented state estimation is reinitialized in order to switch to a new target vehicle in terms of relative navigation. In Figure 14, for the left couple, the cyan-represented unit acts as master of the new assembled cyan-red spacecraft. Likewise, for the right couple, the green vehicle is the master in the green-magenta assembly.

## 8. Conclusion

In this work, we are suggesting a complete solution for guidance, navigation, and control of planar multiple spacecraft assembly maneuvers. Guidance is performed by dynamically defining a desired state vector, so that the spacecraft can prepare for docking and correctly connect. The control is based on a real time LQR approach. As for the relative navigation, augmented state estimation is proposed, allowing for correct awareness of the other spacecraft configuration, even in the event of low frequency measurements update. The framework adapts itself to the evolving spacecraft, by switching among different values of mass properties and sensors and actuators configuration, when a new unit assembles to the aggregate.

Theoretical developments are presented for the three-degree-of-freedom case, considering a planar motion for the relative position and a single axis of rotation.

The experimental validation of the proposed methodology is presented, via floating spacecraft simulators, using an assembly maneuver as baseline. Experiments show how the augmented state estimation can cope with low frequency measurement updates, correctly performing the relative navigation, driving the mission to success. On the other hand, Classical Kalman Estimation, is not accurate for close distances with low frequency measurement updates as demonstrated in the three-degree-of-freedom experimental section. The dynamic guidance and control demonstrate real-time feasibility and the capability of performing autonomous assembly.

## Nomenclature

### Acronyms

COTS:	Commercial off the shelf
DoF:	Degree of freedom
GNC:	Guidance, navigation, and control
IE:	Input estimation
LQR:	Linear quadratic regulator
LVLH:	Local vertical local horizontal reference frame centered on the chaser spacecraft
NPS:	Naval postgraduate school

PWM: Pulse width modulation  
SRL: Spacecraft robotics laboratory.

### Variables and Symbols

$\alpha$ :	Docking safety cone semiaperture
$\beta$ :	Commanded orbiting angle around target docking port in docking phase
$\Delta t$ :	Control system sample time
$\omega_{LVLH}$ :	Chaser orbital angular velocity
$\theta$ :	Target attitude angle in chaser S/C body frame
$\dot{\theta}$ :	Target angular velocity in chaser S/C body frame
$C_{T_i}^j \Phi$ :	Transition Matrix of the dynamics of the $i$ th target S/C with respect to $j$ th chaser S/C
$C_{T_i}^j \Phi_A$ :	Augmented transition matrix of the dynamics of the $i$ th Target S/C with respect to $j$ th chaser S/C
$C_{T_i}^j \Theta$ :	Attitude state vector of $i$ th target S/C with respect to $j$ th chaser S/C
$C_{T_i}^j \Theta_A$ :	Augmented attitude state vector of $i$ th Target S/C with respect to $j$ th chaser S/C
$a$ :	Scaling factor in $\mathbf{R}$ LQR weighting matrix
$k$ :	Discrete time index
$m, m_{\text{comb}}$ :	Single & combined mass of the spacecraft simulator
$r$ :	Torque Arm: Thruster-center of mass arm
$\mathbf{r}_{\text{rsw}}$ :	Spacecraft-to-spacecraft vector
$\mathbf{r}_{\text{goal}}$ :	Docking port-to-corresponding docking spacecraft vector
$\mathbf{r}_{\text{port}}$ :	Center of mass to docking port vector
$r_{\text{dock}}$ :	Spacecraft-to-spacecraft transition distance between far away phase and docking phase
$t$ :	Time
$\mathbf{u}_C$ :	Chaser's control vector
$\mathbf{u}_T$ :	Target's control vector
$\mathbf{u}$ :	Relative control vector
$\mathbf{u}_{\text{LQR}}$ :	Optimal control vector
$u_{\text{max}} > 0$ :	Single engine maximum thrust
$u_{\text{thr}} > 0$ :	Threshold value for required thrust Before using PWM
$\mathbf{A}$ :	State matrix
$\mathbf{B}$ :	Control distribution matrix
$\mathbf{C}$ :	State-output mapping matrix
$\mathbf{D}$ :	Control-output mapping matrix
$C_{T_i}^j \mathbf{B}$ :	Control matrix referred of $i$ th target S/C with respect to $j$ th chaser S/C

$C_j^{T_i} \mathbf{B}(k)$ :	Discretized control matrix referred of $i$ th target S/C with respect to $j$ th chaser S/C
$C_j^{T_i} \mathbf{B}_A(k)$ :	Discretized augmented control matrix referred of $i$ th target S/C with respect to $j$ th chaser S/C
$C_j^{T_i} \mathbf{F}$ :	State matrix of $i$ th target S/C with respect to $j$ th chaser S/C
$C_j^{T_i} \mathbf{F}_A$ :	Augmented state matrix of $i$ th target S/C with respect to $j$ th chaser S/C
$\mathbf{G}$ :	Input noise matrix
$\mathbf{H}$ :	Measurement matrix
$\mathbf{G}_A$ :	Augmented input noise matrix
$\mathbf{H}_A$ :	Augmented measurement matrix
$J$ :	Cost function
$J_z, J_{z\_comb}$ :	Inertia of a single and combined Spacecraft simulator about the vertical axis
$\mathbf{K}_{LQR}$ :	LQR resulting gain matrix
$M_C$ :	Chaser spacecraft torque
$M_T$ :	Target spacecraft torque
$\mathbf{P}_0$ :	Initial state error covariance matrix
$\mathbf{Q}$ :	LQR state error weighting matrix
$\overline{\mathbf{Q}}_k$ :	Process noise covariance matrix
$\mathbf{R}$ :	LQR control effort weighting matrix
$R_k$ :	Measurement noise covariance
$T$ :	Maneuver total time
$T_s$ :	Sampling time
$\mathbf{V}$ :	Measurement noise vector, assumed to be gaussian white zero mean with covariance $R_k$
$V$ :	Scaling factor in $\mathbf{Q}$ LQR weighting matrix
$\mathbf{W}$ :	Input noise vector, assumed to be Gaussian white zero mean with covariance $Q_k$
$C_j^{T_i} \mathbf{X}$ :	State vector of $i$ th target S/C with respect to $j$ th chaser S/C,
$C_j^{T_i} \mathbf{X}_A$ :	Augmented state vector of $i$ th target S/C with respect to $j$ th chaser S/C
$\mathbf{X}_{rel}$ :	Relative state vector between two S/C
$C_j^{T_i} \mathbf{S}$ :	Complete $i$ th target state vector ref to $j$ th chaser S/C
$\mathbf{Z}$ :	Measurement vector
$\mathbf{Z}_A$ :	Augmented measurement vector
$\mathbf{I}_{p \times p}$ :	Identity matrix,
$\mathbf{0}_{q \times s}$ :	Zeros matrix
$[x \ y \ z]^T$ :	Target Cartesian coordinates in chaser S/C body frame
$[\dot{x} \ \dot{y} \ \dot{z}]^T$ :	Target linear velocities in chaser S/C body frame.

## Acknowledgments

This research was performed while Dr. Bevilacqua was holding a National Research Council Research Associateship Award at the Spacecraft Robotics Laboratory of the US Naval Postgraduate School.

## References

- [1] S. A. Frost and M. J. Balas, "Adaptive key component controllers for evolving systems," in *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, Honolulu, Hawaii, USA, August 2008.
- [2] M. J. Balas and S. A. Frost, "Evolving systems: a theoretical foundation," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, Colo, USA, August 2006.
- [3] T. S. VanZwieten and M. J. Balas, "Evolving systems: inheriting stability with decentralized adaptive control," in *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, Hilton Head, SC, USA, August 2007.
- [4] S. A. Frost and M. J. Balas, "Stabilizing controllers for evolving systems with application to flexible space structures," in *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, Hilton Head, SC, USA, August 2007.
- [5] S. Dong, K. Allen, P. Bauer et al., "Self-assembling wireless autonomously reconfigurable module design concept," *Acta Astronautica*, vol. 62, no. 2-3, pp. 246–256, 2008.
- [6] R. Bevilacqua, J. S. Hall, J. Homing, and M. Romano, "Ad hoc wireless networking and shared computation for autonomous multirobot systems," *Journal of Aerospace Computing, Information and Communication*, vol. 6, no. 5, pp. 328–353, 2009.
- [7] R. Bevilacqua and M. Romano, "Fuel-optimal spacecraft rendezvous with hybrid on-off continuous and impulsive thrust," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 4, pp. 1175–1178, 2007.
- [8] R. A. Singer, "Estimating optimal tracking filter performance for manned maneuvering targets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 6, no. 4, pp. 473–483, 1970.
- [9] Y. T. Chan and F. Couture, "Manoeuvre detection and track correction by input estimation," *IEE Proceedings F*, vol. 140, no. 1, pp. 21–28, 1993.
- [10] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, John Wiley & Sons, New York, NY, USA, 2001.
- [11] Y. Bar-shalom and K. Birmiwal, "Variable dimension filtering for maneuvering target tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 18, no. 5, pp. 621–629, 1982.
- [12] P. L. Bogler, "Tracking a maneuvering target using input estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 23, no. 3, pp. 298–310, 1987.
- [13] H. Khaloozadeh and A. Karsaz, "A new state augmentation for maneuvering targets detection," in *Proceedings of International Conference on Signal Processing and Communications (SPCOM '04)*, pp. 65–69, December 2004.
- [14] M. H. Bahari, A. Karsaz, and M. B. Naghibi-S, "Intelligent error covariance matrix resetting for maneuver target tracking," *Journal of Applied Sciences*, vol. 8, no. 12, pp. 2279–2285, 2008.
- [15] M. H. Bahari and N. Pariz, "High maneuvering target tracking using an input estimation technique associated with fuzzy

- forgetting factor,” *Scientific Research and Essays*, vol. 4, no. 10, pp. 936–944, 2009.
- [16] V. Pellegrini, R. Bevilacqua, M. Romano, and F. Curti, “Spacecraft proximity navigation and autonomous assembly based on augmented state estimation: analysis and experiments,” in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Toronto, Canada, August 2010.
- [17] S. B. McCamish, M. Romano, S. Nolet, C. M. Edwards, and D. W. Miller, “Flight testing of multiple-spacecraft control on SPHERES during close-proximity operations,” *Journal of Spacecraft and Rockets*, vol. 46, no. 6, pp. 1202–1213, 2009.
- [18] <http://mathworks.com/matlabcentral/fileexchange/2651>.
- [19] P. Zarchan and H. Musoff, *Fundamental of Kalman Filtering: A Practical Approach*, AIAA, Reston, Va, USA, 2009.
- [20] M. Romano, D. A. Friedman, and T. J. Shay, “Laboratory experimentation of autonomous spacecraft approach and docking to a collaborative target,” *Journal of Spacecraft and Rockets*, vol. 44, no. 1, pp. 164–173, 2007.
- [21] J. L. Schwartz, M. A. Peck, and C. D. Hall, “Historical review of air-bearing spacecraft simulators,” *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 4, pp. 513–522, 2003.
- [22] F. Curti, M. Romano, and R. Bevilacqua, “Lyapunov-based thrusters’ selection for spacecraft control: analysis and experimentation,” *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 4, pp. 1143–1160, 2010.
- [23] R. Bevilacqua, T. Lehmann, and M. Romano, “Development and experimentation of LQR/APF guidance and control for autonomous proximity maneuvers of multiple spacecraft,” *Acta Astronautica*, vol. 68, no. 7-8, pp. 1260–1275, 2011.
- [24] <https://www.rtai.org/RTAILAB/RTAI-Lab-tutorial.pdf>.
- [25] <http://www.mathworks.com/products/xpctarget/>.
- [26] L. Dozio and P. Mantegazza, “Real time distributed control system using RTAI,” in *Proceedings of the 6th IEE International Symposium on Object-Oriented Real-Time Distributed Computing*, May 2003.
- [27] G. Quaranta and P. Mantegazza, “Using MATLAB-simulink RTW to build real time control applications in user space with RTAI-LXRT,” in *Realtime Linux Workshop*, Milan, Italy, 2001.
- [28] S. Ricci and A. Scotti, “Aeroelastic testing on a three surface airplane,” in *Proceedings of the 47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, pp. 7131–7147, Newport, RI, USA, May 2006.
- [29] L. Dozio, A. Toso, W. Corbetta, E. Vigoni, and G. Ghiringhelli, “Comparison of feedback and feedforward strategies for active structural—acoustic control of broadband sound transmission into a cavity,” in *Proceedings of the 13th AIAA/CEAS Aeroacoustics Conference, 28th AIAA Aeroacoustics Conference*, 2007.
- [30] M. G. Perhinschi, M. R. Napolitano, G. Campa, B. Seanor, S. Gururajan, and G. Yu, “Design and flight testing of intelligent flight control laws for the WVU YF-22 model aircraft,” in *Proceedings of AIAA Guidance, Navigation, and Control Conference*, pp. 5925–5936, San Francisco, Calif, USA, August 2005.
- [31] Y. Zheng, B. J. Martin, and N. Villaume, “VERSE—virtual equivalent real-time simulation environment,” in *Proceedings of AIAA Modeling and Simulation Technologies Conference*, pp. 368–376, San Francisco, Calif, USA, August 2005.
- [32] W. H. Clohessy and R. S. Wiltshire, “Terminal guidance system for satellite rendezvous,” *Journal of the Aerospace Sciences*, vol. 27, no. 9, pp. 653–658, 1960.
- [33] S. D’Amico, E. Gill, M. Garcia, O. Montenbruck, and E. Gill, “GPS-based real-time navigation for the PRISMA formation flying mission,” in *Proceedings of the 3rd ESA Workshop on Satellite Navigation User Equipment Technologies (NAVITEC ’06)*, Noordwijk, The Netherlands, December 2006.
- [34] S. D’Amico, J. S. Ardaens, and O. Montenbruck, “Navigation of formation flying spacecraft using GPS: the PRISMA technology demonstration,” in *Proceedings of the 22nd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION-GNSS ’09)*, pp. 2250–2264, Savannah, Ga, USA, September 2009.
- [35] K. Ogata, *Discrete-Time Control Systems*, Prentice-Hall, New York, NY, USA, 2nd edition, 1995.
- [36] C. Lugini and M. Romano, “A ballistic-pendulum test stand to characterize small cold-gas thruster nozzles,” *Acta Astronautica*, vol. 64, no. 5-6, pp. 615–625, 2009.
- [37] [http://aa.nps.edu/~mromano/Movies4Web/GNC10\\_Video1.flv](http://aa.nps.edu/~mromano/Movies4Web/GNC10_Video1.flv).
- [38] [http://aa.nps.edu/~mromano/Movies4Web/GNC10\\_Video2.flv](http://aa.nps.edu/~mromano/Movies4Web/GNC10_Video2.flv).

## Research Article

# Cascade-Based Controlled Attitude Synchronization and Tracking of Spacecraft in Leader-Follower Formation

Rune Schlanbusch,<sup>1</sup> Antonio Loría,<sup>2</sup> and Per Johan Nicklasson<sup>1</sup>

<sup>1</sup>Department of Technology, Narvik University College, PB 385, 8505 Narvik, Norway

<sup>2</sup>CNRS, LSS-SUPELEC, 3 rue Joliot Curie, 91192 Gif-sur-Yvette, France

Correspondence should be addressed to Rune Schlanbusch, [runsch@hin.no](mailto:runsch@hin.no)

Received 30 June 2010; Revised 5 January 2011; Accepted 23 February 2011

Academic Editor: Giampiero Campa

Copyright © 2011 Rune Schlanbusch et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose controllers for leader-follower attitude synchronization of spacecraft formations in the presence of disturbances, that is, the leader spacecraft is controlled to follow a given reference, while a follower spacecraft is controlled to synchronize its motion with the leader's. In the ideal disturbance-free scenario, we show that synchronization takes place asymptotically. Moreover, we show the property of uniform practical asymptotic stability which implies that the synchronization is robust to bounded disturbances.

## 1. Introduction

In recent years, formation flying has become an increasingly popular subject of study. This is a new method of performing space operations, by replacing large and complex spacecraft with an array of simpler microspacecraft, bringing out new possibilities and opportunities of cost reduction, redundancy, and improved resolution aspects of onboard payload. One of the main challenges is the requirement of synchronization between spacecraft; robust and reliable control of relative position and attitude is necessary to make the spacecraft cooperate to gain the possible advantages made feasible by spacecraft formations. For fully autonomous spacecraft formations, both path- and attitude-planning must be performed online which introduces challenges like collision avoidance and restrictions on pointing instruments towards required targets, with the lowest possible fuel expenditure.

Synchronization of dynamical systems was first studied by Christian Huygens in the XVIIth century. In recent years, the problem has obtained increasing interest in various research areas due to its impact in technology development and challenges it imposes; see, for example, [1–4].

Model-based controlled synchronization consists in using the physics laws and control theory in order to induce synchronization in dynamical systems. Successful

instances include synchronization of robot manipulators [5, 6], leader-follower spacecraft formations [7–10], and ship replenishment operations [11, 12]. Another form of synchronization is consensus, in which a group of systems coordinate their motion without any subsystem having a higher hierarchy with respect to the others. An instance of consensus is cooperative control in which a group of systems is controlled in a way that they collaborate in order to achieve a task as a team of agents. Examples may be found in the areas of autonomous vehicles [13–15], underactuated marine vessels [16, 17], and rigid bodies [18–20].

In this paper, we address the simultaneous control problems of attitude tracking and leader-follower synchronization. That is, we propose a tracking controller for the leader spacecraft which makes it follow a prescribed reference. Independently, we construct a synchronization control law for the follower spacecraft which makes it track the attitude of the leader, thereby synchronizing in the classical master-slave configuration.

Our controllers are reminiscent of classical tracking controllers for robot manipulators passivity-based control which exploits the system's physical properties; see [21]. For tracking control, see the passivity-based PD+ of [22] and the wrongly called “sliding-mode” controller of [23] which may rather be casted in the passivity-based framework.

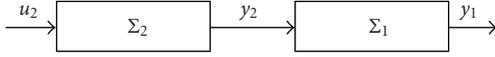


FIGURE 1: Cascade interconnection of two dynamical systems.

Although insightful, these popular control approaches for robot manipulators may not be directly applied in spacecraft tracking control and synchronization. The first obstacle is the specificity of spacecraft nonlinear models, expressed in quaternion coordinates. We revise the model in the following section.

Besides the difficulties imposed by the modeling of spacecraft, simultaneous tracking control and master-slave synchronization implicitly suggest controlling the leader spacecraft towards a reference *independently* of the slave system dynamics. Correspondingly, the synchronization controller inevitably couples the follower spacecraft to the dynamics of the leader. However, the synchronization controller is demanded to achieve the task regardless of the master dynamics as well as the reference that system intends to track. The ability to control two coupled systems separately is called *separation principle* and is known *not* to hold in general for nonlinear systems (see e.g. [24]). This is where cascades theory enters in play.

Cascaded systems theory consists in analyzing complex systems by dividing them in subsystems simpler to control and to analyze (see [25] and references within). It must be emphasized that such representation is purely schematic, for the purpose of analysis only. Generally speaking, the stability analysis problem consists in finding conditions for two systems as in Figure 1 so that, considering that both subsystems separately are stable, they conserve that property when interconnected.

In the context of the present paper, the block on the left corresponds to the leader system in *closed* loop with a tracking controller, while the block on the right consists in the follower spacecraft in closed loop with the synchronization controller. The blocks are interconnected via the tracking errors of the leader system. Hence, in the ideal case, when the leader spacecraft is perfectly controlled, the systems are decoupled.

The topic of cascaded systems have received a great deal of attention and has successfully been applied to a wide number of applications. In [26], a cascaded adaptive control scheme for marine vehicles including actuator dynamics was introduced, while [27] solved the problem of synchronization of two pendula through use of cascades. The authors of [28] studied the problem of global stabilizability of feed-forward systems by a systematic recursive design procedure for autonomous systems, while time-varying systems were considered in [29] for stabilization of robust control, while [30] established sufficient conditions for uniform global asymptotical stability (UGAS) of cascaded nonlinear time-varying systems. The aspect of practical and semiglobal stability for nonlinear time-varying systems in cascade was pursued in [31, 32]. A stability analysis of spacecraft formations including both leader and follower using relative coordinates is presented in [10], where the controller-observer

scheme is proven input-to-state stable, and backstepping was applied in [17] for leader-follower formation control of multiple underactuated autonomous underwater vehicles (AUVs). For the control problems at hand, we show that the closed-loop system has the property of uniform asymptotic stability. Significantly, uniform asymptotic stability guarantees robustness with respect to bounded disturbances. In this regard, we extend our result to the case where bounded perturbations affect the system (atmospheric drag, gravity gradient, etc.). In this scenario, we guarantee uniform practical asymptotic stability. This pertains to the case when there exists a steady-state tracking and synchronization error which can be arbitrarily diminished via an appropriate tuning of the control parameters.

The contribution of this paper is application of the framework for stability analysis of cascaded systems of rigid bodies in leader-follower formation and synchronization of PD+ and sliding surface control laws adapted for quaternion space. The equilibrium points of the PD+ controller in closed loop with the rigid body dynamics are proven uniformly asymptotically stable (UAS) when disturbances are considered known, while a sliding surface controller is utilized to prove uniform practical asymptotical stability (UPAS) when disturbances are considered unknown but bounded. Simulation results of a leader-follower spacecraft formation are presented to show the performance of our proposed control laws.

The rest of the paper is organized as follows: in Section 2, we recall the modeling of rigid bodies in quaternion coordinates and present the main theoretical tools of cascaded systems theory, which are instrumental to our control design. Controller design is presented in Section 3 for known disturbances while for unknown but upper-bounded disturbances in Section 4. Simulation results are presented in Section 5, and everything is wrapped up with conclusions in Section 6.

## 2. Mathematical Background

In the following, we denote by  $\dot{\mathbf{x}}$  the time derivative of a vector  $\mathbf{x}$ , that is,  $\dot{\mathbf{x}} = d\mathbf{x}/dt$ ; moreover,  $\ddot{\mathbf{x}} = d^2\mathbf{x}/dt^2$ , and  $\|\cdot\|$  denotes the  $\ell^2$  norm of vectors and induced  $\ell^2$  norm of matrices. Coordinate reference frames are denoted by  $\mathcal{F}^{(\cdot)}$ , where the superscript denotes the frame in question. Moreover, we denote a rotation matrix between frame  $a$  and frame  $b$  by  $\mathbf{R}_a^b \in \text{SO}(3)$ , and the angular velocity of frame  $a$  relative to frame  $b$ , referenced in frame  $c$ , is denoted by  $\omega_{b,a}^c \in \mathbb{R}^3$ . We denote by  $\mathbf{x}(t; t_0, \mathbf{x}(t_0))$  the solutions of the differential equation  $\dot{\mathbf{x}} = f(t, \mathbf{x})$  with initial conditions  $(t_0, \mathbf{x}(t_0))$ . When the context is sufficiently explicit, we may omit the arguments of a function, vector or matrix.

*2.1. Cartesian Coordinate Frames.* The coordinate reference frames used throughout the paper are defined as follows.

*Earth-Centered Inertial Frame.* The Earth-centered inertial (ECI) frame is denoted by  $\mathcal{F}^i$  and has its origin in the center of the Earth. The axes are denoted by  $x_i, y_i,$  and  $z_i$ , where

the  $z_i$ -axis is directed along the axis of rotation of the Earth toward the celestial North Pole, the  $x_i$ -axis is pointing in the direction of the vernal equinox,  $\mathbf{Y}$ , which is the vector pointing from the center of the sun toward the center of the Earth during the vernal equinox, and finally the  $y_i$ -axis completes the right-handed orthonormal frame.

*Spacecraft Orbit Reference Frame.* The orbit frame is denoted by  $\mathcal{F}^{so}$ , where the sub-/superscript  $s = l, f$  denotes the leader and follower spacecraft, respectively, such that we throughout the paper denote, for example,  $\mathcal{F}^{lo}$  and  $\mathcal{F}^{fo}$  as  $\mathcal{F}^{so}$ , which has its origin located in the center of the mass of the spacecraft. The  $\mathbf{e}_r$ -axis in the frame coincides with the vector  $\mathbf{r} = [r_x, r_y, r_z]^\top$  from the center of the Earth to the spacecraft, and the  $\mathbf{e}_h$ -axis is parallel to the orbital angular momentum vector, pointing in the orbit normal direction. The  $\mathbf{e}_\theta$ -axis completes the right-handed orthonormal frame. The basis vectors of the frame can be defined as

$$\mathbf{e}_r = \frac{\mathbf{r}}{r}, \quad \mathbf{e}_\theta = \mathbf{e}_h \times \mathbf{e}_r, \quad \mathbf{e}_h = \frac{\mathbf{h}}{h}, \quad (1)$$

where  $\mathbf{h} = \mathbf{r} \times \dot{\mathbf{r}}$  is the angular momentum vector of the orbit,  $h = |\mathbf{h}|$  and  $r = |\mathbf{r}|$ . This frame is also known as the local vertical/local horizontal (LVLH) frame.

*Spacecraft Body Reference Frame.* The body frame of the spacecraft is denoted by  $\mathcal{F}^{sb}$  and is located at the center of the mass of the spacecraft, and its basis vectors are aligned with the principle axis of inertia.

*2.2. Quaternions.* The attitude of a rigid body is often represented by a rotation matrix  $\mathbf{R} \in \text{SO}(3)$  fulfilling

$$\text{SO}(3) = \{\mathbf{R} \in \mathbb{R}^{3 \times 3} : \mathbf{R}^\top \mathbf{R} = \mathbf{I}, \det \mathbf{R} = 1\}, \quad (2)$$

which is the special orthogonal group of order three. Quaternions are often used to parameterize members of  $\text{SO}(3)$  where the unit quaternion is defined as  $\mathbf{q} = [\eta, \boldsymbol{\epsilon}^\top]^\top \in \mathbb{S}^3 = \{\mathbf{x} \in \mathbb{R}^4 : \mathbf{x}^\top \mathbf{x} = 1\}$ , where  $\eta \in \mathbb{R}$  is the scalar part and  $\boldsymbol{\epsilon} \in \mathbb{R}^3$  is the vector part. The rotation matrix may be described by [33]

$$\mathbf{R} = \mathbf{I} + 2\eta \mathbf{S}(\boldsymbol{\epsilon}) + 2\mathbf{S}^2(\boldsymbol{\epsilon}), \quad (3)$$

where the matrix  $\mathbf{S}(\cdot)$  is the cross product operator defined as

$$\mathbf{S}(\boldsymbol{\epsilon}) = \boldsymbol{\epsilon} \times = \begin{bmatrix} 0 & -\epsilon_z & \epsilon_y \\ \epsilon_z & 0 & -\epsilon_x \\ -\epsilon_y & \epsilon_x & 0 \end{bmatrix}, \quad (4)$$

where  $\boldsymbol{\epsilon} = [\epsilon_x, \epsilon_y, \epsilon_z]^\top$ . The inverse rotation can be performed by using the inverse conjugate of  $\mathbf{q}$  as  $\bar{\mathbf{q}} = [\eta, -\boldsymbol{\epsilon}^\top]^\top$ . The set  $\mathbb{S}^3$  forms a group with quaternion multiplication, which is distributive and associative, but not commutative, and the quaternion product of two arbitrary quaternions  $\mathbf{q}_1$  and  $\mathbf{q}_2$  is defined as [33]

$$\mathbf{q}_1 \otimes \mathbf{q}_2 = \begin{bmatrix} \eta_1 \eta_2 - \boldsymbol{\epsilon}_1^\top \boldsymbol{\epsilon}_2 \\ \eta_1 \boldsymbol{\epsilon}_2 + \eta_2 \boldsymbol{\epsilon}_1 + \mathbf{S}(\boldsymbol{\epsilon}_1) \boldsymbol{\epsilon}_2 \end{bmatrix}. \quad (5)$$

*2.3. Kinematics and Dynamics.* The time derivative of (3) can be written as [33]

$$\dot{\mathbf{R}}_b^a = \mathbf{S}(\boldsymbol{\omega}_{a,b}^a) \mathbf{R}_b^a = \mathbf{R}_b^a \mathbf{S}(\boldsymbol{\omega}_{a,b}^b), \quad (6)$$

and the kinematic differential equations can be expressed as [33]

$$\dot{\mathbf{q}}_s = \mathbf{T}(\mathbf{q}_s) \boldsymbol{\omega}_{i,sb}^{sb}, \quad (7)$$

where

$$\mathbf{T}(\mathbf{q}_s) = \frac{1}{2} \begin{bmatrix} -\boldsymbol{\epsilon}_s^\top \\ \eta_s \mathbf{I} + \mathbf{S}(\boldsymbol{\epsilon}_s) \end{bmatrix} \in \mathbb{R}^{4 \times 3}. \quad (8)$$

The dynamical model of a rigid body can be described by a differential equation for angular velocity and is deduced from Euler's moment equation. This equation describes the relationship between applied torque and angular momentum on a rigid body as [34]

$$\mathbf{J}_s \dot{\boldsymbol{\omega}}_{i,sb}^{sb} = -\mathbf{S}(\boldsymbol{\omega}_{i,sb}^{sb}) \mathbf{J}_s \boldsymbol{\omega}_{i,sb}^{sb} + \boldsymbol{\tau}_s^{sb}, \quad (9)$$

where  $\boldsymbol{\tau}_s^{sb} \in \mathbb{R}^3$  is the total torque working on the body frame and  $\mathbf{J}_s = \text{diag}\{J_{sx}, J_{sy}, J_{sz}\} \in \mathbb{R}^{3 \times 3}$  is the moment of inertia. The torque working on the body is expressed as  $\boldsymbol{\tau}_s^{sb} = \boldsymbol{\tau}_{sa}^{sb} + \boldsymbol{\tau}_{sd}^{sb}$ , where  $\boldsymbol{\tau}_{sa}^{sb}$  is the disturbance torque and  $\boldsymbol{\tau}_{sa}^{sb}$  is the actuator (control) torque. It might be desirable to express the rotation between the body frame and the orbit frame which can be done by applying

$$\boldsymbol{\omega}_{s,sb}^{sb} = \boldsymbol{\omega}_{i,sb}^{sb} - \mathbf{R}_i^{sb} \boldsymbol{\omega}_{i,s}^i, \quad (10)$$

where  $\boldsymbol{\omega}_{i,s}^i = \mathbf{S}(\mathbf{r}_s^i) \mathbf{v}_s^i / \mathbf{r}_s^{\top} \mathbf{r}_s^i$ , and  $\mathbf{r}_s^i$  and  $\mathbf{v}_s^i$  are the spacecraft radius and velocity vector, respectively, represented in the inertial frame.

*2.4. Cascaded Systems Theory.* A typical nonlinear cascaded time-varying system on closed-loop dynamical form is expressed as

$$\Sigma_1 : \dot{x}_1 = f_1(t, x_1) + g(t, x) x_2, \quad (11)$$

$$\Sigma_2 : \dot{x}_2 = f_2(t, x_2), \quad (12)$$

where  $x_1 \in \mathbb{R}^n$ ,  $x_2 \in \mathbb{R}^m$ ,  $x = [x_1^\top, x_2^\top]^\top$ , and the functions  $f_1(\cdot, \cdot)$ ,  $f_2(\cdot, \cdot)$ , and  $g(\cdot, \cdot)$  are continuous in their arguments, locally Lipschitz in  $x$ , and uniform in  $t$ , and  $f_1(\cdot, \cdot)$  is continuously differentiable in both arguments. Note that (12) is decoupled from (11), hence, it will be called the *driving* system, and its state enters as an input to the upper system with state  $x_1$  through the *interconnection* term  $g(t, x) x_2$ .

In the context of this paper, the dynamics

$$\dot{x}_1 = f_1(t, x_1) \quad (13)$$

represents the synchronization error dynamics of the leader-follower configuration, assuming that perfect tracking is achieved for the leader system, that is, the tracking error for

the latter is represented by  $x_2$ , and its closed-loop dynamics under tracking control will be represented by (12).

We will evoke [25, Theorem 1] to prove uniform asymptotic stability of the equilibrium point of a closed-loop system on the form (11)-(12) under the following assumptions.

*Assumption 1.* There exist constants  $c_1, c_2, \delta' > 0$  and a Lyapunov function  $V(t, x_1)$  for (13) such that  $V : \mathbb{R}_{\geq 0} \times \mathbb{R}^n \rightarrow \mathbb{R}_{\geq 0}$  is positive definite, radially unbounded,  $\dot{V}(t, x_1) \leq 0$  and

$$\left\| \frac{\partial V}{\partial x_1} \right\| \|x_1\| \leq c_1 V(t, x_1) \quad \forall \|x_1\| \geq \delta', \quad (14)$$

$$\left\| \frac{\partial V}{\partial x_1} \right\| \leq c_2 \quad \forall \|x_1\| \leq \delta'. \quad (15)$$

*Assumption 2.* There exist two continuous functions  $\xi_1, \xi_2 : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$  such that  $g(t, x)$  satisfies

$$\|g(t, x)\| \leq \xi_1(\|x_2\|) + \xi_2(\|x_2\|)\|x_1\|. \quad (16)$$

*Assumption 3.* There exists a class  $\mathcal{K}$  function  $\alpha(\cdot)$  such that, for all  $t_0 \geq 0$ , the trajectories of the system (12) satisfy

$$\int_{t_0}^{\infty} \|x_2(t; t_0, x_2(t_0))\| dt \leq \alpha(\|x_2(t_0)\|). \quad (17)$$

The Theorem cited above may be extended to the case when the subsystems possess the weaker property of practical asymptotic stability. This pertains to the situation in which the errors do not converge to zero but to a bounded region that may be made arbitrarily small; see [35]. A related popular concept, for instance, in control of mechanical systems is that of ultimate boundedness. However, practical asymptotic stability is stronger than ultimate boundedness since the later is only a notion of convergence and does not imply stability in the sense of Lyapunov. In other words, the fact that the errors converge to a bounded region does not imply that they remain always arbitrarily close to it.

### 3. Uniform Asymptotic Stabilization

We are ready to present the first results on tracking and synchronization cascade-based control. The control strategy relies on using models for two single spacecraft coupled through synchronized control, and stability properties are proved using cascade theory for known disturbances, that is, we assume that the disturbances  $\tau_{id}^{lb}$  and  $\tau_{fd}^{fb}$  can be modeled correctly (see, e.g., [34, 36, 37]).

*3.1. Problem Formulation.* The control problem is to design two controllers; the first one makes the states  $\mathbf{q}_l(t)$ ,  $\omega_{i,lb}^{lb}(t)$ , and  $\dot{\omega}_{i,lb}^{lb}(t)$  converge towards the generated references specified as  $\mathbf{q}_d(t)$ ,  $\omega_{i,d}^{lb}(t)$ , and  $\dot{\omega}_{i,d}^{lb}(t)$ , satisfying the kinematic equation

$$\dot{\mathbf{q}}_d = \mathbf{T}(\mathbf{q}_d)\omega_{i,d}^{lb}, \quad (18)$$

and acts as a solution to the dynamical model presented in (9). It should be noted that we apply (10) and its derivative to the generated reference rather than the dynamical equation to keep a simple control law structure compared to [38]. The second controller is designed as a synchronizing controller to make the states  $\mathbf{q}_f(t)$  and  $\omega_{i,fb}^{fb}(t)$  converge towards  $\mathbf{q}_l(t)$  and  $\omega_{i,lb}^{fb}(t)$ . The error quaternion  $\tilde{\mathbf{q}}_s = [\tilde{\eta}_s, \tilde{\boldsymbol{\epsilon}}_s^\top]^\top$  is found by applying the quaternion product

$$\tilde{\mathbf{q}}_s = \mathbf{q}_s \otimes \bar{\mathbf{q}}_d = \begin{bmatrix} \eta_s \eta_d + \boldsymbol{\epsilon}_s^\top \boldsymbol{\epsilon}_d \\ \eta_d \boldsymbol{\epsilon}_s - \eta_s \boldsymbol{\epsilon}_d - \mathbf{S}(\boldsymbol{\epsilon}_s) \boldsymbol{\epsilon}_d \end{bmatrix}, \quad (19)$$

where the sub-/superscript  $s = l, f$  denotes the leader and follower spacecraft, respectively, and the error kinematics can be expressed as [39]

$$\dot{\tilde{\mathbf{q}}}_s = \frac{1}{2} \mathbf{T}(\tilde{\mathbf{q}}_s) \mathbf{e}_{s\omega}, \quad (20)$$

where  $\mathbf{e}_{s\omega} = \omega_{i,fb}^{sb} - \omega_{i,d}^{sb}$  is the angular velocity error. Due to the redundancy in the quaternion representation,  $\tilde{\mathbf{q}}_s$  and  $-\tilde{\mathbf{q}}_s$  represent the same physical attitude, but, mathematically, it differs by a  $2\pi$  rotation about an arbitrary axis. Therefore, we are not able to achieve a global representation since the term global refers to the whole state space  $\mathbb{R}^n$  according to [35]. Since both equilibrium points represent the same physical representation, the equilibrium point which requires the shortest rotation is usually chosen, thus minimizing the path length; hence,  $\tilde{\mathbf{q}}_{s+} = [1, \mathbf{0}^\top]^\top$  is chosen if  $\tilde{\eta}_s(t_0) \geq 0$  and  $\tilde{\mathbf{q}}_{s-} = [-1, \mathbf{0}^\top]^\top$  is chosen if  $\tilde{\eta}_s(t_0) < 0$ . For the positive equilibrium point, an attitude error function is chosen as  $\mathbf{e}_{sq+} = [1 - \tilde{\eta}_s, \tilde{\boldsymbol{\epsilon}}_s^\top]^\top$  (see [9]), while, for the negative equilibrium point, the attitude error function is chosen as  $\mathbf{e}_{sq-} = [1 + \tilde{\eta}_s, \tilde{\boldsymbol{\epsilon}}_s^\top]^\top$ . In accordance with general kinematic relations

$$\dot{\mathbf{e}}_{sq\pm} = \mathbf{T}_{se}(\mathbf{e}_{sq\pm}) \mathbf{e}_{s\omega}, \quad (21)$$

where

$$\mathbf{T}_{se}(\mathbf{e}_{sq\pm}) = \frac{1}{2} \begin{bmatrix} \pm \tilde{\boldsymbol{\epsilon}}_s^\top \\ \tilde{\eta}_s \mathbf{I} + \mathbf{S}(\tilde{\boldsymbol{\epsilon}}_s) \end{bmatrix}. \quad (22)$$

The attitude error function is chosen *a priori* and kept throughout the maneuver even if  $\tilde{\eta}_s(t)$  should happen to switch sign for some  $t$ . The control problem is presented as a cascaded system on the form (11)-(12), where the states are defined as  $\mathbf{x}_1 = [\mathbf{e}_{fq}^\top, \mathbf{e}_{f\omega}^\top]^\top$  and  $\mathbf{x}_2 = [\mathbf{e}_{lq}^\top, \mathbf{e}_{l\omega}^\top]^\top$ . The control objective is to stabilize the equilibrium point  $(\mathbf{e}_{fq}(t), \mathbf{e}_{f\omega}(t), \mathbf{e}_{lq}(t), \mathbf{e}_{l\omega}(t)) = (\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0})$  as  $t \rightarrow \infty$  for all initial values.

*3.2. Control of Leader Spacecraft.* For control of the leader spacecraft attitude, we incorporate a model-dependent control law as in [40]. It is assumed that we have available information of its attitude  $\mathbf{q}_l$ , angular velocity  $\omega_{i,lb}^{lb}$ , and inertia matrix  $\mathbf{J}_l$ , and, temporarily, we assume to know external perturbations. We choose the equilibrium such that  $\mathbf{e}_{lq\pm} = [1 \mp \tilde{\eta}_l, \tilde{\boldsymbol{\epsilon}}_l^\top]^\top$  is either the positive or negative

equilibrium point, which does not change during the maneuver. By pure conventionalism, consider the positive equilibrium, that is,  $\mathbf{e}_{lq} = \mathbf{e}_{lq+}$ . We define desired attitude  $\mathbf{q}_d(t)$ , desired angular velocity  $\boldsymbol{\omega}_{i,d}^{lb}(t)$ , and desired angular acceleration  $\dot{\boldsymbol{\omega}}_{i,d}^{lb}(t)$  which are all bounded functions. The control law is expressed as

$$\boldsymbol{\tau}_{la}^{lb} = \mathbf{J}_l \dot{\boldsymbol{\omega}}_{i,d}^{lb} - \mathbf{S}(\mathbf{J}_l \boldsymbol{\omega}_{i,lb}^{lb}) \boldsymbol{\omega}_{i,d}^{lb} - k_{lq} \mathbf{T}_{le}^\top \mathbf{e}_{lq} - k_{l\omega} \mathbf{e}_{l\omega} - \boldsymbol{\tau}_{ld}^{lb}, \quad (23)$$

where  $k_{lq} > 0$  and  $k_{l\omega} > 0$  are feedback gain scalars. By insertion of (23) into (9), the system is on form (12), and, by applying the property  $\mathbf{S}(\boldsymbol{\omega}_{i,lb}^{lb}) \mathbf{J}_l \boldsymbol{\omega}_{i,lb}^{lb} = -\mathbf{S}(\mathbf{J}_l \boldsymbol{\omega}_{i,lb}^{lb}) \boldsymbol{\omega}_{i,lb}^{lb}$ , we obtain the closed-loop dynamics

$$\mathbf{J}_l \dot{\mathbf{e}}_{l\omega} + (k_{l\omega} \mathbf{I} - \mathbf{S}(\mathbf{J}_l \boldsymbol{\omega}_{i,lb}^{lb})) \mathbf{e}_{l\omega} + k_{lq} \mathbf{T}_{le}^\top \mathbf{e}_{lq} = \mathbf{0}. \quad (24)$$

A radial unbounded, positive definite Lyapunov function candidate is defined as

$$V_l = \frac{1}{2} \mathbf{e}_{l\omega}^\top \mathbf{J}_l \mathbf{e}_{l\omega} + \frac{1}{2} \mathbf{e}_{lq}^\top k_{lq} \mathbf{e}_{lq} > 0 \quad \forall \mathbf{e}_{l\omega} \neq \mathbf{0}, \mathbf{e}_{lq} \neq \mathbf{0}. \quad (25)$$

Indeed, we have

$$\frac{1}{2} \min\{j_{lm}, k_{lq}\} \|\mathbf{x}_2\|^2 \leq V_l \leq \frac{1}{2} \max\{j_{lM}, k_{lq}\} \|\mathbf{x}_2\|^2, \quad (26)$$

where  $j_{lm} \leq \|J_l\| \leq j_{lM}$ . By differentiation of (25) and inserting (24) and (21), we obtain

$$\dot{V}_l = \mathbf{e}_{lq}^\top k_{lq} \mathbf{T}_{le} \mathbf{e}_{l\omega} + \mathbf{e}_{l\omega}^\top [\mathbf{S}(\mathbf{J}_l \boldsymbol{\omega}_{i,lb}^{lb}) - k_{l\omega} \mathbf{I}] \mathbf{e}_{l\omega} - \mathbf{e}_{l\omega}^\top k_{lq} \mathbf{T}_{le}^\top \mathbf{e}_{lq}, \quad (27)$$

where the first part of the second term in (27) is zero because  $\mathbf{S}(\mathbf{J}_l \boldsymbol{\omega}_{i,lb}^{lb})$  is a skew-symmetric matrix, which leads to

$$\dot{V}_l = -\mathbf{e}_{l\omega}^\top k_{l\omega} \mathbf{e}_{l\omega} \leq 0 \quad (28)$$

so the origin of the system is uniformly stable and the trajectories are bounded. The rest of the proof consists in showing that the position errors and the velocity tracking errors are square integrable. Then it suffices to invoke [41, Lemma 3].

Let  $\mathcal{V}_l(t) = V_l(\mathbf{e}_{lq}(t), \mathbf{e}_{l\omega}(t))$  and  $\mathbf{x}_2(t) = [\mathbf{e}_{lq}(t)^\top, \mathbf{e}_{l\omega}(t)^\top]^\top$ . Then, from (28), we obtain by integrating on both sides

$$\int_{t_0}^t \dot{\mathcal{V}}_l(s) ds = - \int_{t_0}^t \mathbf{e}_{l\omega}^\top(s) k_{l\omega} \mathbf{e}_{l\omega}(s) ds, \quad (29)$$

$$-\mathcal{V}_l(t) + \mathcal{V}_l(t_0) = k_{l\omega} \int_{t_0}^t \|\mathbf{e}_{l\omega}(s)\|^2 ds. \quad (30)$$

Since  $-\mathcal{V}_l(t) \leq 0$ , we can write

$$\begin{aligned} k_{l\omega} \int_{t_0}^t \|\mathbf{e}_{l\omega}\|^2 ds &\leq \mathcal{V}_l(t_0) \\ &= \frac{1}{2} (\mathbf{e}_{l\omega}^\top(t_0) \mathbf{J}_l \mathbf{e}_{l\omega}(t_0) + \mathbf{e}_{lq}^\top(t_0) k_{lq} \mathbf{e}_{lq}(t_0)) \\ &\leq \frac{1}{2} \max\{j_{lM}, k_{lq}\} \|\mathbf{x}_2(t_0)\|^2, \end{aligned} \quad (31)$$

and thus

$$\int_{t_0}^t \|\mathbf{e}_{l\omega}(s)\|^2 ds \leq c_3 \|\mathbf{x}_2(t_0)\|^2, \quad (32)$$

where  $c_3 = \max\{j_{lM}, k_{lq}\}/2k_{l\omega}$ .

Now let  $\mathcal{W}_l(t) = W_l(\mathbf{e}_{lq}(t), \mathbf{e}_{l\omega}(t))$  such that

$$\mathcal{W}_l(t) = \mathbf{e}_{lq}(t)^\top \mathbf{T}_{le}(t) k_{lq} \mathbf{J}_l \mathbf{e}_{l\omega}(t), \quad (33)$$

and, by differentiation, we obtain

$$\begin{aligned} \dot{\mathcal{W}}_l(t) &= \dot{\mathbf{e}}_{lq}^\top(t) \mathbf{T}_{le}(t) k_{lq} \mathbf{J}_l \mathbf{e}_{l\omega}(t) \\ &\quad + \mathbf{e}_{lq}^\top(t) \dot{\mathbf{T}}_{le}(t) k_{lq} \mathbf{J}_l \mathbf{e}_{l\omega}(t) + \mathbf{e}_{lq}^\top(t) \mathbf{T}_{le}(t) k_{lq} \mathbf{J}_l \dot{\mathbf{e}}_{l\omega}(t). \end{aligned} \quad (34)$$

By inserting  $\dot{\mathbf{T}}_{le}^\top(t) \mathbf{e}_{lq}(t) = \mathbf{G}_l(t) \mathbf{e}_{l\omega}(t)$ , where

$$\mathbf{G}_l(t) = \frac{1}{2} [\tilde{\eta}_l(t) \mathbf{I} + \mathbf{S}(\tilde{\boldsymbol{\zeta}}_l(t))] - \frac{1}{4} \mathbf{I}, \quad (35)$$

and (21) and (24) into (34), we obtain

$$\begin{aligned} \dot{\mathcal{W}}_l &= \mathbf{e}_{l\omega}^\top(t) \mathbf{T}_{le}^\top(t) \mathbf{T}_{le}(t) k_{lq} \mathbf{J}_l \mathbf{e}_{l\omega}(t) + \mathbf{e}_{l\omega}^\top \mathbf{G}_l^\top(t) k_{lq} \mathbf{J}_l \mathbf{e}_{l\omega}(t) \\ &\quad - \mathbf{e}_{lq}^\top(t) \mathbf{T}_{le}(t) k_{lq} [-\mathbf{S}(\mathbf{J}_l \boldsymbol{\omega}_{i,lb}^{lb}(t)) + k_{l\omega} \mathbf{I}] \mathbf{e}_{l\omega}(t) \\ &\quad - \mathbf{e}_{lq}^\top(t) \mathbf{T}_{le}(t) k_{lq}^2 \mathbf{T}_{le}^\top(t) \mathbf{e}_{lq}(t) \\ &\leq -c_4 \|\mathbf{e}_{lq}\|^2 + c_5 \|\mathbf{e}_{l\omega}\|^2 + \|\mathbf{e}_{lq}\| c_6 \|\mathbf{e}_{l\omega}\|, \end{aligned} \quad (36)$$

where  $c_4 = k_{lq}^2/4$ ,  $c_5 = k_{lq} j_{lM} \|\mathbf{G}_l^\top(t)\|$ ,

$$c_6 = \frac{k_{lq}}{2(\|\mathbf{S}(\mathbf{J}_l \boldsymbol{\omega}_{i,lb}^{lb}(t))\| + k_{l\omega})}, \quad (37)$$

and  $\|\mathbf{G}_l^\top(t)\| \leq 3/4$ . The last term of (36) can be rewritten as

$$\|\mathbf{e}_{lq}\| c_6 \|\mathbf{e}_{l\omega}\| \leq \kappa \|\mathbf{e}_{l\omega}(t)\|^2 + \frac{c_6^2}{\kappa} \|\mathbf{e}_{lq}(t)\|^2, \quad (38)$$

and by choosing  $\kappa \gg 1$  such that  $c_4 > 2c_6^2/\kappa$ , we obtain

$$\dot{\mathcal{W}}_l(t) \leq - \frac{c_4}{2\|\mathbf{e}_{lq}(t)\|^2 + (c_5 + \kappa)\|\mathbf{e}_{l\omega}(t)\|^2}. \quad (39)$$

By applying the same line of arguments as in (29)–(31) and inequality (32), (39) may be expressed as

$$\mathcal{W}_l(t_0) + (c_5 + \kappa) c_3 \|\mathbf{x}_2(t_0)\|^2 \geq \frac{c_4}{2} \int_{t_0}^t \|\mathbf{e}_{lq}(s)\|^2 ds. \quad (40)$$

By inserting the upper bound

$$\begin{aligned} \|\mathcal{W}_l(t_0)\| &\leq \frac{1}{2} \max\{k_{lq}, j_{lM}\} \|\mathbf{e}_{lq}(t_0)\| \|\mathbf{e}_{l\omega}(t_0)\| \\ &\leq \max\{k_{lq}, j_{lM}\} \left( \|\mathbf{e}_{lq}(t_0)\|^2 + \|\mathbf{e}_{l\omega}(t_0)\|^2 \right) \\ &\leq \max\{k_{lq}, j_{lM}\} \|\mathbf{x}_2(t_0)\|^2 \end{aligned} \quad (41)$$

into (40), the expression may be written as

$$\int_{t_0}^t \left\| \mathbf{e}_{lq}(s) \right\|^2 ds \leq c_7 \|\mathbf{x}_2(t_0)\|^2, \quad (42)$$

where  $c_7 = 2(1/2 \max\{k_{lq}, j_{lM}\} + (c_5 + \kappa)c_3)/c_4$ .

We conclude from Lemma 3 of [41] that the origin is uniformly exponentially stable.

**3.3. Control of Follower Spacecraft.** For control of the follower spacecraft attitude, we incorporate a similar control law as in Section 3.2 but add terms for synchronization. For the control law, it is assumed that we have available information of the attitude and angular velocity for both spacecraft and inertia matrix,  $\mathbf{J}_f$ , and that all perturbations are known and accounted for. In the following, it is assumed that the equilibrium point is chosen such that  $\mathbf{e}_{fq\pm} = [1 \mp \tilde{\eta}_f, \tilde{\boldsymbol{\epsilon}}_f^\top]^\top$  is either the positive or negative equilibrium point and does not change during the maneuver. The control law is expressed as

$$\begin{aligned} \boldsymbol{\tau}_{fa}^{fb} &= \mathbf{J}_f \dot{\boldsymbol{\omega}}_{i,d}^{fb} - \mathbf{S}(\mathbf{J}_f \boldsymbol{\omega}_{i,fb}^{fb}) \boldsymbol{\omega}_{i,d}^{fb} \\ &\quad - k_{fq} (\mathbf{T}_{fe}^\top \mathbf{e}_{fq} - \mathbf{T}_{le}^\top \mathbf{e}_{lq}) - k_{f\omega} (\mathbf{e}_{f\omega} - \mathbf{e}_{l\omega}) - \boldsymbol{\tau}_{fd}^{fb}, \end{aligned} \quad (43)$$

where  $k_{fq} > 0$  and  $k_{f\omega} > 0$  are feedback gain scalars and the last two terms are for synchronization. By insertion of (43) into (9), we obtain the closed-loop rotational dynamics

$$\begin{aligned} \dot{\mathbf{e}}_{f\omega} &= \mathbf{J}_f^{-1} \left[ - \left( k_{f\omega} - \mathbf{S}(\mathbf{J}_f \boldsymbol{\omega}_{i,fb}^{fb}) \right) \mathbf{e}_{f\omega} \right. \\ &\quad \left. - k_{fq} \mathbf{T}_{fe}^\top \mathbf{e}_{fq} + k_{fq} \mathbf{T}_{le}^\top \mathbf{e}_{lq} + k_{f\omega} \mathbf{e}_{l\omega} \right] \\ &= f(\mathbf{x}_1) + g(\mathbf{x}) \mathbf{x}_2, \end{aligned} \quad (44)$$

where  $f(\mathbf{x}_1)$  is similar to the closed-loop system derived in Section 3.2 hence the proof of uniform asymptotic stability follows along similar lines.

The interconnection function is

$$g(\mathbf{x}) = \begin{bmatrix} \mathbf{J}_f^{-1} k_{fq} \mathbf{T}_{le}^\top & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_f^{-1} k_{f\omega} \end{bmatrix}. \quad (45)$$

In what follows we show that Assumptions 1–3 hold and, hence, that the origin of the closed-loop system is uniformly asymptotically stable.

*Proof of Assumption 1.* We start by considering  $\delta' = 1$  (where in this case  $\delta'$  is the parameter given in Assumption 1). By evaluating (14) on  $V_f$ , we obtain

$$\begin{aligned} \left\| \mathbf{e}_{f\omega}^\top \mathbf{J}_f + \mathbf{e}_{fq}^\top k_{fq} \right\| \|\mathbf{x}_1\| &\leq c_1 V_f(\mathbf{x}_1) \\ &= \frac{c_1}{2} \left[ \mathbf{e}_{f\omega}^\top \mathbf{J}_f \mathbf{e}_{f\omega} + \mathbf{e}_{fq}^\top k_{fq} \mathbf{e}_{fq} \right], \end{aligned} \quad (46)$$

and by applying the triangular inequality on the left side of (46) and squaring, we obtain

$$\begin{aligned} \left( \left\| \mathbf{e}_{f\omega}^\top \mathbf{J}_f \right\| + \left\| \mathbf{e}_{fq}^\top k_{fq} \right\| \right)^2 \|\mathbf{x}_1\|^2 \\ \leq \frac{c_1^2}{4} \left[ \mathbf{e}_{f\omega}^\top \mathbf{J}_f \mathbf{e}_{f\omega} + \mathbf{e}_{fq}^\top k_{fq} \mathbf{e}_{fq} \right]^2. \end{aligned} \quad (47)$$

On the left side of (47), we apply  $\|x\|^2 + \|y\|^2 + 2\|x\|\|y\| \leq 2(\|x\|^2 + \|y\|^2)$  and  $j_{fm} \leq \|\mathbf{J}_f\| \leq j_{fM}$  such that

$$\begin{aligned} \left( \left\| \mathbf{e}_{f\omega}^\top \mathbf{J}_f \right\| + \left\| \mathbf{e}_{fq}^\top k_{fq} \right\| \right)^2 \\ \leq 2\varphi \left( \mathbf{e}_{f\omega}^\top \mathbf{J}_f \mathbf{e}_{f\omega} + \mathbf{e}_{fq}^\top k_{fq} \mathbf{e}_{fq} \right), \end{aligned} \quad (48)$$

where  $\varphi = \max\{j_{fM}, k_{fq}\}$ . By insertion of (48) in (47), we obtain

$$2\varphi \left( \mathbf{e}_{f\omega}^\top \mathbf{e}_{f\omega} + \mathbf{e}_{fq}^\top \mathbf{e}_{fq} \right) \leq \frac{c_1^2 \rho}{4} \left( \mathbf{e}_{f\omega}^\top \mathbf{e}_{f\omega} + \mathbf{e}_{fq}^\top \mathbf{e}_{fq} \right), \quad (49)$$

where  $\rho = \min\{j_{fm}, k_{fq}\}$ , and thus, we have to choose

$$c_1 \geq \sqrt{\frac{8\varphi}{\rho}} \quad (50)$$

to fulfill (14). For (15), we have that

$$\left\| \mathbf{e}_{f\omega}^\top \mathbf{J}_f + \mathbf{e}_{fq}^\top k_{fq} \right\| \leq c_2, \quad (51)$$

and by using the triangular inequality and squaring and applying (48) we obtain

$$2\varphi^2 \|\mathbf{x}_1\|^2 \leq c_2^2. \quad (52)$$

Since  $\|\mathbf{x}_1\| \leq 1$ , we have to choose

$$c_2 \geq \sqrt{2}\varphi \quad (53)$$

to fulfill (15), and thus Assumption 1 is fulfilled.  $\square$

*Proof of Assumption 2.* Since  $\|\mathbf{T}_{le}\| = 1/2$ , (45) obviously fulfills the growth rate criteria of (16), such as

$$\begin{aligned} \|g(\mathbf{x})\| &\leq \frac{1}{2} \left[ \left( \frac{1}{J_x^2} + \frac{1}{J_y^2} + \frac{1}{J_z^2} \right) (k_{fq}^2 + 4k_{f\omega}^2) \right]^{1/2} \\ &= \xi_1, \end{aligned} \quad (54)$$

where  $\xi_1$  is constant, and thus Assumption 2 is fulfilled.  $\square$

*Proof of Assumption 3.* This assumption holds because  $\|\mathbf{x}_2(t)\|$  converges to zero exponentially fast.

We conclude that the equilibrium point  $(\mathbf{e}_{lq}, \mathbf{e}_{l\omega}, \mathbf{e}_{fq}, \mathbf{e}_{f\omega}) = (\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0})$  of the cascaded system is UAS.

The proof for the other equilibria follows *mutatis mutandis*.  $\square$

## 4. Robustness to Disturbances

In this section, we develop our results from Section 3 by introducing unknown bounded disturbances. We use a control law reminiscent of the Slotine and Li controller for manipulators; see [23], based on a control structure which has often been shown to be favorable from a stability analysis point of view. In the case of bounded additive nonvanishing disturbances, a steady-state error is unavoidable; hence, only practical asymptotic stability may be obtained. Although the control approach is the same, the technical analysis tools are more sophisticated. We refer the reader to [32].

**4.1. Control of Leader Spacecraft.** We start by assuming that we have available controller gains according to  $\theta_1 = [k_{fq}, k_{f\omega}]$ ,  $\theta_2 = [k_{lq}, k_{l\omega}] \in \Theta_1 = \Theta_2 = \mathbb{R}_{>0}^2$ . The uniform asymptotic stability in Section 3.2 is a result of the assumption that  $\tau_{ld}^{lb}$  is known and accounted for in the control law. We relax this assumption and assume that  $\tau_{ld}^{lb}$  is unknown, but bounded, and thus there exists a  $\beta_l > 0$  such that  $\|\tau_{ld}^{lb}\| \leq \beta_l$ . Note that many of the disturbances for Earth-orbiting spacecraft can be reasonably well modeled as  $\hat{\mathbf{f}}_{sd}^{sb}$ , which can be added to the overall analysis such that  $\hat{\mathbf{f}}_{sd}^{sb} = \mathbf{f}_{sd}^{sb} - \hat{\mathbf{f}}_{sd}^{sb}$ . This strategy can reduce the upper bound such that  $\|\hat{\mathbf{f}}_{sd}^{sb}(t)\| < \beta_{\tilde{s}} < \beta_s$ , based on the quality of the disturbance modeling. We apply the control law

$$\tau_{la}^{lb} = \mathbf{J}_l \dot{\omega}_{i,r}^{lb} - \mathbf{S}(\mathbf{J}_l \omega_{i,r}^{lb}) \omega_{i,r}^{lb} - k_{lq} \mathbf{T}_{le}^T \mathbf{e}_{lq} - k_{l\omega} \mathbf{s}_l, \quad (55)$$

$$\omega_{i,r}^{lb} = \omega_{i,d}^{lb} - \mathbf{\Gamma}_l \mathbf{T}_{le}^T (\mathbf{e}_{lq}) \mathbf{e}_{lq}, \quad (56)$$

$$\mathbf{s}_l = \omega_{i,lb}^{lb} - \omega_{i,r}^{lb} = \mathbf{e}_{l\omega} + \mathbf{\Gamma}_l \mathbf{T}_{le}^T \mathbf{e}_{lq}, \quad (57)$$

where  $k_{lq} > 0$ ,  $k_{l\omega} > 0$  and  $\mathbf{\Gamma}_l = \mathbf{\Gamma}_l^T > 0$  are feedback gains, and, by inserting (55) into (9), we obtain the closed-loop dynamics

$$\dot{\mathbf{s}}_l = \mathbf{J}_l^{-1} [\mathbf{S}(\mathbf{J}_l \omega_{i,lb}^{lb}) \mathbf{s}_l - k_{lq} \mathbf{T}_{le}^T \mathbf{e}_{lq} - k_{l\omega} \mathbf{s}_l]. \quad (58)$$

A radial unbounded, positive definite Lyapunov function candidate is defined as

$$\tilde{V}_l = \frac{1}{2} (\mathbf{s}_l^T \mathbf{J}_l \mathbf{s}_l + \mathbf{e}_{lq}^T k_{lq} \mathbf{e}_{lq}) > 0 \quad \forall \mathbf{s}_l \neq \mathbf{0}, \mathbf{e}_{lq} \neq \mathbf{0}, \quad (59)$$

and, by differentiation of (59) and insertion of (58), we obtain

$$\begin{aligned} \dot{\tilde{V}}_l &= -\mathbf{s}_l^T k_{l\omega} \mathbf{s}_l - \mathbf{e}_{lq}^T \mathbf{T}_{le} \mathbf{\Gamma}_l k_{lq} \mathbf{T}_{le}^T \mathbf{e}_{lq} + \mathbf{s}_l^T \tau_{ld}^{lb} \\ &= -\chi_2^T \mathbf{Q}_l \chi_2 + \mathbf{s}_l^T \tau_{ld}^{lb} \\ &\leq -q_{lm} \|\chi_2\|^2 + \beta_l \|\chi_2\|, \end{aligned} \quad (60)$$

where  $\chi_2 = [\mathbf{e}_{lq}^T, \mathbf{s}_l^T]^T$ ,  $\mathbf{Q}_l = \text{diag}\{\mathbf{T}_{le} k_{lq} \mathbf{\Gamma}_l \mathbf{T}_{le}^T, k_{l\omega} \mathbf{I}\}$ , and  $q_{lm} > 0$  is the smallest eigenvalue of  $\mathbf{Q}_l$ . Accordingly,  $\dot{\tilde{V}}_l < 0$  when  $\|\chi_2\| > \beta_l / q_{lm} = \delta_2$ , and, as  $\beta_l$  increases, it can be counteracted by increasing the controller gains. Hence, the perturbed system is uniformly practically asymptotically stable (UPAS); see [35]. We cannot claim semiglobal results for the same reasons as we cannot claim global results. Nevertheless, we assume that both  $\Delta_1$  and  $\Delta_2$  can be chosen arbitrary large to make it easier to follow the line of the proof.

**4.2. Control of Follower Spacecraft.** A similar disturbance as in Section 4.1 is introduced which are bounded such that  $\|\tau_{fd}^{fb}\| \leq \beta_f$ , and we apply a synchronizing controller based on the Slotine and Li structure such as

$$\begin{aligned} \tau_{fa}^{lb} &= \mathbf{J}_f \dot{\omega}_{i,r}^{fb} - \mathbf{S}(\mathbf{J}_f \omega_{i,r}^{fb}) \omega_{i,r}^{fb} - k_{fq} (\mathbf{T}_{ef}^T \mathbf{e}_{fq} - \mathbf{T}_{le}^T \mathbf{e}_{lq}) \\ &\quad - k_{f\omega} (\mathbf{s}_f - \mathbf{s}_l), \end{aligned} \quad (61)$$

where  $\omega_{i,r}^{fb}$  and  $\mathbf{s}_f$  are defined similar to (56)-(57). By inserting (61) into (9), we obtain

$$\dot{\mathbf{s}}_f = \tilde{f}(\chi_1) + \tilde{g}(\chi) \chi_2, \quad (62)$$

where  $\chi_1 = [\mathbf{e}_{fq}^T, \mathbf{s}_f^T]^T$  and  $\chi = [\chi_1^T, \chi_2^T]^T$ , and  $\tilde{f}(\chi_1)$  and  $\tilde{g}(\chi)$  are on the same form as (58) and (45), respectively.

The rest of the proof consists in verifying the conditions of [32, Theorem 3.3].

*Proof of Assumption 3.2 (see [32]).* The function  $\tilde{g}(\chi) \chi_2$  is uniformly bounded both in time and in  $\theta_2$  and vanishes with  $\chi_2$ ; thus, for any  $\theta_1 \in \Theta_1$ , we choose

$$G_{\theta_1}(\|\chi\|) = \frac{1}{2} \left[ \left( \frac{1}{J_x^2} + \frac{1}{J_y^2} + \frac{1}{J_z^2} \right) (k_{fq}^2 + 4k_{f\omega}^2) \right]^{1/2}, \quad (63)$$

$$\Psi_{\theta_1}(\|\chi_2\|) = \|\chi_2\|,$$

thus,  $G_{\theta_1}(\|\chi\|)$  is constant and  $\Psi_{\theta_1}(\|\chi_2\|)$  is of class  $\mathcal{K}_\infty$ , and the assumption is fulfilled for all  $\theta_2 \in \Theta_2$  and all  $\chi \in \mathcal{S}^3 \times \mathbb{R}^3 \times \mathcal{S}^3 \times \mathbb{R}^3$ .  $\square$

*Proof of Assumption 3.4 (see [32]).* This Assumption was proven in Section 4.1.  $\square$

*Proof of Assumption 3.5 (see [32]).* By applying the same reasoning as in Section 4.1, we achieve  $\dot{\tilde{V}}_f < 0$  when  $\|\chi_1\| > \beta_f / q_{fm} = \delta_1$ , where  $q_{fm} > 0$  is the smallest eigenvalue of  $\mathbf{Q}_f = \text{diag}\{\mathbf{T}_{fe} k_{fq} \mathbf{\Gamma}_f \mathbf{T}_{fe}^T, k_{f\omega} \mathbf{I}\}$ , which is defined similar to  $\mathbf{Q}_l$  in (60). An increase of  $\beta_f$  can as well be counteracted by increasing the controller gains; thus, given any  $\Delta_1 > \delta_1 > 0$ , there exists a parameter  $\theta_1^*(\delta_1, \Delta_1) \in \Theta_1$ . We choose  $\underline{\alpha}_{\delta_1, \Delta_1}(\|\chi_1\|) = 1/2 \min\{j_{fm}, k_{fq}\} \|\chi_1\|^2$  and  $\bar{\alpha}_{\delta_1, \Delta_1}(\|\chi_1\|) = 1/2 \max\{j_{fM}, k_{fq}\} \|\chi_1\|^2$ , and thus the first part of the assumption is fulfilled, and the second part is fulfilled for  $\alpha_{\delta_1, \Delta_1}(\|\chi_1\|) = \min\{k_{fq}/4, k_{f\omega}\} \|\chi_1\|^2$ . The last inequality also holds similar to (51)–(53) with  $c_{\delta_1, \Delta_1}(\|\chi_1\|) = \sqrt{2}\sigma \|\chi_1\|^{1/2}$ , and thus Assumption 3.5 holds for all  $\chi_1 \in \mathcal{H}(\delta_1, \Delta_1)$ , where  $\Delta_1$  can be chosen arbitrary large by assumption.  $\square$

*Proof of Assumption 3.6 (see [32]).* We define a LFC for the leader and follower spacecraft as

$$\tilde{V}_{lf}(t, x) = \frac{1}{2} (\mathbf{s}_l^T \mathbf{J}_l \mathbf{s}_l + \mathbf{e}_{lq}^T k_{lq} \mathbf{e}_{lq} + \mathbf{s}_f^T \mathbf{J}_f \mathbf{s}_f + \mathbf{e}_{fq}^T k_{fq} \mathbf{e}_{fq}), \quad (64)$$

which is lower and upper bounded by

$$\underline{\alpha}_{lf} = \frac{1}{2} \min\{j_{lm}, j_{fm}, k_{lq}, k_{fq}\} \|\chi\|^2, \quad (65)$$

$$\bar{\alpha}_{lf} = \frac{1}{2} \max\{j_{lM}, j_{fM}, k_{lq}, k_{fq}\} \|\chi\|^2, \quad (66)$$

respectively, and it can be shown that the second equation of [32, Proposition 2.13] is fulfilled by differentiation of (64). There exists a positive constant  $\Delta_0$  such that for any given positive number  $\delta_1, \Delta_1, \delta_2, \Delta_2$ , satisfying  $\Delta_1 > \max\{\delta_1, \Delta_0\}$

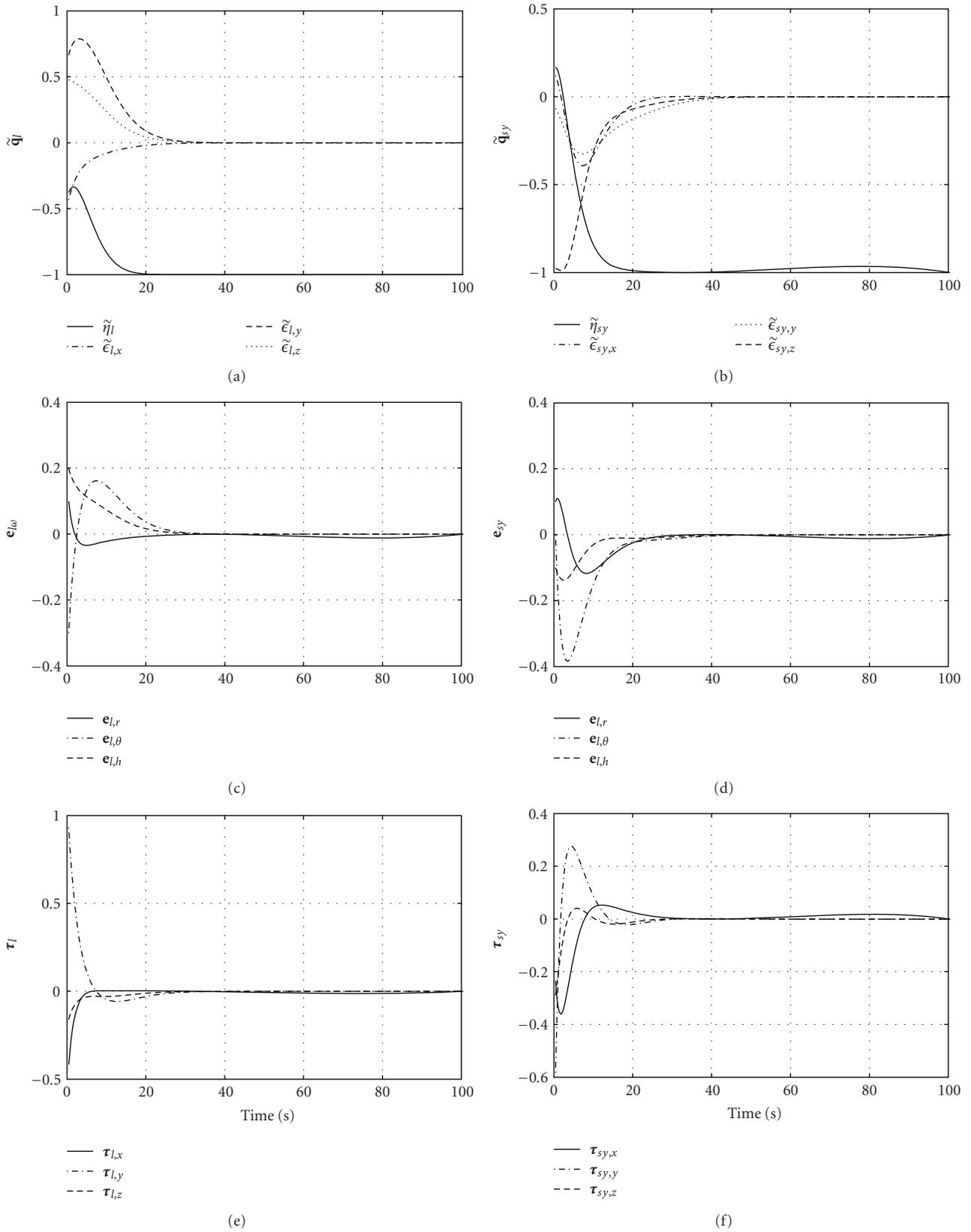


FIGURE 2: Attitude and angular velocity error and control torque for leader ((a), (c), (e)) and synchronizing follower ((b), (d), (f)) spacecraft during maneuver without measurement noise and disturbances.

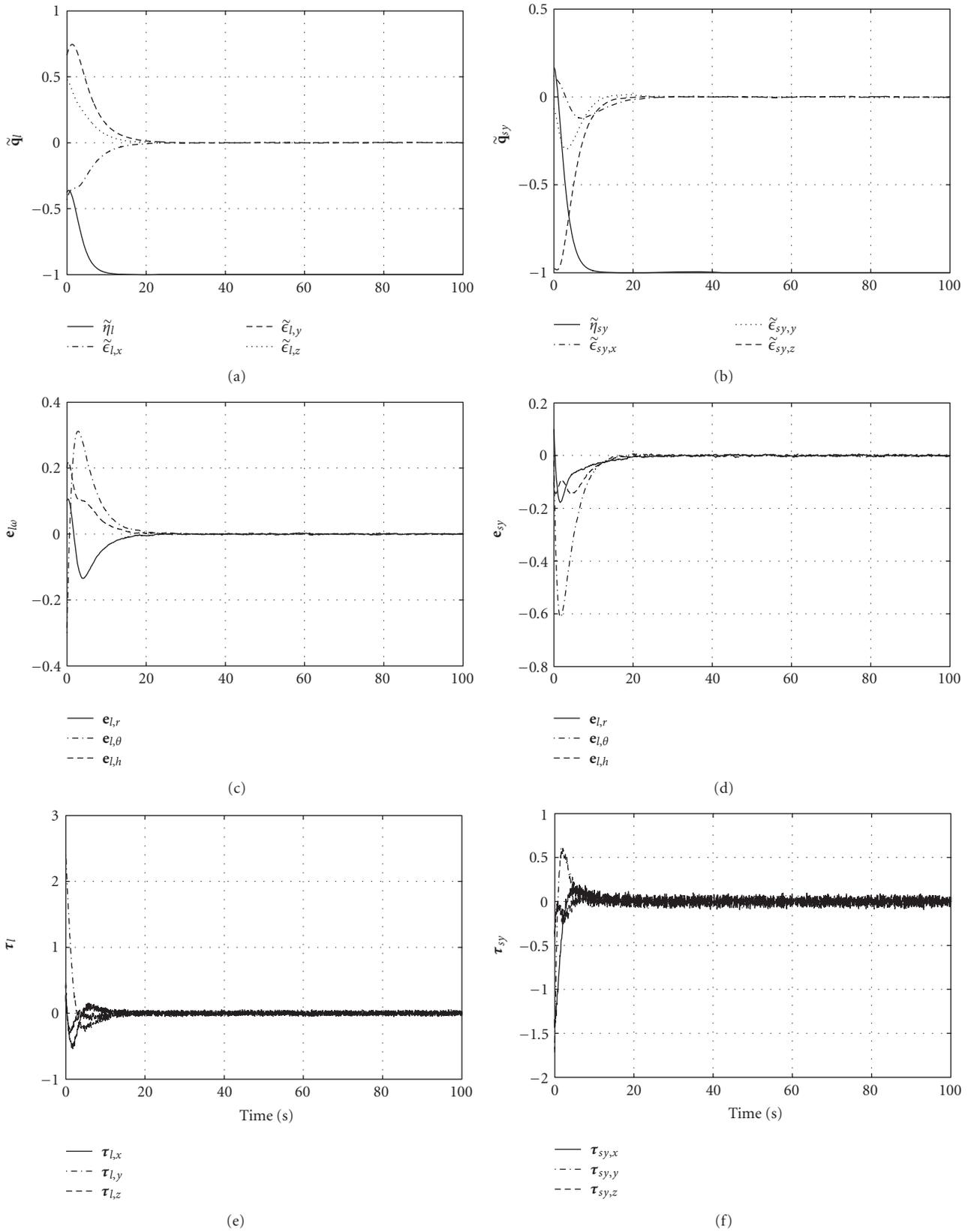


FIGURE 3: Attitude and angular velocity error and control torque for leader ((a), (c), (e)) and synchronizing follower ((b), (d), (f)) spacecraft during maneuver including measurement noise and disturbances.

and  $\Delta_2 > \delta_2$ , there exists a  $\delta_1$  such that  $\bar{\alpha}_{lf}(\delta_1) < \underline{\alpha}_{lf}(\Delta_1)$ . As mentioned in Section 4.1, given any  $\beta_l$ , we can achieve any  $\delta_2$  by increasing the gains  $\theta_2$ , and thus there exists a parameter  $\theta_2^* \in \mathcal{D}_{f_2}(\delta_2, \Delta_2) \cap \Theta_2$ , such that by applying the last inequality of [32, Proposition 2.13] using the bounds (65), we see that the first equation of Assumption 3.6 is fulfilled for

$$\gamma(\Delta_1) = \sqrt{\frac{\min\{j_{lm}, j_{fm}, k_{lq}, k_{fq}\} \Delta_1^2}{\max\{j_{lM}, j_{fM}, k_{lq}, k_{fq}\}}}. \quad (67)$$

We have that

$$\underline{\alpha}_{\delta_1, \Delta_1}^{-1} \circ \bar{\alpha}_{\delta_1, \Delta_1}(\delta_1) = \sqrt{\frac{\max\{j_{fM}, k_{fq}\} \delta_1^2}{\min\{j_{fm}, k_{fq}\}}}, \quad (68)$$

then, for any  $\Delta_* > \delta_* > 0$ , there exist parameters  $\delta_1$ ,  $\delta_2$ ,  $\Delta_1$ , and  $\Delta_2$  such that

$$\min\left\{\Delta_1, \Delta_2, \sqrt{\frac{\min\{j_{lm}, j_{fm}, k_{lq}, k_{fq}\} \Delta_1^2}{\max\{j_{lM}, j_{fM}, k_{lq}, k_{fq}\}}}\right\} \geq \Delta_* \quad (69)$$

since  $\Delta_1$  and  $\Delta_2$  can be chosen arbitrarily large and the constants  $j_{sm}, j_{sM}, k_{sq}, k_{sw} > 0$ , and

$$\max\left\{\delta_2, \sqrt{\frac{\max\{j_{fM}, k_{fq}\} \delta_1^2}{\min\{j_{fm}, k_{fq}\}}}\right\} \leq \delta_* \quad (70)$$

is fulfilled since, by decreasing  $\delta_1$ ,  $k_{fq}$  is increased but only of order one, and thus the two last inequalities of Assumption 3.6 are fulfilled and Assumption 3.6 holds. It can then be concluded that the equilibrium points of the cascaded system are UPAS.

By setting  $\mathbf{e}_{lq} = \mathbf{e}_{lq-}$  or  $\mathbf{e}_{fq} = \mathbf{e}_{fq-}$ , the other three proofs are performed in a similar way, and we thus conclude that the set of equilibrium points  $(\mathbf{e}_{lq\pm}, \mathbf{e}_{l\omega}, \mathbf{e}_{fq\pm}, \mathbf{e}_{f\omega}) = (\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0})$  are UPAS.  $\square$

## 5. Simulations

In the following, simulation results of a leader spacecraft in an elliptic LEO with the follower spacecraft following the same orbit with one-second delay are presented to validate the proposed approach. The simulations were performed in Simulink using a variable sample-time Runge-Kutta ODE45 solver with relative and absolute tolerance of  $10^{-9}$ . The moments of inertia were chosen as  $\mathbf{J}_l = \mathbf{J}_f = \text{diag}\{4.35, 4.33, 3.664\}$  kgm<sup>2</sup>, and the spacecraft orbits were chosen with apogee at 750 km, perigee at 600 km, inclination at 71°, and the argument of perigee and the right ascension of the ascending node at 0°. The initial conditions were chosen as  $\mathbf{q}_l(t_0) = [-0.3772, -0.4329, 0.6645, 0.4783]^\top$ ,  $\mathbf{q}_f(t_0) = 1/\sqrt{4}[1, 1, 1, 1]^\top$ ,  $\boldsymbol{\omega}_l(t_0) = [0.1 - 0.30.2]^\top$ , and  $\boldsymbol{\omega}_f(t_0) = [0.2 - 0.30.1]^\top$ , controller gains as  $k_{sq} = 1$ ,  $k_{sw} = 2$ , and  $\boldsymbol{\Gamma}_s = \mathbf{I}$  for (23), (43), (55), and (61), and desired trajectory as  $\dot{\boldsymbol{\omega}}_d = 0.1[-10c_0^2 \cos(8c_0t), 48c_0^2 \sin(16c_0t), -8c_0^2 \cos(4c_0t)]$ ,

$\boldsymbol{\omega}_d$  its integrate, and  $\mathbf{q}_d(t_0) = [1, \mathbf{0}]^\top$ , where  $c_0 = \pi/t_0$  and  $t_0$  denotes the orbital period. For relative attitude (synchronization error) and angular velocity between the leader and follower spacecraft, we define  $\tilde{\mathbf{q}}_{sy} = \mathbf{q}_f \otimes \bar{\mathbf{q}}_l$  and  $\mathbf{e}_{sy} = \mathbf{e}_{f\omega} - \mathbf{e}_{l\omega}$ , respectively.

In Figure 2, simulation results are presented without disturbances and noise. From Figure 2(a) we see that the leader spacecraft settles at the negative equilibrium, the angular velocity error go towards zero, and the actuator torque is presented in the bottommost plot. On the right-hand side we see that the relative attitude and angular velocity error go towards zero, and thus the follower spacecraft is able to synchronize with the leader spacecraft. The bottommost plot on the right hand depicts the follower actuator torque.

In the second set of simulation results, we introduce measurement noise as  $\sigma_{\mathbb{B}^n} = \{x \in \mathbb{R}^n : \|x\| \leq \sigma\}$  which was added according to  $\tilde{\mathbf{e}}_q = (\mathbf{e}_q + 0.05\mathbb{B}^4)/\|\mathbf{e}_q + 0.05\mathbb{B}^4\|$  and  $\tilde{\mathbf{e}}_\omega = \mathbf{e}_\omega + 0.01\mathbb{B}^3$  for both the leader and follower spacecraft. Also, since we are considering a slightly elliptic LEO, we only considered the disturbance torques which are the major contributors to these kind of orbits, that is, gravity gradient torque and torques caused by atmospheric drag and  $J_2$  effect (cf. [34, 36, 37]). The latter is caused by nonhomogeneous mass distribution of a planet. The torques generated by atmospheric drag and  $J_2$  were induced because of a 10 cm displacement of the center of mass. All disturbances are considered continuous and bounded.

As it can be seen from Figure 3, the control laws derived for unknown disturbances also are able to make the leader track the reference and make the follower synchronize with the leader, similar to the results from the first simulation. One notable difference is that these control laws are in general faster than the results presented in Figure 2, though demanding larger absolute control torque.

## 6. Conclusions

In this paper, we have presented control laws for leader-follower synchronization of spacecraft, performed stability analysis based on cascade theory, and proven the equilibrium points of the overall system to be uniformly asymptotically stable (UAS) when disturbances were assumed to be known, and uniformly practically asymptotically stable (UPAS) when unknown, but bounded disturbances were included. Simulation results were presented to validate the proposed method for the overall system showing that the follower spacecraft was able to synchronize with the leader spacecraft in a satisfactory way even when disturbances were presented.

## References

- [1] I. Blekhman, *Synchronization in Science and Technology*, ASME Press Translations, ASME Press, New York, NY, USA, 1988.
- [2] A. Rodriguez-Angeles, *Synchronization of mechanical systems*, Ph.D. thesis, Eindhoven University of Technology, Eindhoven, The Netherlands, 2002.
- [3] H. Nijmeijer and A. Rodriguez-Angeles, *Synchronization of Mechanical Systems*, vol. 46 of *World Scientific Series on Nonlinear Science, Series A*, 2003.

- [4] E. Kyrkjebø, *Motion coordination of mechanical systems: leader-follower synchronization of euler-lagrange systems using output feedback control*, Ph.D. thesis, Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway, 2007.
- [5] A. Rodriguez-Angeles and H. Nijmeijer, "Coordination of two robot manipulators based on position measurements only," *International Journal of Control*, vol. 74, no. 13, pp. 1311–1323, 2001.
- [6] A. K. Bondhus, K. Y. Pettersen, and H. Nijmeijer, "Master-slave synchronization of robot manipulators," in *Proceedings of the IFAC Symposium on Nonlinear Control Systems Design*, Stuttgart, Germany, 2004.
- [7] J. Lawton and R. W. Beard, "Elementary attitude formation maneuver via leader-following and behaviour-based control," in *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Denver, Colo, USA, 2000.
- [8] A. K. Bondhus, K. Y. Pettersen, and J. T. Gravdahl, "Leader/follower synchronization of satellite attitude without angular velocity measurements," in *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference (CDC-ECC '05)*, pp. 7270–7277, Seville, Spain, December 2005.
- [9] R. Kristiansen, *Dynamic synchronization of spacecraft—modeling and coordinated control of leader-follower spacecraft formations*, Ph.D. thesis, Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway, 2008.
- [10] E. I. Grøtli, *Robust stability and control of spacecraft formations*, Ph.D. thesis, Norwegian University of Science and Technology, 2010.
- [11] S. H. Fu, C. C. Cheng, and C. Y. Yin, "Nonlinear adaptive tracking control for underway replenishment process," in *Proceeding of IEEE International Conference on Networking, Sensing and Control*, vol. 2, pp. 707–712, Taipei, Taiwan, March 2004.
- [12] E. Kyrkjebø, E. Panteley, A. Chaillet, and K. Y. Pettersen, "A virtual vehicle approach to underway replenishment," in *Group Coordination and Cooperative Control*, K. Y. Pettersen, J. T. Gravdahl, and H. Nijmeijer, Eds., vol. 336 of *Lecture Notes in Control and Information Sciences*, pp. 171–189, Springer, Berlin, Germany, 2006.
- [13] J. R. Lawton, B. J. Young, and R. W. Beard, "Decentralized approach to elementary formation maneuvers," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA '00)*, vol. 3, pp. 2728–2733, April 2000.
- [14] G. Antonelli and S. Chiaverini, "Kinematic control of platoons of autonomous vehicles," *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1285–1292, 2006.
- [15] F. Arrichiello, *Coordination control of multiple mobile robots*, Ph.D. thesis, Cassino University, Cassino, Italy, 2006.
- [16] F. Arrichiello, S. Chiaverini, and T. I. Fossen, "Formation control of marine surface vessels using the null-space-based behavioral control," in *Group Coordination and Cooperative Control*, K. Y. Pettersen, J. T. Gravdahl, and H. Nijmeijer, Eds., vol. 336 of *Lecture Notes in Control and Information Sciences*, chapter 1, pp. 1–19, Springer, Berlin, Germany, 2006.
- [17] R. Cui, S. S. Ge, B. voon Ee How, and Y. Sang Choo, "Leader-follower formation control of underactuated autonomous underwater vehicles," *Ocean Engineering*, vol. 37, no. 17–18, pp. 1491–1502, 2010.
- [18] H. Bai, M. Arcak, and J. T. Wen, "Adaptive motion coordination: using relative velocity feedback to track a reference velocity," *Automatica*, vol. 45, no. 4, pp. 1020–1025, 2009.
- [19] A. Sarlette, R. Sepulchre, and N. E. Leonard, "Autonomous rigid body attitude synchronization," *Automatica*, vol. 45, no. 2, pp. 572–577, 2009.
- [20] D. V. Dimarogonas, P. Tsiotras, and K. J. Kyriakopoulos, "Leader-follower cooperative attitude control of multiple rigid bodies," *Systems and Control Letters*, vol. 58, no. 6, pp. 429–435, 2009.
- [21] R. Ortega, A. Loria, P. J. Nicklasson, and H. Sira-Ramírez, *Passivity-based Control of Euler-Lagrange Systems: Mechanical, Electrical and Electromechanical Applications*, Communications and Control Engineering, Springer, London, UK, 1998.
- [22] B. Paden and R. Panja, "Globally asymptotically stable 'PD+' controller for robot manipulators," *International Journal of Control*, vol. 47, no. 6, pp. 1697–1712, 1988.
- [23] J. J. E. Slotine and W. Li, "On the adaptive control of robot manipulators," *International Journal of Robotics Research*, vol. 6, no. 3, pp. 49–59, 1987.
- [24] F. Mazenc, L. Praly, and W. P. Dayawansa, "Global stabilization by output feedback: examples and counterexamples," *Systems and Control Letters*, vol. 23, no. 2, pp. 119–125, 1994.
- [25] A. Loria and E. Panteley, "Cascaded nonlinear time-varying systems: analysis and design," in *Advanced Topics in Control Systems Theory*, vol. 311 of *Lecture Notes in Control and Information Sciences*, chapter 2, pp. 23–64, Springer, Berlin, Germany, 2005.
- [26] T. I. Fossen and O.-E. Fjellstad, "Cascaded adaptive control of ocean vehicles with significant actuator dynamics," in *Proceedings of the IFAC World Congress*, Sydney, Australia, 1993.
- [27] A. Loria, H. Nijmeijer, and O. Egeland, "Cascaded synchronization of two pendula," in *Proceedings of the American Control Conference*, 1998.
- [28] M. Jankovic, R. Sepulchre, and P. V. Kokotovic, "Constructive Lyapunov stabilization of nonlinear cascade systems," *IEEE Transactions on Automatic Control*, vol. 41, no. 12, pp. 1723–1735, 1996.
- [29] Z. P. Jiang and I. M. Y. Mareels, "A small-gain control method for nonlinear cascaded systems with dynamic uncertainties," *IEEE Transactions on Automatic Control*, vol. 42, no. 3, pp. 292–308, 1997.
- [30] E. Panteley and A. Loria, "On global uniform asymptotic stability of nonlinear time-varying systems in cascade," *Systems and Control Letters*, vol. 33, no. 2, pp. 131–138, 1998.
- [31] A. Chaillet and A. Loria, "Uniform semiglobal practical asymptotic stability for non-autonomous cascaded systems and applications," *Automatica*, vol. 44, no. 2, pp. 337–347, 2008.
- [32] A. Chaillet, *On stability and robustness of nonlinear systems: applications to cascaded systems*, Ph.D. thesis, UFR Scientifique D'Orsay, Paris, France, 2006.
- [33] O. Egeland and J. T. Gravdahl, *Modeling and Simulation for Automatic Control*, Marine Cybernetics, Trondheim, Norway, 2002.
- [34] M. J. Sidi, *Spacecraft Dynamics and Control*, Cambridge University Press, New York, NY, USA, 1997.
- [35] W. Hahn, *Stability of Motion*, Springer, Berlin, Germany, 1967.
- [36] O. Montenbruck and E. Gill, *Satellite Orbits. Models, Methods, Applications*, Springer, Berlin, Germany, 1st edition, 2001.
- [37] J. R. Wertz, Ed., *Spacecraft Attitude Determination and Control*, Kluwer Academic Publishers, London, UK, 1978.
- [38] Ø. Hegrenæs, J. T. Gravdahl, and P. Tøndel, "Spacecraft attitude control using explicit model predictive control," *Automatica*, vol. 41, no. 12, pp. 2107–2114, 2005.

- [39] T. I. Fossen, *Marine Control Systems: Guidance, Navigation, and Control of Ships, Rigs and Underwater Vehicles*, Marine Cybernetics, Trondheim, Norway, 2002.
- [40] J. T. Y. Wen and K. Kreutz-Delgado, "The attitude control problem," *IEEE Transactions on Automatic Control*, vol. 36, no. 10, pp. 1148–1162, 1991.
- [41] A. Loria, E. Panteley, D. Popovic, and A. R. Teel, " $\delta$ -persistence of excitation: a necessary and sufficient condition for uniform attractivity," in *Proceedings of the 41st IEEE Conference on Decision and Control*, vol. 3, pp. 3506–3511, Las Vegas, Nev, USA, December 2002.

## Research Article

# Design of an Extended Interacting Multiple Models Adaptive Estimator for Attitude Determination of a Stereoimagery Satellite

**Hossein Bolandi, Farhad Fani Saberi, and Amir Mehrjardi Eslami**

*Electrical Engineering Department, Iran University of Science and Technology (IUST), Narmak, Tehran 16846-13114, Iran*

Correspondence should be addressed to Farhad Fani Saberi, farhad\_fanisaberi@ee.iust.ac.ir

Received 26 January 2011; Revised 28 April 2011; Accepted 6 June 2011

Academic Editor: Yu Gu

Copyright © 2011 Hossein Bolandi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We will design an extended interacting multiple models adaptive estimator (EIMMAE) for attitude determination of a stereoimagery satellite. This algorithm is based on interacting multiple models (IMM) extended kalman filters (EKF) using star sensor and gyroscope data. In this method, the motion of satellite during stereoimaging manoeuvres is partitioned into two different modes: “manoeuvring motion” mode and “uniform motion” mode. The proposed algorithm will select the suitable Kalman filter structure to estimate gyro errors accurately in order to maintain the peak attitude estimation error less enough at the beginning of manoeuvres while the satellite is in “manoeuvring motion” mode and then will select the suitable star sensor measurement noise level at the end of manoeuvres while the satellite is in “uniform motion” mode to reduce attitude estimation errors. It will be shown that using the proposed algorithm, the attitude estimation accuracy of stereoimagery satellite will be increased considerably. The effectiveness of the proposed algorithm will be examined and compared with the previous proposed methods through numerical simulations.

## 1. Introduction

Data collection in stereo mode is the simplest and most convenient way for 3D topographic data acquisition to produce new and revision of old inaccurate databases and maps which has been matured over the 100 years [1–4]. This methodology involves identifying and measuring targets on images of an object which have been taken from disparate viewpoints. These images are then used to compute three dimensional coordinates of the locations of the object.

In [5], it has been suggested an effective stereoimaging scenario to obtain the advantages of previous proposed methods. In this scenario, state-of-the-art solution is to control pitch and roll axes of satellite simultaneously in such a way that short “revisit period” and “repeat cycle” can be obtainable according to Figure 1. Therefore, it needs highly accurate and stable pointing maneuvers to be accomplished in a few seconds that require the satellite to rotate along a relatively large angle attitude very fast.

To achieve the above-mentioned scenario, an accurate and fast attitude determination system is crucial to provide precise attitude knowledge for successful satellite operations.

So, we consider two basic modes of flight for achieving high accuracy attitude determination task during each of stereoimaging maneuvers: “uniform motion” mode with constant angular velocity while the satellite reaches the end of stereoimaging maneuvers and “maneuvering motion” mode with high angular velocity at the beginning of stereoimaging maneuvers [6]. Therefore, a suitable attitude determination system for the stereoimagery satellite has to meet the following requirements properly in the two above-mentioned modes [6]: maintaining the peak attitude estimation errors less enough while the satellite is in the “maneuvering motion” mode and reducing the attitude estimation errors at the end of maneuvers while imaging takes place.

Difficulties associated with high-accuracy attitude estimation techniques to satisfy the above-mentioned requirements are due to the inherent nonlinearities of satellite dynamic model [7–10], estimation of gyroscope errors, and identifying of star sensor measurement noise levels which strongly affect the performance of the attitude estimation system at the end of maneuvers [6].

In order to deal with the above-mentioned problems, several approaches have been proposed [11–14]. In [15],

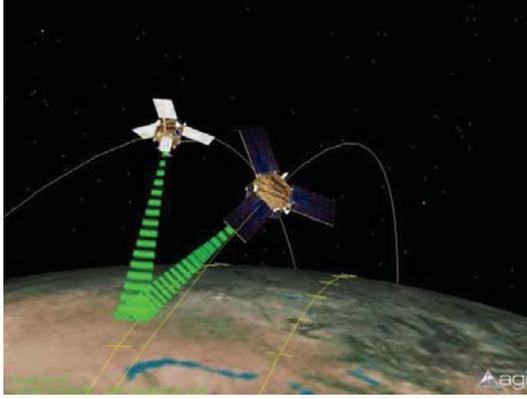


FIGURE 1: Stereoimaging Scenario.

a traditional 6-state attitude determination filter, containing three star attitude errors and three gyro bias errors, has been proposed. But, in high-rate maneuvering operating conditions (e.g., “maneuvering motion” mode), the ability of this filter to maintain attitude knowledge performance is degraded by gyro scale factor and misalignment errors. To solve this problem, a 15-state Kalman filter (containing three star attitude errors, three gyro bias errors, and nine gyro error parameters) has been proposed in [11] to achieve attitude estimation and gyro calibration in high-rate maneuvering operating conditions. Drawback associated with this structure is that in “uniform motion” mode, the 15-state filter behaves no differently from a 6-state filter in providing attitude estimation and gyro bias correction. Therefore, a multirate profile attitude estimation structure has been designed in [16, 17] based on star sensor and gyroscope measurements data. This structure increases the estimation convergence rate of gyro bias and misalignment for large and fast maneuvers and will maintain the peak attitude estimation errors less enough while the satellite is in the “maneuvering motion” mode. But, this structure cannot improve attitude estimation accuracy at the end of maneuvers while the satellite is in “uniform motion” mode.

To circumvent the aforementioned problems in order to maintain the peak attitude estimation errors less enough in the “maneuvering motion” mode and to reduce the attitude estimation errors in the “uniform motion” mode, in this paper, we will design a new adaptive attitude estimation structure based on interacting multiple models (IMM) using star sensor and gyroscope measurements data. The proposed structure consists of two different IMM estimator structures which are called: “IMM\_CT” and “IMM\_L”. The “IMM\_CT” structure will be selected automatically based on the model conditional likelihood functions to estimate the gyro errors in order to maintain the peak attitude estimation errors less enough while the satellite is in “maneuvering motion” mode. The “IMM\_L” structure will be selected automatically to identify the suitable star sensor measurement noise level to reduce attitude estimation errors while the satellite is in “uniform motion” mode. It will be shown that by using the proposed adaptive attitude estimation method, the attitude estimation requirements of stereoimagery satellite will be

satisfied simultaneously. The effectiveness of the proposed algorithm method will be examined and compared with previous proposed methods through numerical simulations.

## 2. Mathematical Model

The satellite being studied here is assumed to be rigid and equipped with a three-axis gyro, a star sensor, and four reaction wheels.

*2.1. Dynamic and Kinematic Model of Satellite.* The nonlinear kinematic equations of a satellite can be written as follows [18]:

$$\begin{aligned}\omega_x &= \dot{\phi} - \dot{\psi} \sin \theta - \omega_o \cos \theta \sin \psi, \\ \omega_y &= \dot{\theta} \cos \phi + \dot{\psi} \cos \theta \sin \phi \\ &\quad - \omega_o (\cos \phi \cos \psi + \sin \phi \sin \theta \sin \psi), \\ \omega_z &= \dot{\psi} \cos \theta \cos \phi - \dot{\theta} \sin \phi \\ &\quad - \omega_o (-\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi),\end{aligned}\quad (1)$$

where  $\omega = [\omega_x \ \omega_y \ \omega_z]$  are the angular velocity components of the body frame,  $\omega_o$  is orbit angular velocity, and  $\phi, \theta, \psi$  are the roll, pitch, and yaw angles, respectively. Motion dynamic equations of a rigid satellite can be written as

$$J\dot{\omega} = \tau + \omega \times h - u_w \quad (2)$$

$J$  denotes the inertia tensor of satellite.  $u_w$  and  $\tau$  are the reaction wheels control torque vector and external torque inputs, respectively.  $h = [h_x \ h_y \ h_z]^T$  is total satellite angular momentum in the body axes and defined as follows:

$$h = J\omega + Ch_a. \quad (3)$$

The angular momentum vector of the wheels  $h_a$  is defined as follows [19]:

$$h_a = I_w C^T \omega + I_w \omega_w \quad (4)$$

$I_w$  is moment of inertial matrix for the wheels and  $\omega_w$  is wheels speeds matrix. Moreover,

$$u_w = [\dot{h}_{\omega_x} \ \dot{h}_{\omega_y} \ \dot{h}_{\omega_z}]^T = C\dot{h}_a, \quad (5)$$

where  $[h_{\omega_x} \ h_{\omega_y} \ h_{\omega_z}]^T$  is reaction wheels angular momentum in body axes and  $C$  is the orientation matrix of reaction wheels.

*2.2. Gyroscope Model.* We formulate a simplified gyro model as described in [11] as follows:

$$\omega_g = \omega + b + g_{sf} + g_{ma} + n_a, \quad (6)$$

where  $\omega$  is the true rate,  $b$  is the bias drift, and  $n_a$  is the white noise accounting for angular random walk at the gyro

angle level.  $g_{sf}$ ,  $g_{ma}$  are the gyro scale factor and misalignment errors, respectively, expressed as follows:

$$g_{sf} = \text{diag}\left(\begin{bmatrix} k_{sfx} & k_{sfy} & k_{sfz} \end{bmatrix}\right) \cdot \omega,$$

$$\begin{bmatrix} g_{max} \\ g_{may} \\ g_{maz} \end{bmatrix} = \begin{bmatrix} 0 & k_{xy} & k_{xz} \\ k_{yx} & 0 & k_{yz} \\ k_{zx} & k_{zy} & 0 \end{bmatrix} \cdot \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}. \quad (7)$$

By integrating (7), gyro rate measurement of (6) can be rewritten as follows [11]:

$$\omega_g = (I + K)\omega + b + n_a, \quad (8)$$

where

$$K = \begin{bmatrix} k_{sfx} & k_{xy} & k_{xz} \\ k_{yx} & k_{sfy} & k_{yz} \\ k_{zx} & k_{zy} & k_{sfz} \end{bmatrix}. \quad (9)$$

The drift bias is modeled as a random walk process as

$$\frac{d}{dt} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = \begin{bmatrix} n_{rx} \\ n_{ry} \\ n_{rz} \end{bmatrix}, \quad (10)$$

where  $n_r$  is the white noise accounting for the rate random walk (RRW) noise process at the rate level. The elements of the gyro misalignment matrix are modeled as a random walk process as [11]

$$\frac{d}{dt} \begin{bmatrix} k_{sfx} \\ k_{sfy} \\ k_{sfz} \end{bmatrix} = \begin{bmatrix} n_{sfx} \\ n_{sfy} \\ n_{sfz} \end{bmatrix}, \quad \frac{d}{dt} \begin{bmatrix} k_{xy} \\ k_{xz} \\ k_{yx} \\ k_{yz} \\ k_{zx} \\ k_{zy} \end{bmatrix} = \begin{bmatrix} n_{gxy} \\ n_{gxz} \\ n_{gyx} \\ n_{gyz} \\ n_{gzx} \\ n_{gzy} \end{bmatrix}. \quad (11)$$

The continuous noise covariance matrix is defined as

$$Q^{\text{Gyro}} = \begin{bmatrix} \sigma_a^2 I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & \sigma_r^2 I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & \sigma_{sf}^2 I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & \sigma_{usf}^2 I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & \sigma_{lsf}^2 I_{3 \times 3} \end{bmatrix}, \quad (12)$$

where  $\sigma_a^2$  is variance of the gyro angular random walk,  $\sigma_r^2$  is the variance of the gyro rate random walk,  $\sigma_{sf}^2$  is variance of the gyro scale factor,  $\sigma_{usf}^2$  is elements variance of the gyro upper misalignment, and  $\sigma_{lsf}^2$  is elements variance of the gyro lower misalignment.

At the end of stereoisaging maneuvers, the effect of scale factor and misalignment on the estimator is minimal. So, we can formulate a simplified gyro model while the satellite is in "uniform motion" mode as follows [11]:

$$\omega_g = \omega + b + n_a. \quad (13)$$

**2.3. Star Sensor Model.** The stars are assumed to be inertially fixed neglecting the effects of proper motion and velocity aberration. It is established in that the photograph image plane coordinates of the  $j$ th star are determined by the stellar collinearity equations [20]

$$x_j = -f \left( \frac{A_{11}s_{xj} + A_{12}s_{yj} + A_{13}s_{zj}}{A_{31}s_{xj} + A_{32}s_{yj} + A_{33}s_{zj}} \right),$$

$$y_j = -f \left( \frac{A_{21}s_{xj} + A_{22}s_{yj} + A_{23}s_{zj}}{A_{31}s_{xj} + A_{32}s_{yj} + A_{33}s_{zj}} \right), \quad (14)$$

where  $f$  is the known focal length and  $s_j$  is the vector toward the  $j$ th star as

$$s_{xj} = \cos \delta_j \cos \alpha_j, \quad s_{yj} = \cos \delta_j \sin \alpha_j, \quad s_{zj} = \sin \delta_j, \quad (15)$$

where  $\alpha_i$  is the right ascension,  $\delta_i$  is the declination of the  $j$ th star in the earth-centered inertial (ECI) coordinate system that are supposed to be available in a computer-accessible catalog [20].  $A_{ij}$  are elements of the satellite attitude matrix that are not known. So, if the measured stars can be identified as specific cataloged stars, then the attitude matrix and associated satellite orientation angles ( $\varphi, \theta, \psi$ ) will be determined from the measured stars in image coordinates and identified stars in inertial coordinates. This will be accomplished using the maximum-likelihood approach minimizing the following loss function [20]:

$$j(\hat{A}) = \sum_{j=1}^N \sigma_{\text{star\_vector}_j}^{-2} \tilde{b}_j^T \hat{A} s_j, \quad (16)$$

where  $\sigma_{\text{star\_vector}_j}$  is standard deviation of the  $j$ th star measurement noise ( $v_j$ ) with covariance matrix  $Q_j^{\text{star\_vector}}$ . The solution of this problem has been known as the q-method [20]. Therefore, in order to determine the attitude of the satellite using star vectors, the star observation can be reconstructed in unit vector form as

$$b_j = A s_j, \quad j = 1, 2, \dots, N, \quad (17)$$

where  $N$  is the total number of star observations and

$$b_j = \frac{1}{\sqrt{f^2 + x_j^2 + y_j^2}} \begin{bmatrix} -x_j \\ -y_j \\ f \end{bmatrix}, \quad s_j = \begin{bmatrix} s_{xj} & s_{yj} & s_{zj} \end{bmatrix}^T. \quad (18)$$

It has been shown in [9] that nearly all the probability of the errors is concentrated on a very small area about the direction of  $A s_j$ , so the sphere containing that point can be approximated by [20]

$$\tilde{b}_j = A s_j + v_j, \quad v_j A s_j = 0, \quad (19)$$

where  $\tilde{b}_j$  denotes the  $j$ th star measurement and the corresponding error  $v_j$  is approximately zero-mean Gaussian noise with covariance matrix as follows [21]:

$$Q_j^{\text{star\_vector}} = \frac{\sigma_s^2}{1 + d(x_i^2 + y_i^2)} \begin{bmatrix} (1 + dx_i^2)^2 & (dx_i y_i)^2 \\ (dx_i y_i)^2 & (1 + dy_i^2)^2 \end{bmatrix}, \quad (20)$$

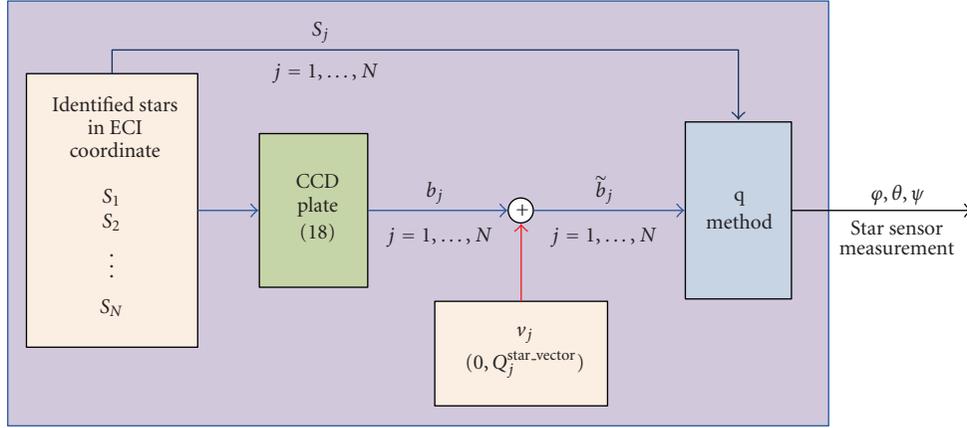


FIGURE 2: Star sensor model.

where  $d$  is on the order of one and  $\sigma_s$  is standard deviation of CCD centroiding error which is assumed to be known.

In summary, the computed orientation angles using the star vectors (19) and applying the q-method will be considered as the star sensor measurements  $(\varphi, \theta, \psi)$ . So, the accuracy of determined attitude  $(\varphi, \theta, \psi)$  depends on the number of identified stars and their position in the photograph image plane. The block diagram of star sensor model is shown in Figure 2.

According to Figure 2, the number of identified star changes during the motion of satellite relative to ECI coordinate system. This will affect the standard deviation of star sensor output measurement noise ( $\sigma^{\text{star}}$ ) strongly.

### 3. Interacting Multiple Models (IMM) Attitude Estimation Baseline

This adaptive estimator structure contains a bank of Kalman filters run in parallel at every time, each based on a particular model, to obtain model-based estimates and check the status of the operation system; the overall state estimate is a kind of combination of those model-based estimates. Therefore, each cycle of IMM algorithm consists of four major steps: (i) model conditional reinitialization (interacting or mixing of the estimates), in which the input to the filter matched to a certain mode is obtained by mixing the estimates of all filters at the previous time under the assumption that this particular mode is in effect at the present time; (ii) model-conditional filtering, performed in parallel for each mode; (iii) mode probability update, based on the model conditional likelihood functions; (iv) estimate combination, which yields the overall state estimate as the probabilistically weighted sum of the updated state estimates of all filters. The probability of a mode being in effect plays a key role in determining the weights in the combination of state estimates and covariance matrices. The dynamics of the plant for designing the Kalman filters for each mode are described as follows:

$$\begin{aligned} x(k+1) &= F_j(k) \cdot x(k) + G_j(k) \cdot U(k) + \xi_j(k), \\ z(k) &= H_j(k) \cdot x(k) + \vartheta_j(k). \end{aligned} \quad (21)$$

The subscript  $j$  denotes quantities pertaining to model  $m_j$ . System matrices  $F_j$ ,  $G_j$ , and  $H_j$  may be of different structures for different  $j$ . The process noise and noise measurement vectors  $\xi_j$  and  $\vartheta_j$  are white Gaussian noises of zero mean and covariance matrices  $Q_j$  and  $R_j$  as follows:

$$E[\xi_j(k)\xi_j^T(k)] = Q_j, \quad E[\vartheta_j(k)\vartheta_j^T(k)] = R_j. \quad (22)$$

The system mode sequence is assumed to be a first-order Markov chain with transition probabilities matrix [22]

$$\begin{aligned} P\{m_j(k+1) | m_i(k)\} &= \pi_{ij}(k) \quad \forall m_i, m_j \in S, \\ \sum_j \pi_{ij}(k) &= 1, \quad i = 1, 2, \dots, N, \end{aligned} \quad (23)$$

where  $P\{\cdot\}$  denotes probability,  $m(k)$  is the discrete valued modal state at time  $k$ , which denotes the mode in effect during the sampling period ending at  $t_k$ , and  $\pi_{ij}$  is the transition probability from mode  $m_i$  to mode  $m_j$ .  $S = \{m_1, m_2, \dots, m_N\}$  is the set of all possible system models. A general structure of the IMM method is shown in Figure 3.

Steps from 1 to 4 presents a complete cycle of the IMM estimator with Kalman filters. More details can be found in [22–27].

*Step 1.* Interaction/mixing of the estimates (for  $j = 1, 2, \dots, N$ ):

Predicted mode probability:

$$\mu_j(k+1 | k) = \sum_{i=1}^N \pi_{ij} \mu_i(k). \quad (24)$$

Transition probability:

$$\mu_{ij} = \frac{\sum_{i=1}^N \pi_{ij} \mu_i(k)}{\sum_{i=1}^N \mu_j(k+1 | k)}. \quad (25)$$

Mixing estimate:

$$\hat{x}_j^0(k | k) = \sum_{i=1}^N \hat{x}_i(k | k) \mu_{ij}(k). \quad (26)$$

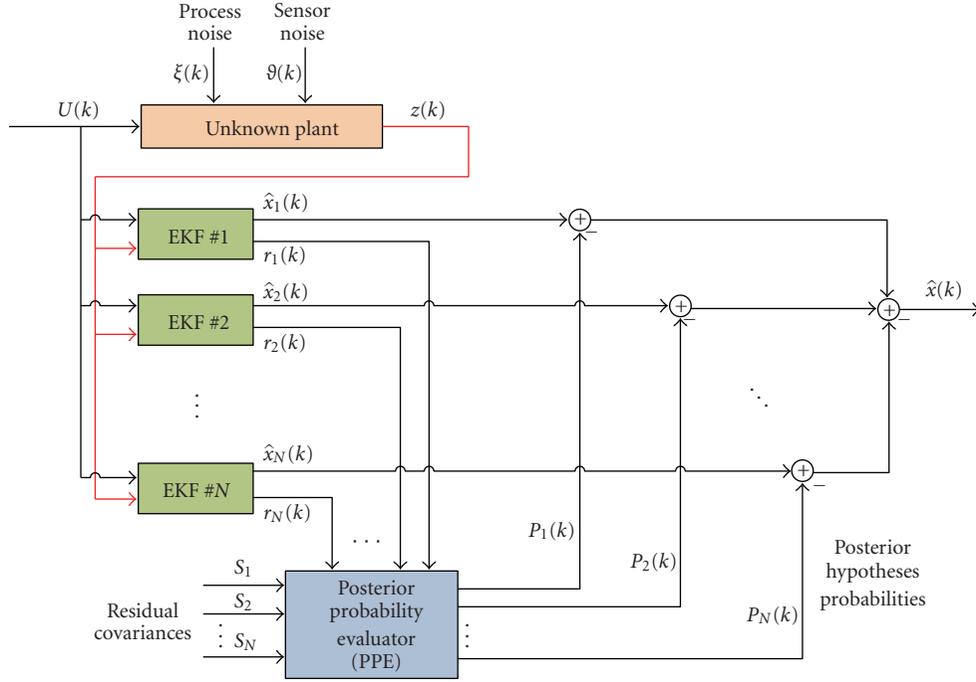


FIGURE 3: General structure of the IMM method [22].

Mixing covariance:

$$P_j^0(k | k) = \sum_{i=1}^N \left\{ P_j(k | k) + \left[ \hat{x}_j^0(k | k) - \hat{x}_i(k | k) \right] \right. \\ \left. \times \left[ \hat{x}_j^0(k | k) - \hat{x}_i^T(k | k) \right] \right\} \mu_{ij}(k). \quad (27)$$

Step 2. Model-condition filtering (for  $j = 1, 2, \dots, N$ ):

Predicted state:

$$\hat{x}_j(k+1 | k+1) = F_j(k) \hat{x}_j^0(k | k) + G_j(k) U(k) + \xi_j. \quad (28)$$

Predicted covariance:

$$P_j(k+1 | k) = F_j(k) P_j^0(k | k) F_j^T(k) + Q_j(k). \quad (29)$$

Measurement residual:

$$r_j = z(k+1) - H_j(k+1) \hat{x}_j(k+1 | k). \quad (30)$$

Residual covariance:

$$S_j = H_j(k+1) P_j(k+1 | k) H_j^T(k+1) + R_j(k+1). \quad (31)$$

Filter gain:

$$K_j = P_j(k+1 | k) H_j^T(k+1) S_j^{-1}. \quad (32)$$

Updated state:

$$\hat{x}_j(k+1 | k+1) = \hat{x}_j(k+1 | k) + K_j r_j. \quad (33)$$

Updated covariance:

$$P_j(k+1 | k+1) = P_j(k+1 | k) + K_j S_j K_j^T. \quad (34)$$

Step 3. Mode probability update (for  $j = 1, 2, \dots, N$ ):

Likelihood function:

$$L_j(k+1) = \frac{1}{\sqrt{|2\pi S_j|}} \exp\left(-\frac{1}{2} r_j^T S_j^{-1} r_j\right). \quad (35)$$

Mode probability:

$$\mu_j(k+1) = \frac{\mu_j(k+1 | k) L_j(k+1)}{\sum_{i=1}^N \mu_i(k+1 | k) L_i(k+1)}. \quad (36)$$

Step 4. Combination of estimates

Overall state estimate:

$$\hat{x}(k+1 | k+1) = \sum_{i=1}^N \hat{x}_i(k+1 | k+1) \mu_i(k+1). \quad (37)$$

Overall covariance:

$$P(k+1 | k+1) \\ = \sum_{i=1}^N \left\{ P_i(k+1 | k+1) \right. \\ \left. + [\hat{x}(k+1 | k+1) - \hat{x}_i(k+1 | k+1)] \right. \\ \left. \times [\hat{x}(k+1 | k+1) - \hat{x}_i(k+1 | k+1)]^T \right\} \\ \times \mu_i(k+1). \quad (38)$$

3.1. IMM Attitude Estimator in “Maneuvering Motion” Mode (IMM\_CT). In this section, the IMM baseline algorithm will be adopted to design an attitude estimator while the satellite is in “maneuvering motion” mode. This estimator consists of three EKF’s ( $N = 3$ ) with different structures (6, 9, and 15 states) which are required in different operating conditions of the satellite during each of stereomaging maneuvers to maintain the peak attitude estimation errors less enough while the satellite starts to maneuver (“maneuvering motion” mode). In this structure, the 15- and 9-state EKF estimators will be selected at the beginning of maneuvers while the satellite is in “maneuvering motion” mode, and the 6-state EKF estimator will be selected at the end of maneuvers while

the satellite placed in the “uniform motion” mode [17]. This structure is depicted in Figure 4.

Using (1), (7), and (10), we can form a 15- state dynamic error model for the 15-state Kalman filter as described in a compact form of first order differential equation

$$\begin{aligned}\dot{x} &= f_{15 \times 1}(x, u, t) + w, \\ y &= Cx + v,\end{aligned}\quad (39)$$

where  $C$  is the measurement matrix,  $w$  and  $v$  are process and sensor measurement noise, respectively, and

$$\begin{aligned}x &= [\Delta\theta_x \ \Delta\theta_y \ \Delta\theta_z \ \Delta b_x \ \Delta b_y \ \Delta b_z \ \Delta k_{sfx} \ \Delta k_{sfy} \ \Delta k_{sfz} \ \Delta k_{xy} \ \Delta k_{xz} \ \Delta k_{yx} \ \Delta k_{yz} \ \Delta k_{zx} \ \Delta k_{zy}]^T, \\ y &= [\Delta\theta_x \ \Delta\theta_y \ \Delta\theta_z]^T, \quad u = [\omega_x \ \omega_y \ \omega_z]^T.\end{aligned}\quad (40)$$

The 9-state Kalman filter contains three star attitude errors, three gyro bias errors and three scale factor errors. The 6-state Kalman filter contains three star attitude errors and three gyro bias errors. The linearization in the operating point of the system  $(x_0, u_0)$  converts the system (39) to the form

$$\begin{aligned}\dot{x} &= Ax + Bu + w, \\ y &= Cx + v,\end{aligned}\quad (41)$$

where

$$\begin{aligned}C &= [I_{3 \times 3} \ 0_{3 \times 12}]_{3 \times 15}, \\ B &= \begin{bmatrix} I_{3 \times 3} \\ 0_{12 \times 3} \end{bmatrix}.\end{aligned}\quad (42)$$

Based on continuous system discretization theory, the discrete time model of (41) can be written as

$$\begin{aligned}x(k+1) &= F(k) \cdot x(k) + G(k) \cdot U(k) + \xi(k), \\ y(k) &= H(k) \cdot x(k) + \vartheta(k).\end{aligned}\quad (43)$$

According to Figure 4 and using design principle described in Steps from 1 to 4, the estimate of gyro bias and misalignments, actual angular velocity of the satellite, and attitude estimation are updated according to the following steps (one cycle of the IMM\_CT method) [17].

Step 5. Residual update (for  $j = 1, 2, 3$ )

$$\hat{x}_j(4:15) = \hat{x}_j(4:15) - \hat{x}(4:15),$$

$$\begin{bmatrix} \hat{x}(1) \\ \hat{x}(2) \\ \hat{x}(3) \end{bmatrix}_n = \begin{bmatrix} 1 & 0 & -\sin(\theta_n) \\ 0 & \cos(\varphi_n) & \cos(\theta_n) \sin(\varphi_n) \\ 0 & -\sin(\varphi_n) & \cos(\varphi_n) \cos(\theta_n) \end{bmatrix}$$

$$\times \left( \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix}_0 - \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix}_n \right),$$

$$\begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix}_0 = \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix}_{n-1}$$

$$+ \begin{bmatrix} 1 & 0 & -\sin(\theta_{n-1}) \\ 0 & \cos(\varphi_{n-1}) & \cos(\theta_{n-1}) \sin(\varphi_{n-1}) \\ 0 & -\sin(\varphi_{n-1}) & \cos(\varphi_{n-1}) \cos(\theta_{n-1}) \end{bmatrix}^{-1}$$

$$\times \begin{bmatrix} \hat{x}_j(1) \\ \hat{x}_j(2) \\ \hat{x}_j(3) \end{bmatrix}_n.\quad (44)$$

Step 6. IRU compensated rate:

$$\omega = (I - \hat{K})\omega_g - \hat{b}.\quad (45)$$

Step 7. Steps 1 to 4 ( $N = 3$ ).

Step 8. Attitude, angular rate, and IRU compensation parameters update

$$\begin{aligned}
 \begin{bmatrix} \hat{\varphi} \\ \hat{\theta} \\ \hat{\psi} \end{bmatrix}_{\text{new}} &= \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix}_n \\
 &+ \begin{bmatrix} 1 & 0 & -\sin(\theta_n) \\ 0 & \cos(\varphi_n) & \cos(\theta_n) \sin(\varphi_n) \\ 0 & -\sin(\varphi_n) & \cos(\varphi_n) \cos(\theta_n) \end{bmatrix}^{-1} \\
 &\cdot \begin{bmatrix} \hat{x}(1) \\ \hat{x}(2) \\ \hat{x}(3) \end{bmatrix}_{n+1} \\
 \hat{b}(n+1) &= \hat{b}(n) + \hat{x}_{4:6}(n+1), \\
 \hat{g}_{ij}(n+1) &= \hat{g}_{ij}(n) + \hat{x}_{7:15}(n+1), \\
 \hat{\omega}_n &= \omega_n - \begin{bmatrix} x(4) \\ x(5) \\ x(6) \end{bmatrix}_{n+1} - \begin{bmatrix} \omega_x & 0 & 0 \\ 0 & \omega_y & 0 \\ 0 & 0 & \omega_z \end{bmatrix} \\
 &\cdot \begin{bmatrix} x(7) \\ x(8) \\ x(9) \end{bmatrix}_{n+1} \\
 &- \begin{bmatrix} \omega_y & 0 & 0 \\ 0 & \omega_z & 0 \\ 0 & 0 & \omega_x \end{bmatrix} \cdot \begin{bmatrix} x(10) \\ x(11) \\ x(12) \end{bmatrix}_{n+1} \\
 &- \begin{bmatrix} \omega_z & 0 & 0 \\ 0 & \omega_x & 0 \\ 0 & 0 & \omega_y \end{bmatrix} \cdot \begin{bmatrix} x(13) \\ x(14) \\ x(15) \end{bmatrix}_{n+1}.
 \end{aligned} \tag{46}$$

In Steps from 5 to 8, we have considered that

$$\begin{aligned}
 R_j &= \max(\sigma^{\text{star}})^2, \quad j = 1, 2, 3, \\
 Q_1 &= Q^{\text{Gyro}}, \\
 Q_2 &= \begin{bmatrix} \sigma_a^2 I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & \sigma_r^2 I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & \sigma_{\text{st}}^2 I_{3 \times 3} \end{bmatrix}, \\
 Q_3 &= \begin{bmatrix} \sigma_a^2 I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & \sigma_r^2 I_{3 \times 3} \end{bmatrix}.
 \end{aligned} \tag{47}$$

We will illustrate that applying this algorithm maintains the peak estimation errors less enough at the start of maneuvers.

But, as mentioned before, this structure cannot improve the steady-state estimation accuracy while the satellite is in “uniform motion” mode (deficiency of IMM\_CT structure). So, in the next section we will design the “IMM.L” attitude estimator to reduce the steady-state estimation errors while the satellite is in “uniform motion” mode.

3.2. IMM Attitude Estimator in “Uniform Motion” Mode (IMM.L). As we mentioned in Section 2.3, the number of identified star changes during the motion of satellite relative to ECI coordinate system which will affect the standard deviation of star sensor measurement noise ( $\sigma^{\text{star}}$ ) strongly.

Here, the IMM baseline algorithm will be adopted to (21) with different star sensor measurement noise levels as a new idea to decrease attitude estimation errors while the satellite is in “uniform motion” mode. The “IMM.L” estimator will be designed based on 6-state EKF structure in order to identify the suitable star sensor measurement noise level. This will improve attitude estimation accuracy of the satellite at the end of maneuvers while imaging takes place. We will augment (2) and (5) with 6-state filter structure to consider the dynamic model of satellite and reaction wheels. Thus, the proposed IMM.L structure consists of 12 states as follows:

$$\begin{aligned}
 \dot{x} &= f_{12 \times 1}(x, u, t) + w, \\
 y &= Cx + v,
 \end{aligned} \tag{48}$$

where

$$\begin{aligned}
 x &= [\Delta\theta_x \ \Delta\theta_y \ \Delta\theta_z \ \omega_x \ \omega_y \ \omega_z \ h_{\omega_x} \ h_{\omega_y} \ h_{\omega_z} \ \Delta b_x \ \Delta b_y \ \Delta b_z], \\
 y &= [\Delta\theta_x \ \Delta\theta_y \ \Delta\theta_z \ \omega_x \ \omega_y \ \omega_z \ h_{\omega_x} \ h_{\omega_y} \ h_{\omega_z}], \\
 U &= [\dot{h}_{\omega_x} \ \dot{h}_{\omega_y} \ \dot{h}_{\omega_z} \ \tau_x \ \tau_y \ \tau_z],
 \end{aligned} \tag{49}$$

where  $C$  is the measurement matrix as follows:

$$C = \begin{bmatrix} I_{9 \times 9} & \begin{bmatrix} 0_{3 \times 3} \\ I_{3 \times 3} \\ 0_{3 \times 3} \end{bmatrix} \end{bmatrix}_{9 \times 12}. \tag{50}$$

The linearization and then discretization, convert the system (47) to the form of (43). The IMM.L attitude estimator structure is shown in Figure 5. The estimation of gyro bias, actual angular velocity, and also attitude estimation of the satellite is updated according to Steps from 9 to 12 the (one cycle of the IMM.L method).

Step 9. Residual update (for  $j = 1, 2, \dots, 8$ )

$$\begin{aligned}
 \dot{x}_j(4:9) &= \hat{x}_j(4:9), \\
 \dot{x}_j(10:12) &= \hat{x}_j(10:12) - \hat{x}(10:12), \\
 \begin{bmatrix} \dot{x}(1) \\ \dot{x}(2) \\ \dot{x}(3) \end{bmatrix}_n &= \begin{bmatrix} 1 & 0 & -\sin(\theta_n) \\ 0 & \cos(\varphi_n) & \cos(\theta_n) \sin(\varphi_n) \\ 0 & -\sin(\varphi_n) & \cos(\varphi_n) \cos(\theta_n) \end{bmatrix} \\
 &\times \left( \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix}_0 - \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix}_n \right),
 \end{aligned} \tag{51}$$

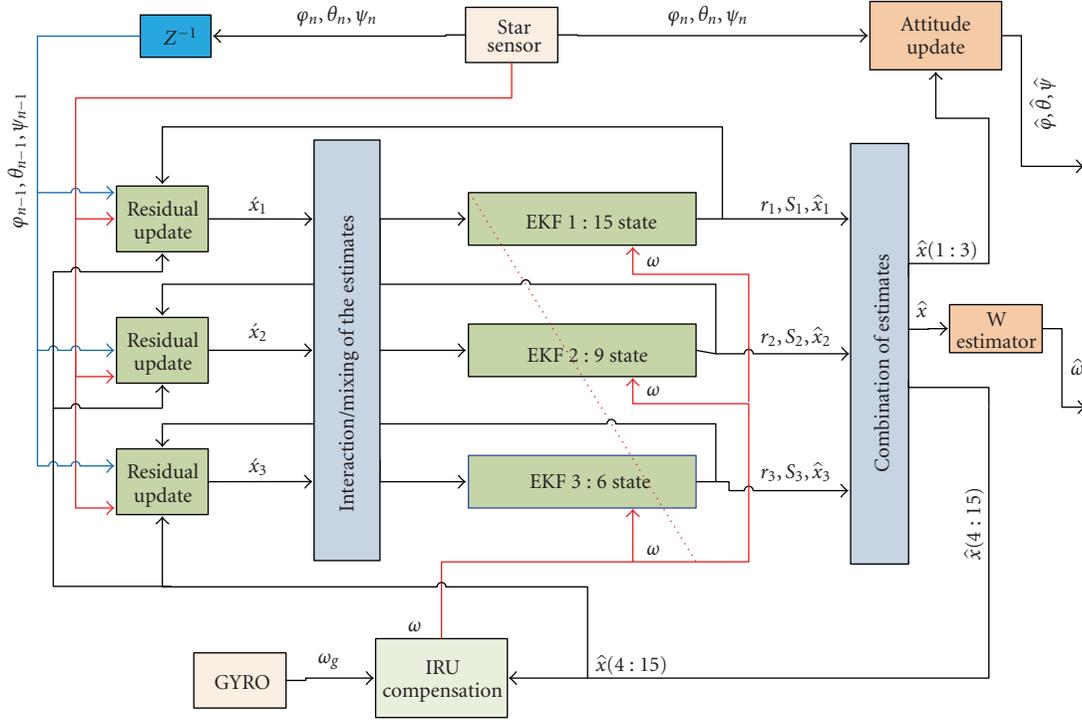


FIGURE 4: IMM\_CT attitude estimator structure.

where

$$\begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix}_0 = \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix}_{n-1} + \begin{bmatrix} 1 & 0 & -\sin(\theta_{n-1}) \\ 0 & \cos(\varphi_{n-1}) & \cos(\theta_{n-1}) \sin(\varphi_{n-1}) \\ 0 & -\sin(\varphi_{n-1}) & \cos(\varphi_{n-1}) \cos(\theta_{n-1}) \end{bmatrix}^{-1} \times \begin{bmatrix} \hat{x}_j(1) \\ \hat{x}_j(2) \\ \hat{x}_j(3) \end{bmatrix}_n \quad (52)$$

Step 10. IRU compensated rate:

$$\omega = \omega_g - \hat{b}. \quad (53)$$

Step 11. Steps 1 to 4 ( $N = 8$ ).

Step 12. Attitude and IRU compensation parameters update

$$\begin{bmatrix} \hat{\varphi} \\ \hat{\theta} \\ \hat{\psi} \end{bmatrix}_{\text{new}} = \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix}_n + \begin{bmatrix} 1 & 0 & -\sin(\theta_n) \\ 0 & \cos(\varphi_n) & \cos(\theta_n) \sin(\varphi_n) \\ 0 & -\sin(\varphi_n) & \cos(\varphi_n) \cos(\theta_n) \end{bmatrix}^{-1}$$

$$\cdot \begin{bmatrix} \hat{x}(1) \\ \hat{x}(2) \\ \hat{x}(3) \end{bmatrix}_{n+1}$$

$$\hat{\omega}_n = \begin{bmatrix} x(4) \\ x(5) \\ x(6) \end{bmatrix}_{n+1}$$

$$\hat{b}(n+1) = \hat{b}(n) + \hat{x}_{10:12}(n+1).$$

(54)

In this structure, we have considered that

$$R_j = \begin{bmatrix} R_j^{\text{star}} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & \sigma_r^2 I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & \sigma_a^2 I_{3 \times 3} \end{bmatrix}_{9 \times 9},$$

$Q_j$

$= Q$

$$= \begin{bmatrix} Q_{3 \times 3}^{\text{Satellite\_Kinematic}} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & Q_{3 \times 3}^{\text{Satellite\_Dynamic}} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & Q_{3 \times 3}^{hw} & 0_{3 \times 3} \\ 0_{15 \times 3} & 0_{15 \times 3} & 0_{3 \times 3} & \sigma_a^2 I_{3 \times 3} \end{bmatrix}_{12 \times 12} \quad (55)$$

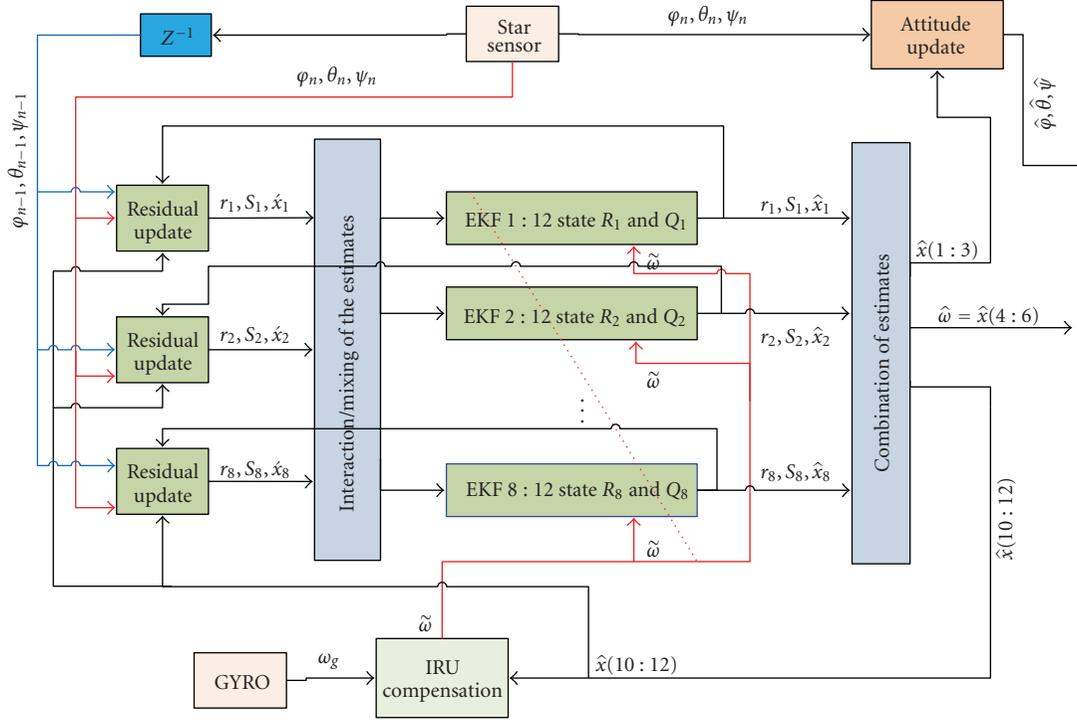


FIGURE 5: IMM.L attitude estimator structure.

where

$$\begin{aligned} Q_{3 \times 3}^{hw} &= \sigma_{hw}^2 I_{3 \times 3}, \\ Q_{3 \times 3}^{\text{Satellite\_Dynamic}} &= \sigma_{SD}^2 I_{3 \times 3}, \\ Q_{3 \times 3}^{\text{Satellite\_Kinematic}} &= \sigma_{SK}^2 I_{3 \times 3}, \end{aligned} \quad (56)$$

and  $R_j^{\text{star}}$ ,  $j = 1, 2, \dots$  are the different covariance matrices of the star sensor measurement errors. It should be mentioned that applying this structure cannot reduce the peak attitude estimation errors at the beginning of the stereoimaging maneuvers (deficiency of IMM.L structure).

**3.3. EIMMAE Attitude Estimator.** In this section, we will design an extended interacting multiple models adaptive estimator (EIMMAE) to maintain the peak attitude estimation errors less enough while the satellite is in “maneuvering motion” mode and to reduce attitude estimation errors while the satellite is in “uniform motion” mode simultaneously. The proposed algorithm will select the suitable Kalman filter structure while the satellite is in “maneuvering motion” mode and then will select the suitable star sensor measurement noise level while the satellite placed in “uniform motion” mode. The proposed EIMMAE attitude estimator structure is shown in Figures 6 and 7. In order to provide combination of estimates from the IMM.L and IMM.CT outputs, we will consider PE1 and PE2 blocks to convert the states and covariance matrix from 15-state EKF structure to states and covariance matrix of 12-state EKF structure and vice versa.

The IMM.L estimator block in Figure 6 has been elaborated in Figure 7.

In “interaction/mixing of the estimates IMM.L” block, we have supposed that the probability of transition between different EKF structures of IMM.L estimator is zero ( $\mu_{L_{ij}} = 0$  for  $i \neq j$ ) and the transition probability from 15 and 9 states EKF structures to IMM.L models is equal to the transition probability between EKFs of the IMM.CT configuration. So, the mixing states estimations stage will be achieved as follows:

$$\begin{aligned} \hat{x}_j^0(k | k) &= \hat{x}_j(k | k)\mu_3(k) + x_1(k | k)\mu_{1|3}(k) \\ &\quad + x_2(k | k)\mu_{2|3}(k). \end{aligned} \quad (57)$$

According to the above assumptions and considering the EIMMAE block diagram, the estimation of gyro bias, actual angular velocity of the satellite, and attitude estimation are updated according to Steps from 13 to 26 the (one cycle of the EIMMAE method).

*Step 13.* Residual update IMM.CT (for  $j = 1, 2$ )

$$\hat{x}_j(4 : 15) = \hat{x}_j(4 : 5) - \hat{x}(4 : 15),$$

$$\begin{aligned} \begin{bmatrix} \hat{x}(1) \\ \hat{x}(2) \\ \hat{x}(3) \end{bmatrix}_n &= \begin{bmatrix} 1 & 0 & -\sin(\theta_n) \\ 0 & \cos(\varphi_n) & \cos(\theta_n) \sin(\varphi_n) \\ 0 & -\sin(\varphi_n) & \cos(\varphi_n) \cos(\theta_n) \end{bmatrix} \\ &\quad \times \left( \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix}_0 - \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix}_n \right), \end{aligned} \quad (58)$$

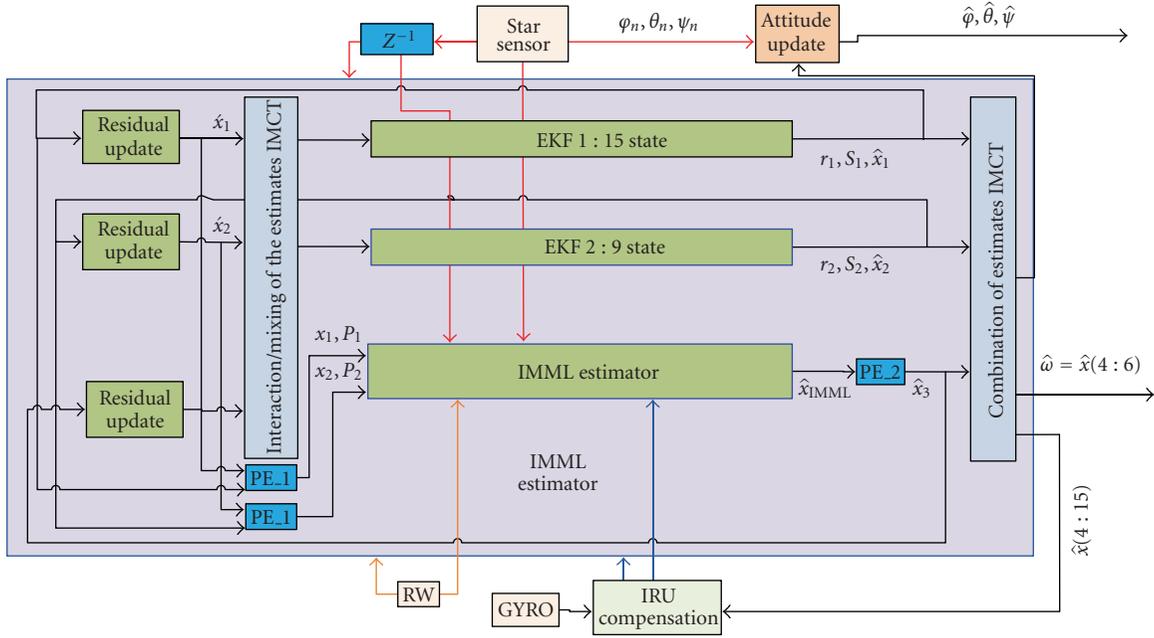


FIGURE 6: EIMMAE structure.

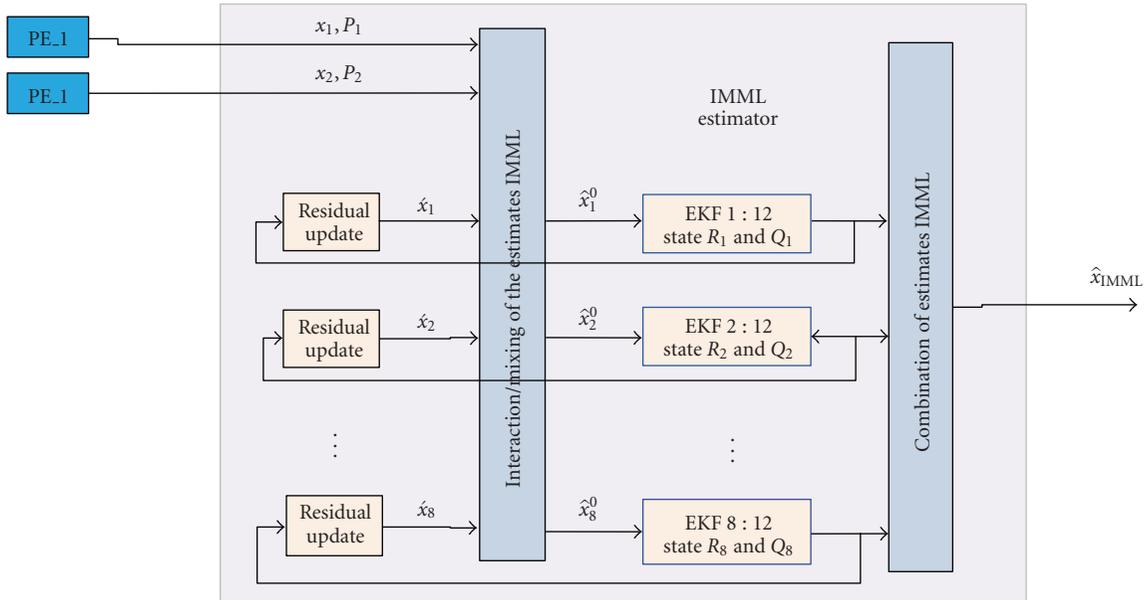


FIGURE 7: IMM.L estimator block.

where

$$\begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix}_0 = \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix}_{n-1}$$

$$+ \begin{bmatrix} 1 & 0 & -\sin(\theta_{n-1}) \\ 0 & \cos(\varphi_{n-1}) & \cos(\theta_{n-1}) \sin(\varphi_{n-1}) \\ 0 & -\sin(\varphi_{n-1}) & \cos(\varphi_{n-1}) \cos(\theta_{n-1}) \end{bmatrix}^{-1}$$

$$\times \begin{bmatrix} \hat{x}_j(1) \\ \hat{x}_j(2) \\ \hat{x}_j(3) \end{bmatrix}_n$$

(59)

Step 14. IRU compensated rate:

$$\omega = (I - \hat{K}) \omega_g - \hat{b}. \quad (60)$$

Step 15. Steps 1 and 2 ( $N = 2$ ).

Step 16. Residual update IMM.L (for  $j = 1, 2, \dots, 8$ )

$$\begin{aligned} \hat{x}_{lj}(4:9) &= \hat{x}_{lj}(4:9), \\ \hat{x}_{lj}(10:12) &= \hat{x}_{lj}(10:12) - \hat{x}_l(10:12), \\ \begin{bmatrix} \hat{x}(1) \\ \hat{x}(2) \\ \hat{x}(3) \end{bmatrix}_{lj} &= \begin{bmatrix} 1 & 0 & -\sin(\theta_n) \\ 0 & \cos(\varphi_n) & \cos(\theta_n) \sin(\varphi_n) \\ 0 & -\sin(\varphi_n) & \cos(\varphi_n) \cos(\theta_n) \end{bmatrix} \\ &\quad \times \left( \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix}_0 - \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix}_n \right), \\ \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix}_0 &= \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix}_{n-1} \\ &\quad + \begin{bmatrix} 1 & 0 & -\sin(\theta_{n-1}) \\ 0 & \cos(\varphi_{n-1}) & \cos(\theta_{n-1}) \sin(\varphi_{n-1}) \\ 0 & -\sin(\varphi_{n-1}) & \cos(\varphi_{n-1}) \cos(\theta_{n-1}) \end{bmatrix}^{-1} \\ &\quad \times \begin{bmatrix} \hat{x}_{lj}(1) \\ \hat{x}_{lj}(2) \\ \hat{x}_{lj}(3) \end{bmatrix}. \end{aligned} \quad (61)$$

Step 17. PE\_1 block (for  $j = 1, 2$ )

$$\begin{aligned} x_j(1:3) &= \hat{x}_j(1:3), \\ x_j(4:6) &= \omega_{n-1} - \begin{bmatrix} \omega_x & 0 & 0 \\ 0 & \omega_y & 0 \\ 0 & 0 & \omega_z \end{bmatrix}_{n-1} \cdot \begin{bmatrix} \hat{x}_j(7) \\ \hat{x}_j(8) \\ \hat{x}_j(9) \end{bmatrix} \\ &\quad - \begin{bmatrix} \omega_y & 0 & 0 \\ 0 & \omega_z & 0 \\ 0 & 0 & \omega_x \end{bmatrix}_{n-1} \cdot \begin{bmatrix} \hat{x}_j(10) \\ \hat{x}_j(11) \\ \hat{x}_j(12) \end{bmatrix} \\ &\quad - \begin{bmatrix} \omega_z & 0 & 0 \\ 0 & \omega_x & 0 \\ 0 & 0 & \omega_y \end{bmatrix} \cdot \begin{bmatrix} \hat{x}_j(13) \\ \hat{x}_j(14) \\ \hat{x}_j(15) \end{bmatrix}, \end{aligned}$$

$$x_j(7:9) = \text{wheels momentum}, \quad x_j(10:12) = \hat{x}_j(4:6). \quad (62)$$

Step 18. Interaction/mixing of the estimates IMM.L (for  $j = 1, 2, \dots, 8$ ):

$$\begin{aligned} \text{Mixing estimate:} \\ \hat{x}_{lj}^0(k|k) &= \hat{x}_{lj}(k|k)\mu_3(k) + x_1(k|k)\mu_{1|3}(k) \\ &\quad + x_2(k|k)\mu_{2|3}(k). \end{aligned} \quad (63)$$

Mixing covariance:

$$\begin{aligned} P_{lj}^0(k|k) &= \sum_{i=1}^2 \left\{ P_j(k|k) + [\hat{x}_{lj}^0(k|k) - x_i(k|k)] \right. \\ &\quad \times [\hat{x}_{lj}^0(k|k) - x_i^T(k|k)] \left. \right\} \mu_{i|3}(k) \\ &\quad + \left\{ P_{lj}(k|k) + [\hat{x}_{lj}^0(k|k) - \hat{x}_{lj}(k|k)] \right. \\ &\quad \times [\hat{x}_{lj}^0(k|k) - \hat{x}_{lj}^T(k|k)] \left. \right\} \mu_{3|3}(k). \end{aligned} \quad (64)$$

Step 19. Steps 2 and 3 by replacing  $\hat{x}, F_j, \hat{x}_{lj}^0, G_j$  with  $\hat{x}_l, F_{lj}, \hat{x}_{lj}^0, G_{lj}$ , respectively.

Step 20. Combination of estimates

Overall state estimate:

$$\hat{x}_l(k+1|k+1) = \sum_{i=1}^8 \hat{x}_{li}(k+1|k+1)\mu_{li}(k+1). \quad (65)$$

Overall covariance:

$$\begin{aligned} P_l(k+1|k+1) &= \sum_{i=1}^8 \left\{ P_{li}(k+1|k+1) \right. \\ &\quad + [\hat{x}_l(k+1|k+1) - \hat{x}_{li}(k+1|k+1)] \\ &\quad \times [\hat{x}_l(k+1|k+1) - \hat{x}_{li}(k+1|k+1)]^T \left. \right\} \\ &\quad \times \mu_{li}(k+1). \end{aligned} \quad (66)$$

Overall residual:

$$\hat{r} = z(k+1) - H_l(k+1)\hat{x}_l(k+1|k). \quad (67)$$

Residual covariance:

$$\begin{aligned} \hat{S}_l(k+1|k+1) &= \sum_{i=1}^8 \left\{ \hat{S}_{li}(k+1|k+1) \right. \\ &\quad + [\hat{r}_l(k+1|k+1) - \hat{r}_{li}(k+1|k+1)] \\ &\quad \times [\hat{r}_l(k+1|k+1) - \hat{r}_{li}(k+1|k+1)]^T \left. \right\} \\ &\quad \times \mu_{li}(k+1). \end{aligned} \quad (68)$$

Step 21. PE\_2 block:

$$\hat{x}_3 = A \cdot \hat{x}_l(1:3), \quad (69)$$

where

$$A = \begin{bmatrix} I_{3 \times 3} & 0_{3 \times 6} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 6} & I_{3 \times 3} \\ 0_{9 \times 3} & 0_{9 \times 6} & 0_{9 \times 3} \end{bmatrix}. \quad (70)$$

Step 22. Residual of tertiary model:

$$\begin{aligned} \hat{r}_3 &= z(k+1) - H(k+1)\hat{x}_l(k+1|k), \\ \hat{S}_3 &= A \cdot \hat{S}_l \cdot A^T. \end{aligned} \quad (71)$$

Step 23. Steps 3 and 4.

Step 24. Attitude, angular rate, and IRU compensation parameters update

$$\begin{aligned} \begin{bmatrix} \hat{\varphi} \\ \hat{\theta} \\ \hat{\psi} \end{bmatrix}_{\text{new}} &= \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix}_n \\ &+ \begin{bmatrix} 1 & 0 & -\sin(\theta_n) \\ 0 & \cos(\varphi_n) & \cos(\theta_n)\sin(\varphi_n) \\ 0 & -\sin(\varphi_n) & \cos(\varphi_n)\cos(\theta_n) \end{bmatrix}^{-1} \\ &\cdot \begin{bmatrix} \hat{x}(1) \\ \hat{x}(2) \\ \hat{x}(3) \end{bmatrix}_{n+1} \end{aligned} \quad (72)$$

$$\hat{b}(n+1) = \hat{b}(n) + \hat{x}_{4:6}(n+1),$$

$$\hat{g}_{ij}(n+1) = \hat{g}_{ij}(n) + \hat{x}_{7:15}(n+1).$$

Step 25. Angular rate estimation (for  $j = 1 : 2$ )

$$\begin{aligned} \hat{\omega}_j &= \omega_j - \begin{bmatrix} \hat{x}_j(4) \\ \hat{x}_j(5) \\ \hat{x}_j(6) \end{bmatrix}_{k+1} - \begin{bmatrix} \omega_x & 0 & 0 \\ 0 & \omega_y & 0 \\ 0 & 0 & \omega_z \end{bmatrix} \cdot \begin{bmatrix} \hat{x}_j(7) \\ \hat{x}_j(8) \\ \hat{x}_j(9) \end{bmatrix}_{k+1} \\ &- \begin{bmatrix} \omega_y & 0 & 0 \\ 0 & \omega_z & 0 \\ 0 & 0 & \omega_x \end{bmatrix} \cdot \begin{bmatrix} \hat{x}_j(10) \\ \hat{x}_j(11) \\ \hat{x}_j(12) \end{bmatrix}_{k+1} \\ &- \begin{bmatrix} \omega_z & 0 & 0 \\ 0 & \omega_x & 0 \\ 0 & 0 & \omega_y \end{bmatrix} \cdot \begin{bmatrix} \hat{x}_j(13) \\ \hat{x}_j(14) \\ \hat{x}_j(15) \end{bmatrix}_{k+1}. \end{aligned} \quad (73)$$

Step 26. Overall angular rate estimate:

$$\hat{\omega} = \mu_1 \hat{\omega}_1 + \mu_2 \hat{\omega}_2 + \mu_3 \hat{\omega}_l. \quad (74)$$

#### 4. Simulation

A simulation is done to show effectiveness and efficiency of the EIMMAE method for attitude estimation of a stereoisagery satellite with

$$J = \begin{bmatrix} 389.99 & -3.28 & -11.57 \\ -3.28 & 391.83 & -7.42 \\ -11.57 & -7.42 & 176.58 \end{bmatrix} \text{Kg m}^2. \quad (75)$$

Consider a scenario for achieving the stereoisagery mission shown in Figure 1. Fulfillment of this scenario requires

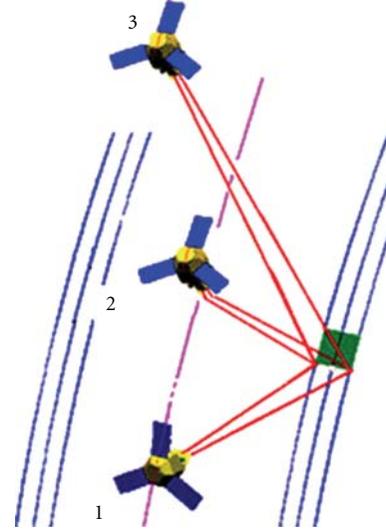


FIGURE 8: Stereoimaging Scenario.

four main slew maneuvers in a sequence. This sequence is shown in Figure 8 and starts from initial attitude (e.g., sun pointing) to nadir pointing which is called “Initial maneuver”. Afterwards, the satellite will slew to point 1 from nadir pointing, and the first picture will be taken by a snapshot after a specific constant pointing stability and accuracy are reached. The second slew is achieved from point 1 to point 2 (e.g., Nadir pointing). The second picture may be taken by a snapshot after a specific constant pointing stability and accuracy are reached. The third slew is achieved from point 2 to target attitude at point 3. The second picture will be usually taken during slew 3 by a snapshot after a specific constant pointing stability and accuracy are reached. The last slew is to back toward nadir pointing under 3-axis control. These required maneuvers are summarized in Table 1 [5].

In this scenario, a typical star sensor data has been considered based on produced data in [20]. A plot of the number of identified stars and star sensor measurements noise with a value of 0.005 deg for  $3\sigma_s$  and associated  $3\sigma$  boundaries over the 540 sec during the stereoisagery maneuvers (according to Table 1) are shown in Figure 9 [20].

According to Figure 9(b), it should be mentioned that  $3\sigma$  boundaries of the star sensor measurement errors will be changed according to the changes of the number of identified stars. In the following simulations, the noise parameters for the gyro measurements in (12) are given by  $\sigma_r = 3 \times 10^{-10} \text{ rad/sec}^{3/2}$  and  $\sigma_a = 3 \times 10^{-6} \text{ rad/sec}^{1/2}$ . The initial bias of the gyro for each axis is given by 0.01 deg/hr. The initial covariance of the gyro for the attitude error is set to 0.28 deg<sup>2</sup>, and the initial covariance for the gyro drift is set to 0.22 (deg/hr)<sup>2</sup>. The sampling interval for the star sensor and gyroscope is considered  $\Delta t = 0.1 \text{ sec}$ .

4.1. Performance Evaluation of the IMM-CT Attitude Estimator. In this simulation, we have supposed that the satellite will achieve a number of large angle maneuvers in a few seconds (100 seconds for each maneuver) to estimate the

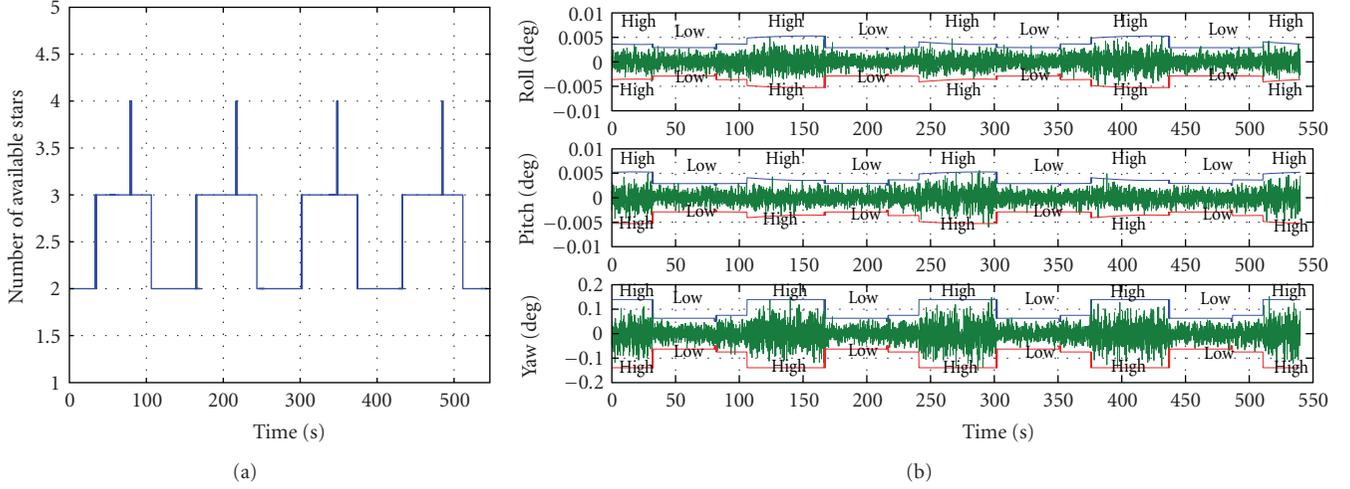


FIGURE 9: (a) Number of identified stars (b) Star sensor measurement errors and boundaries.

TABLE 1: Mission sequence for stereoimaging.

Maneuver	Attitude description	{Roll, Pitch, Yaw} Degree	
		From	to
Initial Slew	Slew from Initial condition to nadir pointing during 100 sec	{10, -10, 10}	{0, 0, 0}
Slew 1	Slew from nadir pointing to point 1 during 100 sec	{0, 0, 0}	{30, 30, 0}
Slew 2	Slew from point 1 to nadir pointing (point 2) during 100 sec	{30, 30, 0}	{0, 0, 0}
Slew 3	Slew from point 2 to point 3 during 100 sec	{0, 0, 0}	{30, -30, 0}
Slew 4	Slew from point 3 to nadir pointing during 100 sec	{30, -30, 0}	{0, 0, 0}

gyro bias, misalignments, and scale factor errors accurately, by applying this method according to Steps from 5 to 8 with  $N=3$  (15, 9, and 6-state EKF) and

$$\pi_{ij} = \begin{bmatrix} 0.999 & 0 & 0.001 \\ 0.008 & 0.99 & 0.002 \\ 0.008 & 0.002 & 0.99 \end{bmatrix}. \quad (76)$$

The posterior probability of EKFs ( $P_i(t)$ ) is shown in Figure 10. This figure shows that at the beginning of the maneuvers, the probability of the 15-state and 9-state EKFs are more than 6-state EKF to estimate gyro errors accurately; but at the end of maneuvers the probability of the 6-state structure is higher than the other structures while satellite is in “uniform motion” mode.

The gyro errors estimations are shown in Figures 11, 12, 13, and 14.

These figures verify the effectiveness of IMM\_CT method to estimate gyro errors accurately and to maintain the peak

TABLE 2: Standard deviation of the star sensor measurement noise levels around each axis.

Levels	$E[\vartheta_j(k)\vartheta_j^T(k)] = (3\sigma/3)^2$		
	Roll	Pitch	Yaw
(Low)	$(0.003/3)^2$	$(0.003/3)^2$	$(0.059/3)^2$
(High)	$(0.005/3)^2$	$(0.005/3)^2$	$(0.15/3)^2$

error of attitude estimation less enough while the satellite is in the “maneuvering motion” mode as mentioned in [17].

**4.2. Performance Evaluation of the IMM\_L Method.** According to Figure 9(b), the  $3\sigma$  boundaries of the star sensor measurement errors change between at least two main levels (marked as “high” and “low”) around each axis which are described in Table 2.

Therefore, we should consider at least  $N = 2^3$  different models, with different star sensor measurements covariance matrix ( $R_j^{\text{star}}, j = 1, 2, \dots, 8$ ), in order to design the IMM\_L

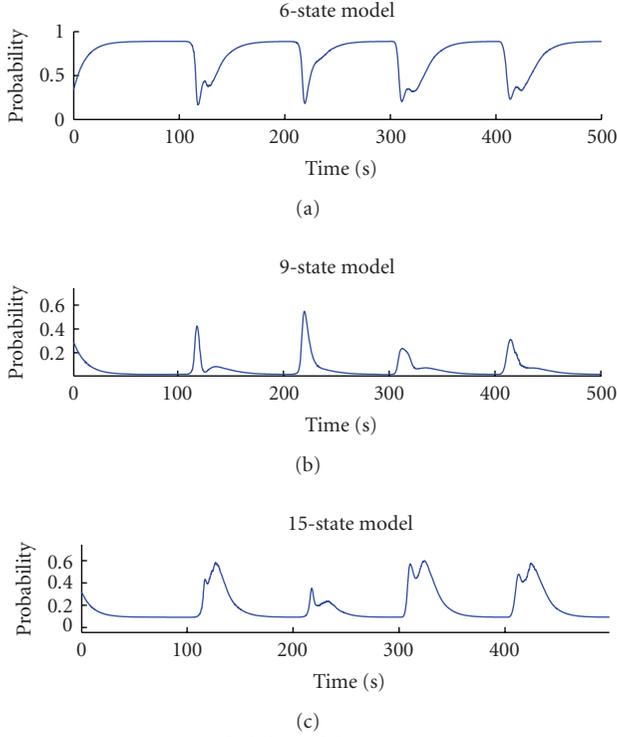


FIGURE 10: Posterior probability of the EKF structures in IMM\_CT method.

TABLE 3: Different model for star sensor measurement noise levels.

Number of models (N)	Star sensor measurements noise		
	Covariance ( $R_j^{\text{star}}, j = 1, 2, \dots, 8$ )		
	Roll	Pitch	Yaw
Model #1	High	High	High
Model #2	High	High	Low
Model #3	High	Low	High
Model #4	High	Low	Low
Model #5	Low	High	High
Model #6	Low	High	Low
Model #7	Low	Low	High
Model #8	Low	Low	Low

TABLE 4: Comparing results of EIMMAE structure with IMM.L and IMM\_CT methods.

Method	Average RMS error for 3 maneuvers (degree)		
	$(1/3) \sum (1/(t_n - t_0)) \sum_{t=t_0}^{t_n}  e(t) $		
	Roll $\Delta\varphi$ (Deg.)	Pitch $\Delta\theta$ (Deg.)	Yaw $\Delta\psi$ (Deg.)
IMM_CT	$0.405 \times 10^{-3}$	$0.411 \times 10^{-3}$	$0.376 \times 10^{-2}$
IMM.L	$0.332 \times 10^{-3}$	$0.345 \times 10^{-3}$	$0.228 \times 10^{-2}$
EIMMAE	$0.256 \times 10^{-3}$	$0.251 \times 10^{-3}$	$0.223 \times 10^{-2}$

estimator in the presence of star sensor measurement noise level changes. These models are mentioned in Table 3.

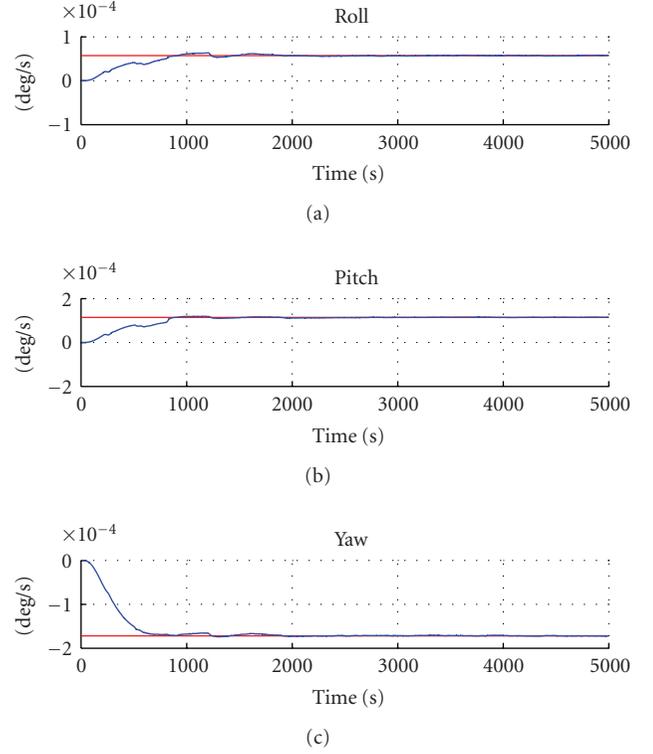


FIGURE 11: Gyro bias estimation.

It has been shown in Figure 9(b) that  $3\sigma$  boundaries of the star sensor measurement errors have the same level (high or low) around all axes during each period of time; so, the model #1 or #8 should be selected if the designed attitude estimator identifies the star sensor noise level correctly, by applying the IMM.L estimator with

$$\begin{aligned} \sigma_{\text{SK}} &= 1 \times 10^{-8}, & \sigma_{hw}^2 &= 3.2 \times 10^{-4}, \\ \sigma_{\text{SD}} &= 3.1 \times 10^{-6}, & \sigma_{\text{wheel}} &= 2.1 \times 10^{-6}, \end{aligned} \quad (77)$$

and the transition probabilities matrix  $\pi$

$$\pi = \begin{bmatrix} 0.93 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 \\ 0.01 & 0.93 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 \\ 0.01 & 0.01 & 0.93 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 \\ 0.01 & 0.01 & 0.01 & 0.93 & 0.01 & 0.01 & 0.01 & 0.01 \\ 0.01 & 0.01 & 0.01 & 0.01 & 0.93 & 0.01 & 0.01 & 0.01 \\ 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.93 & 0.01 & 0.01 \\ 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.93 & 0.01 \\ 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.93 \end{bmatrix}. \quad (78)$$

The number of the most probable model is shown in Figure 15.

According to Figure 15, it is certified that the designed IMM.L estimator identifies the star sensor measurement noise level correctly around all axes and recognizes the correct model (model #1 or model #8) at the end of maneuvers ( $t \in [180 \ 200], [280 \ 300], [380 \ 400], \dots$  sec).

TABLE 5: Comparison of attitude estimation methods.

Method	Gyro noise characteristics	Star sensor noise characteristics	1 sigma estimation error		
			Roll deg.	Pitch deg.	Yaw deg.
EKF [21]	$\sigma_r = \sqrt{10} \times 10^{-10}$ rad/sec <sup>3/2</sup> $\sigma_a = \sqrt{10} \times 10^{-7}$ rad/sec <sup>1/2</sup>	$(\sigma_x, \sigma_y) = 3.6$ to 6 arcsec $\sigma_z = 70.8$ to 180 arcsec	0.00066	0.00072	0.00499
EKF [17]	$\sigma_r = 4.8 \times 10^{-11}$ rad/sec <sup>3/2</sup> $\sigma_a = 1.3 \times 10^{-10}$ rad/sec <sup>1/2</sup>	$(\sigma_x, \sigma_y) = 10$ arcsec $\sigma_z = 20$ arcsec	0.00172	0.00155	0.00172
MMAE [18]	$\sigma_r = \sqrt{10} \times 10^{-10}$ rad/sec <sup>3/2</sup> $\sigma_a = 2.9 \times 10^{-5}$ rad/sec <sup>1/2</sup>	$(\sigma_x, \sigma_y, \sigma_z) = 3.6$ arcsec	0.0016	0.0015	0.0015
IMMAE [18]	$\sigma_r = \sqrt{10} \times 10^{-10}$ rad/sec <sup>3/2</sup> $\sigma_a = 2.9 \times 10^{-5}$ rad/sec <sup>1/2</sup>	$(\sigma_x, \sigma_y, \sigma_z) = 3.6$ arcsec	0.0013	0.0012	0.0013
EIMMAE*	$\sigma_r = 3 \times 10^{-10}$ rad/sec <sup>3/2</sup> $\sigma_a = 3 \times 10^{-6}$ rad/sec <sup>1/2</sup>	$(\sigma_x, \sigma_y) = 3.6$ to 6 arcsec $\sigma_z = 70.8$ to 180 arcsec	0.00032	0.00031	0.0027

\*The attitude estimator proposed in this paper.

**4.3. Performance Evaluation of the EIMMAE Method.** This section clearly shows the effectiveness of the proposed EIMMAE structure for attitude estimation of the stereoimaging satellite. By applying this method during stereoimaging scenario (according to Table 1), the 9-or 15-state Kalman filter should be selected at the start of maneuvers ( $t \in [100 150], [200 250], [300 350], [400 450]$  sec), and the IMM.L structure (6-state kalman filters with different star sensor measurements noise levels) should be selected at the end of maneuvers ( $t \in [180 200], [280 300], [380 400]$  sec), and then the suitable star sensor measurement noise level should be selected using the IMM.L structure in this mode. The posterior probability of EKFs ( $\mu_i(t)$ ) is shown in Figure 16. This figure shows that at the beginning of the maneuvers, the probability of the 15-state and 9- state EKFs is more than IMM.L structure and probability of the IMM.L structure is higher than the other structures while satellite is in “uniform motion” mode at the end of maneuvers.

The posterior probability of the models for star sensor measurements noise level (Table 3) at the end of maneuvers is shown in Figure 17 while the IMM.L structure selected.

The attitude estimation errors using the EIMMAE structure will be compared with IMM.L and IMM.CT method in Figures 18 and 19 with

$$\pi_{ij} = \begin{bmatrix} 0.999 & 0 & 0.001 \\ 0.008 & 0.99 & 0.002 \\ 0.008 & 0.002 & 0.99 \end{bmatrix}. \quad (79)$$

These figures verify that the EIMMAE method will provide the benefits of both IMM.CT and IMM.L methods simultaneously for attitude estimation during the large and fast maneuvers of stereoimaging scenario. In the other words, the designed EIMMAE estimator maintains the peak error less enough while the satellite is in the “maneuvering motion” mode ( $t \in [100 150], [200 250], [300 350], [400 450]$  sec) and reduces the attitude estimation errors at the end of maneuvers while the satellite is in “uniform motion” mode ( $t \in [180 200], [280 300], [380 400]$  sec) simultaneously.

The comparison results between EIMMAE structure with IMM.L and IMM.CT for attitude estimation at the end

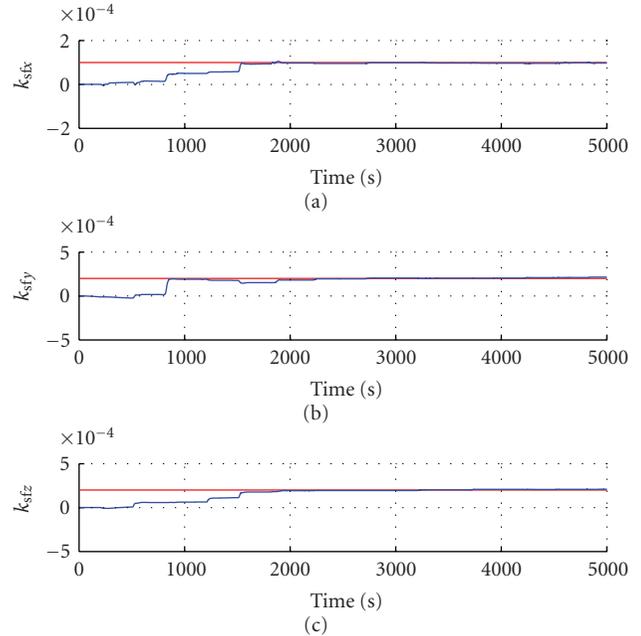


FIGURE 12: Scale factor estimation.

of maneuvers ( $t \in [180 200], [280 300], [380 400]$  sec) in stereoimaging scenario are summarized in Table 4. Entries of this table contain the average of absolute attitude estimation errors around each axis at the end of three main stereoimaging maneuvers during  $t \in [180 200], [280 300], [380 400]$  sec which have been computed around each axis using (80) as follows:

$$\Delta\varphi = \frac{1}{3} \left( \frac{1}{200 - 180} \sum_{t=180}^{200} |e_\varphi(t)| + \frac{1}{300 - 280} \sum_{t=280}^{300} |e_\varphi(t)| + \frac{1}{400 - 380} \sum_{t=380}^{400} |e_\varphi(t)| \right). \quad (80)$$

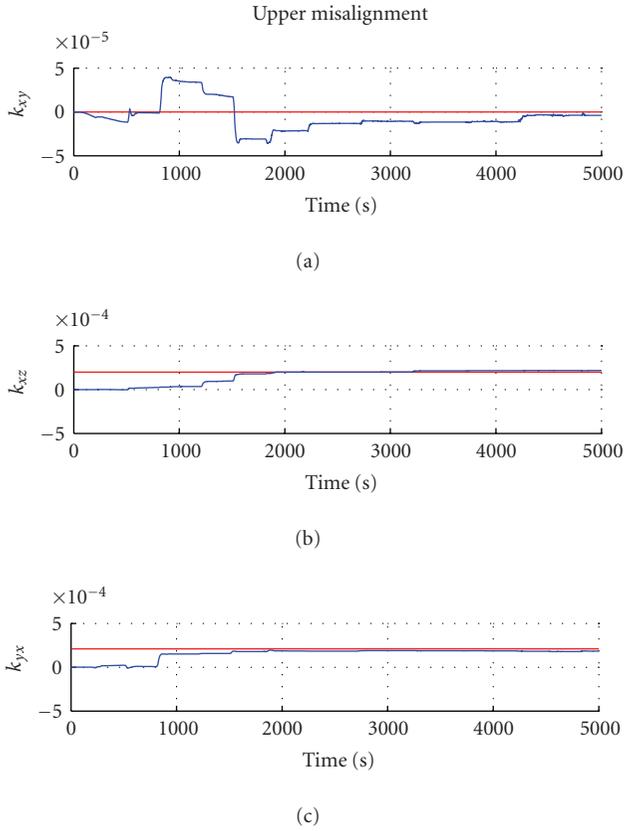


FIGURE 13: Upper misalignment error estimation.

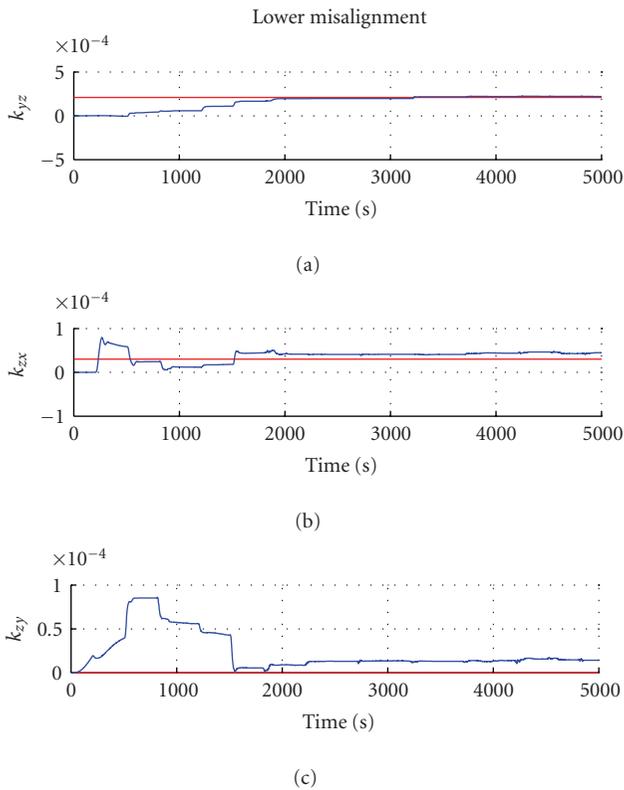


FIGURE 14: Lower misalignment error estimation.

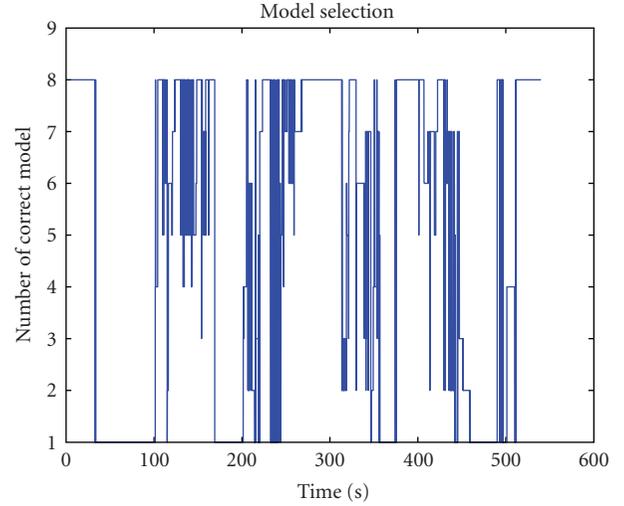


FIGURE 15: Number of correct model in IMM\_L structure.

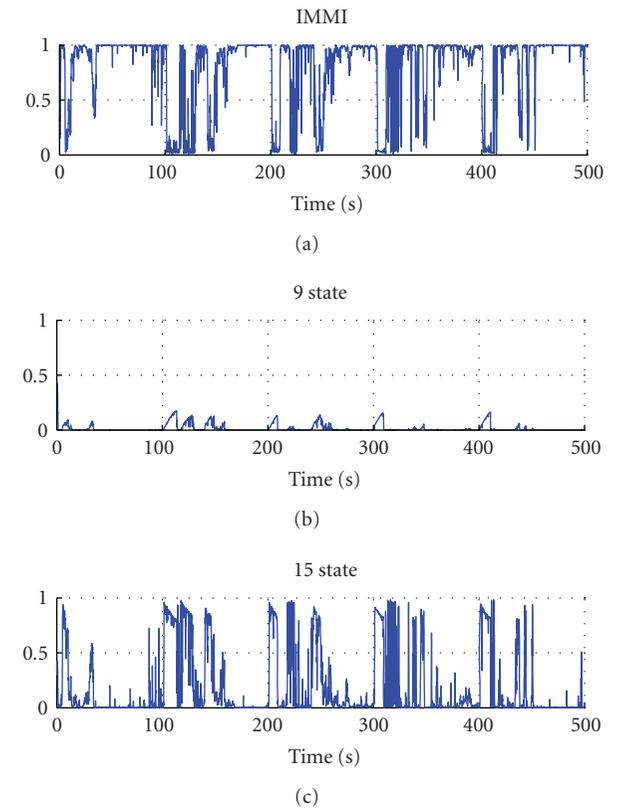


FIGURE 16: Posterior probability of the EKFs structures in EIMMAE structure.

According to Table 4, it is obvious that fulfillment of the EIMMAE method improves pointing accuracy of the satellite up to 25% in comparison with applying the IMM\_L method and improves pointing accuracy of the satellite up to 50% in comparison with applying the IMM\_CT method.

In order to show effectiveness and advantages of the proposed EIMMAE method to estimate the attitude of the satellite at the imaging moment, we will compare the

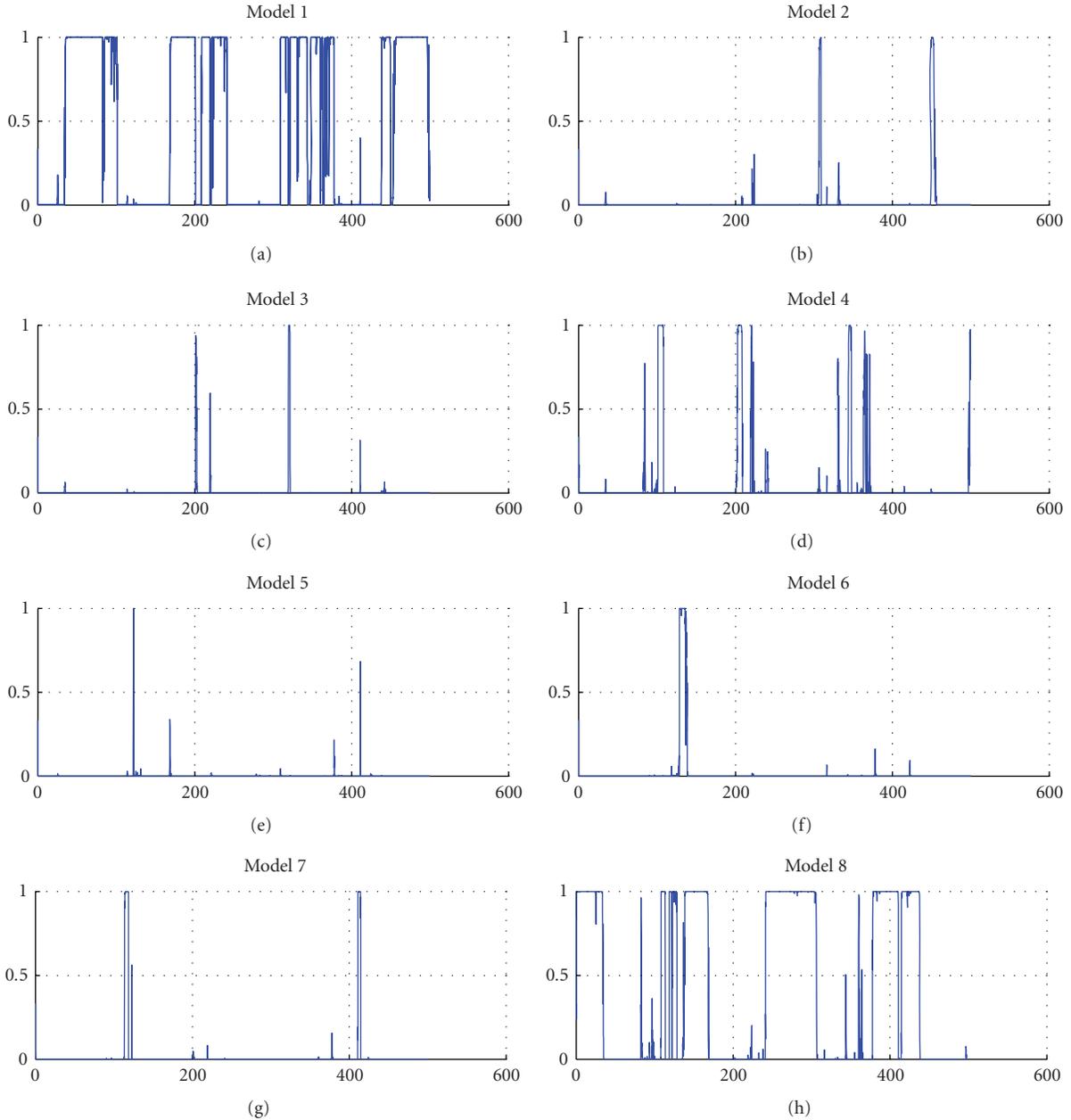


FIGURE 17: Posterior probability of the models for the star sensor measurements noise level.

obtained results with the other previous attitude estimation methods. For this purpose, we have summarized 1 sigma estimation errors of different attitude estimation methods in Table 5.

This table approves that the proposed EIMMAE algorithm improves the steady-state attitude estimation accuracy of the satellite considerably around roll and pitch axes at the end of stereovision maneuvers. According to this table, attitude estimation error around the yaw axis associated with EIMMAE method is more than the attitude estimation error around yaw axis associated with the other methods. The reason is that we have used only one star sensor with lower accuracy than the star sensors which have been used in the other mentioned methods. Therefore, in the similar

situation, by using star sensor with the same characteristics being used in the other methods, we can provide more accurate attitude knowledge by applying the EIMMAE algorithm while imaging takes place. So, fulfillment of the EIMMAE algorithm is more effective than the other previous mentioned methods to reduce the peak attitude estimation errors at the start of the maneuvers and to increase accuracy of attitude estimation at the end of stereovision maneuvers.

## 5. Conclusion

In this paper, we have designed a new adaptive attitude estimator (EIMMAE) based on interacting multiple models

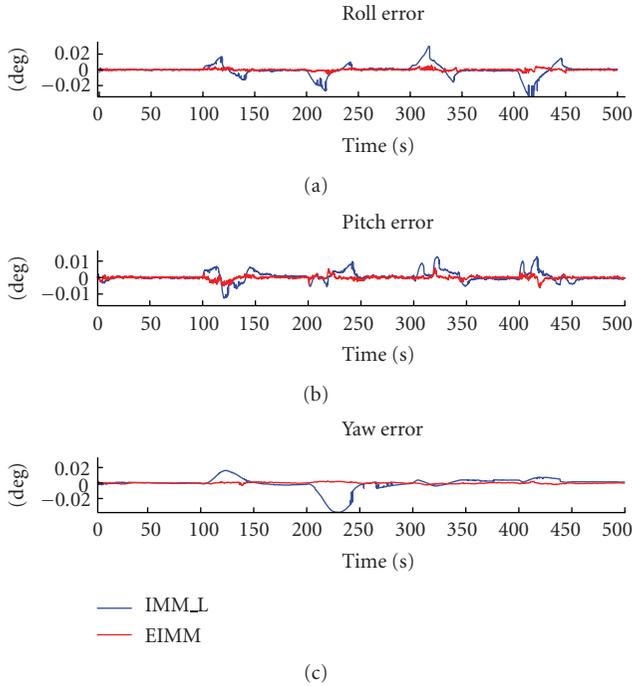


FIGURE 18: Estimation errors using IMM\_L and EIMMAE.

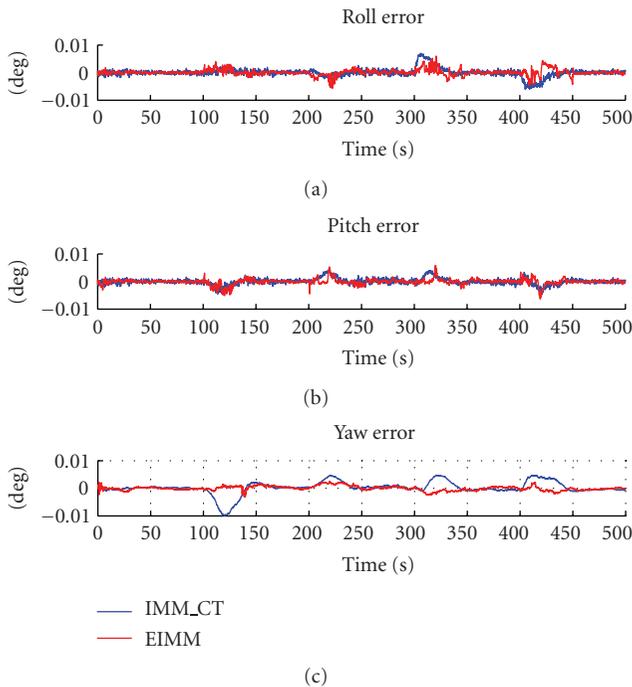


FIGURE 19: Estimation errors using IMM\_CT and EIMMAE.

(IMM) extended Kalman filters (EKF) for attitude determination of a stereoimagery satellite. The proposed algorithm consists of two different IMM estimator structures: “IMM\_CT” structure to select the suitable Kalman filter structure to maintain the peak attitude estimation errors less enough at the start of maneuvers and “IMM\_L” structure

to select the suitable star sensor measurement noise level to reduce attitude estimation errors while the satellite is in “uniform motion” mode. We have shown in Table 4, Figures 18 and 19 that by applying the proposed adaptive attitude estimation method the attitude estimation requirements of stereoimagery satellite will be satisfied simultaneously. In the other hand, we have also compared the obtained results of the proposed EIMMAE method with some other previous attitude estimation methods in Table 5. This table has verified the advantages and effectiveness of the proposed EIMMAE method to provide highly accurate attitude estimation for achieving stereoimagery scenario.

## References

- [1] F. Savopol and C. Armenakis, “Modelling of the IRS-1C satellite pan stereo-imagery using the DLT approach,” in *Proceedings of the ISPRS Commission IV Symposium on GIS, Center for Topographic Information (CTI) Geomatics*, vol. 32, no. 4, Canada, 2006.
- [2] H. Fujioka, H. Kano, and X. Chen, “Motion recovery under perspective stereo vision,” *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 1, pp. 167–182, 2009.
- [3] J. Cheolyong, K. Jedong, and K. Manbae, “3-D stereoscopic tour-into-picture by conversion of a 2-D image into stereoscopic video,” *ICIC Express Letters*, vol. 3, no. 3, pp. 555–560, 2009.
- [4] R. Sandau, “Potential and shortcoming of small satellite for topographic mapping,” German Aerospace Center (DLR), Berlin, Germany, 2004.
- [5] H. Bolandi, F. Fani Saberi, and B. Ghorbani Vaghei, “Design of a Supervisory Adaptive Attitude Control (SAAC) system for a stereo imagery satellite based on multiple model control with switching,” *International Journal of Innovative Computing, Information and Control*, vol. 6, no. 9, pp. 4675–4692, 2010.
- [6] B. S. Yaakov, X. L. Rong, and K. Thiagalingam, *Estimation with Applications to Tracking and Navigation*, John Wiley & Sons, New York, NY, USA, 2006.
- [7] H. J. Kushner, “Nonlinear filtering: the exact dynamical equations satisfied by the conditional mode,” *IEEE Transactions on Automatic Control*, vol. AC-12, no. 3, pp. 262–267, 1967.
- [8] R. L. Bellaire, *Nonlinear estimation with applications to target tracking*, Ph.D. thesis, Georgia Institute of Technology, 1996.
- [9] S. Haykin and N. D. Freitas, “Special issue on sequential state estimation,” *Proceedings of the IEEE*, vol. 92, no. 3, pp. 399–574, 2004.
- [10] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, “A new method for the nonlinear transformation of means and covariances in filters and estimators,” *IEEE Transactions on Automatic Control*, vol. 45, no. 3, pp. 477–482, 2000.
- [11] Q. M. Lam, T. Hunt, P. Sanneman, and S. Underwood, “Analysis and design of a fifteen state stellar inertial attitude determination system,” in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Austin, Tex, USA, August 2003.
- [12] M. F. Landis and J. L. Crassidis, “Nonlinear attitude filtering methods,” in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, San Francisco, Calif, USA, August 2005.
- [13] J. L. Crassidis, “Survey of nonlinear attitude estimation methods,” *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 1, pp. 12–28, 2007.

- [14] S. Marques, R. Clements, and P. Lima, "Comparison of small satellite attitude determination methods," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Lisboa, Portugal, 2000.
- [15] J. W. Murrell, "Precision attitude determination for multi-mission spacecraft," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 1978.
- [16] Q. M. Lam and J. L. Crassidis, "Precision attitude determination using a multiple model adaptive estimation scheme," in *Proceedings of the IEEE Aerospace Conference*, Big Sky, Mont, USA, June 2007.
- [17] Q.M. Lam and J. L. Crassidis, "Evaluation of a multiple model adaptive estimation scheme for space vehicle's enhanced navigation solution," in *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, Hilton Head, SC, USA, August 2007.
- [18] M. Sidi, *Spacecraft Dynamics and Control*, Cambridge University Press, Cambridge, UK, 1997.
- [19] B. Reimer, *Control momentum gyro actuator for small satellite applications*, M.S. thesis, University of Stellenbosch, 2005.
- [20] J. L. Crassidis and J. L. Junkins, *Optimal Estimation of Dynamic Systems*, Chapman & Hall/CRC, Boca Raton, Fla, USA, 2004.
- [21] M. D. Shuster, "Kalman filtering of spacecraft attitude and the QUEST model," *The Journal of the Astronautical Sciences*, vol. 38, no. 3, pp. 377–393, 1990.
- [22] H. A. P. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markovian switching coefficients," *IEEE Transactions on Automatic Control*, vol. 33, no. 8, pp. 780–783, 1988.
- [23] S. Kanev and M. Verhaegen, "Controller reconfiguration for non-linear systems," *Control Engineering Practice*, vol. 8, no. 11, pp. 1223–1235, 2000.
- [24] P. S. Maybeck and R. D. Stevens, "Reconfigurable flight control via multiple model adaptive control methods," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 27, no. 3, pp. 470–479, 1991.
- [25] X. R. Li and Y. Bar-Shalom, "Design of an interacting multiple model algorithm for air traffic control tracking," *IEEE Transactions on Control Systems Technology*, vol. 1, no. 3, pp. 186–194, 1993.
- [26] Y. Zhang and X. R. Li, "Detection and diagnosis of sensor and actuator failures using IMM estimator," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 4, pp. 1293–1312, 1998.
- [27] M. Efe and D. P. Atherton, "The IMM approach to the fault detection problem," in *Proceedings of the 11th IFAC Symposium on System Identification*, Fukuoka, Japan, 1997.

## Research Article

# Decentralized Model Predictive Control for Cooperative Multiple Vehicles Subject to Communication Loss

**Hojjat A. Izadi, Brandon W. Gordon, and Youmin Zhang**

*Department of Mechanical and Industrial Engineering, Concordia University, 1515 St. Catherine W., Montreal, QC, Canada H3G 2W1*

Correspondence should be addressed to Youmin Zhang, ymzhang@encs.concordia.ca

Received 1 July 2010; Revised 24 November 2010; Accepted 28 February 2011

Academic Editor: Yu Gu

Copyright © 2011 Hojjat A. Izadi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The decentralized model predictive control (DMPC) of multiple cooperative vehicles with the possibility of communication loss/delay is investigated. The neighboring vehicles exchange their predicted trajectories at every sample time to maintain the cooperation objectives. In the event of a communication loss (packet dropout), the most recent available information, which is potentially delayed, is used. Then the communication loss problem changes to a cooperative problem when random large communication delays are present. Such large communication delays can lead to poor cooperation performance and unsafe behaviors such as collisions. A new DMPC approach is developed to improve the cooperation performance and achieve safety in the presence of the large communication delays. The proposed DMPC architecture estimates the tail of neighbor's trajectory which is not available due to the large communication delays for improving the performance. The concept of the tube MPC is also employed to provide the safety of the fleet against collisions, in the presence of large intervehicle communication delays. In this approach, a tube shaped trajectory set is assumed around the trajectory of the neighboring vehicles whose trajectory is delayed/lost. The radius of tube is a function of the communication delay and vehicle's maneuverability (in the absence of model uncertainty). The simulation of formation problem of multiple vehicles is employed to illustrate the effectiveness of the proposed approach.

## 1. Introduction

To address the intervehicle communication loss/delay in a network of cooperative multiple vehicles, a previously developed delay-dependent decentralized model predictive control (DMPC) architecture in [1] is modified and extended to subsystems with generic class of dynamics, subject to intervehicle communication packet loss. The proposed decentralized model predictive control (DMPC) features two main contributions. The first contribution is the development of a new DMPC approach that estimates the trajectory of the neighboring vehicles for the tail of the prediction horizon which would otherwise not be available due to the communication losses leading to random communication delays. In this approach, the tail of the cost function is estimated by adding extra decision variables in the cost function.

A relatively small amount of existing works have investigated the implementation issues such as communication loss associated with the exchange of trajectory information, but

so far a few works have proposed a systematic tail estimation process to compensate for large delays. For example, in [2–5], it is assumed that the neighboring vehicles remain stationary at the last delayed states broadcasted by them. Such assumptions may yield poor performance for large communication delays, because the stationary state vector is not necessarily an accurate estimate of a time-varying state vector. Similar issues are investigated in [6], where for the hardware implementation of a robust DMPC to wheeled robots, both computation and communication delays are considered. To account explicitly for the time delays a model of the vehicle is used to estimate their state vector when required. Then the uncertainty arising from this estimation is taken into account by accommodating that into the effective disturbance force used for constraint tightening. Also, in [7], the stability, feasibility, and computation issues of a distributed MPC are investigated. The main practical attraction of the proposed architecture in [7] is that it allows terminating the computations at the end of each sampling

period even if convergence is not achieved; this feature facilitates the real-time implementation of DMPC.

The second contribution of this paper is an extension of the tube-based model predictive control (MPC) approach [8, 9] for the case of the large communication delays in order to guarantee the safety of the fleet against possible collisions during formation control problems. The concept of the tube MPC in the existing works [8, 9] is normally used to calculate a robust bound on the states due to model uncertainty, whereas in this paper the approach is used to calculate bounds on the exchanged neighbor trajectories when large intervehicle communication delays exist.

In [10], the wireless communication packet loss/delay is considered; once the packet loss/delay occurs, the previous available trajectory of the faulty vehicle—whose trajectory is delayed—is extrapolated to predict the future reference trajectory. Also, in [11], the communication failure in formation flight of multiple vehicles leads to break in the communicated messages that force the fleet to redefine the communication graph.

This paper is organized as follows. Section 2 presents a general formulation of the delay-free decentralized model predictive controller and the corresponding algorithm. In Section 3 a new delay-compensated DMPC (DC-DMPC) is developed to address the communication delays arising from communication losses. A safety guarantee method is also developed based on the concept of tube MPC. In Section 4 the proposed algorithms are tested through simulation of a leader-follower formation problem for a fleet of unmanned vehicles.

## 2. Decentralized Model Predictive Control

Consider a team of cooperative vehicles with uncoupled dynamics. Each vehicle is equipped with three main components: (1) measurement sensors, (2) communication device, and (3) computation resource. The measurement sensors of each vehicle measure its own states assuming full state measurement is available. The communication device is used to gather the information from the neighboring vehicles and communicate with human operators. Using the computation resource, each vehicle solves a decentralized optimization problem at each sampling time based on its current measured states (from sensors) and the neighbor's predicted state trajectories (provided through communication). It is further assumed that there is no sensor error, model uncertainty, or communication noise. These assumptions allow focusing on the main problem concerning intervehicle communication loss/delay. The developed algorithms do not rely on such assumptions; in fact, these assumptions are not restrictive for extending the proposed algorithms to nonideal situations. It is worth mentioning that even if a perfect model of neighboring vehicle is available still there is a possibility of mismatch between the predicted neighboring vehicle's trajectory and the actual ones, because the neighboring vehicles may use a particular communication topology which leads to different DRHC optimization problems and hence different solutions. This problem can be overcome only by a centralized approach which is beyond the approach of this

paper, because in a centralized approach only one communication topology which connects all the team members exists. Then, it is intended to focus on such mismatches arising from decentralized nature of the problem rather than mismatches coming from model uncertainty, and so forth.

Moreover, the terms *agent*, *vehicle*, *team*, *member*, and *subsystem* bear the same meaning. And so do the terms *group*, *team*, *fleet*, and *network*.

**2.1. Interaction Graph Topologies.** *Graph theory* is a relevant mathematical tool for modeling and analyzing the interactions among the cooperative vehicles (see for instance, [12, 13]). *Graph topology* is described by two basic elements: *nodes* and *edges*. *Nodes* represent the vehicles, and an *edge* between two nodes denotes the interaction between those vehicles. The *interaction graph* is denoted by  $\mathbb{G}$  and represented as follows:

$$\mathbb{G} = \{\mathbb{V}, \mathbb{E}\}, \quad (1)$$

where  $\mathbb{V}$  is the set of nodes (vehicles) and  $\mathbb{E} \subseteq \mathbb{V} \times \mathbb{V}$  is the set of edges  $(i, j)$ , with  $i, j \in \mathbb{V}$ . Hence, considering a set of  $N_v$  vehicles cooperating to perform a common mission, the  $i$ th vehicle in the team is assigned to the  $i$ th node of the graph. If an edge  $(i, j)$  connecting the  $i$ th and  $j$ th node is present in  $\mathbb{E}$ , it means that the  $i$ th and  $j$ th vehicles have an interaction. This relation is termed as neighborhood for  $i$ th and  $j$ th vehicles, and it is said that

- (i)  $i$ th and  $j$ th vehicles are *neighboring* vehicles;
- (ii)  $i$ th vehicle is a *follower* of the  $j$ th vehicle;
- (iii)  $j$ th vehicle is a *leader* of the  $i$ th vehicle.

Also, it is assumed that the interaction graph is *directed*, that is,  $(i, j) \in \mathbb{E}$  does not necessarily imply  $(j, i) \in \mathbb{E}$ , unless it appears explicitly in  $\mathbb{E}$ . Further, let  $N_l^i$  and  $N_f^i$  denote the number of the *leaders* and *followers* of vehicle  $i$ , respectively.

In this paper, the information exchange structure coincides the control structure; in fact, an edge  $(i, j)$  indicates that (1) vehicle  $j$  sends information on its planned trajectory to  $i$  and (2) the cost function of vehicle  $j$  is coupled to states of vehicle  $i$ . Also, it is assumed that the *interaction graphs* have a particular fixed structure and is set by the human operator prior to the mission.

**2.2. DMPC Notation and Terminology.** With model predictive control, a cost function is optimized over a finite time window called the *prediction horizon*,  $N$ . The first of the computed optimal inputs is applied to the plant. The reader is referred to [14] for a comprehensive review of MPC schemes.

In this paper, the DMPC architecture is formulated based on the quasi-infinite-RHC formulation [15] where the key is to tune the *terminal cost* so that the closed-loop MPC has the property of an infinite horizon controller. In this approach the states are driven to a neighborhood of the origin where it is a positively invariant set under a feedback *terminal controller* for linearized system.

The discrete timing is then shown by  $k$  where  $k \in \mathbb{N}$ . As mentioned for the DMPC scheme presented in this paper the neighboring vehicles exchange their predicted state trajectories to provide cooperation. However, as we will see later for the time periods when the updated leader's state trajectory is not available, it is estimated by the followers. Hence, considering the trajectory of vehicles, those calculated by themselves and those estimated by followers, an appropriate notation is required to present all these possible trajectories (actual, predicted, and estimated). The possible state vectors are introduced as follows:

- (i)  $\mathbf{x}^i(k)$ : the *actual* state vector of the  $i$ th vehicle at time step  $k$ ;
- (ii)  $\mathbf{x}_k^i(p)$ : the state vector of the  $i$ th vehicle at time  $p$ , *predicted* by the  $i$ th vehicle at time step  $k$ , where  $p = 0, 1, 2, \dots, N$ ;
- (iii)  $\mathbf{x}_k^{i,j}(p)$ : the state vector of the  $i$ th vehicle at time step  $k$ , *estimated* by the  $j$ th vehicle at time step  $k$  where  $p = 0, 1, 2, \dots, N$ .

Further, the sequence of states over the prediction horizon is called the state trajectory of vehicle  $i$  calculated by itself and is represented by  $\mathbf{x}_k^i(\cdot)$ :

$$\mathbf{x}_k^i(\cdot) = \{\mathbf{x}_k^i(p) \mid p = 0, 1, 2, \dots, N\}, \quad (2)$$

$$\mathbf{u}_k^i(\cdot) = \{\mathbf{u}_k^i(p) \mid p = 0, 1, 2, \dots, N-1\}.$$

Then let the following represent the concatenated state and input trajectories of the leader(s) of the  $i$ th vehicle at time  $k$ :

$$\mathbf{x}_k^{-i}(\cdot) = \{\mathbf{x}_k^j(\cdot) \mid j \in \mathbb{V}, (i, j) \in \mathbb{E}\}, \quad (3)$$

$$\mathbf{u}_k^{-i}(\cdot) = \{\mathbf{u}_k^j(\cdot) \mid j \in \mathbb{V}, (i, j) \in \mathbb{E}\}.$$

If the trajectory is defined on an interval which is different from  $[k, k+N]$  by other vehicles, then the beginning and end times are indicated as  $\mathbf{x}_k^{i,j}(b : e)$ , that is,

$$\mathbf{x}_k^{i,j}(b : e) = \{\mathbf{x}_k^{i,j}(p) \mid p = b, b+1, \dots, e, j \in \mathbb{V}, (i, j) \in \mathbb{E}\}, \quad (4)$$

where  $[b : e]$  is the interval on which the trajectory is defined.

**2.3. Delay-Free DMPC Formulation.** In this section, a brief overview of a delay-free DMPC problem and its implementation are described. More details can be obtained from [6, 12, 16]. For the DMPC scheme presented in this paper, the predicted trajectories are exchanged instead of being estimated, thereby reducing the online computational time. Figure 1 shows the delay-free intervehicle communication between two neighboring vehicles and the information exchange at time step  $k$ .

In Figure 1, it is assumed that the current leader's predicted trajectories are always available instantly to their followers. However, at least one step delay has to be considered as the current predicted trajectories are not available

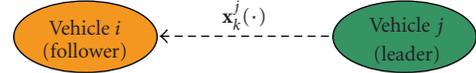


FIGURE 1: The delay-free intervehicle communication between two neighbors.

instantly, even if an infinite communication bandwidth is used. Then, the *information set* of the  $i$ th vehicle for the case of communication delay-free DMPC is introduced as follows:

$$\Gamma^i(k) = \{\mathbf{x}^i(k), \mathbf{x}_{k-1}^{-i}(\cdot)\}, \quad (5)$$

where set  $\Gamma^i(k)$  contains the updated information available to the  $i$ th vehicle at time step  $k$  and is referred to the *information set* in this paper. This collects (1) the instant state vector of the  $i$ th vehicle and (2) the concatenated state trajectory of neighbors calculated at the previous time step,  $\mathbf{x}_{k-1}^{-i}(\cdot)$ .

For the particular case of formation control, the delay-free decentralized cost function for the  $i$ th vehicle in the team at time  $k$  is defined as follows:

$$\begin{aligned} J^i(\Gamma^i(k), \mathbf{u}_k^i(\cdot)) &= \sum_{p=0}^{N-1} \left( \|\mathbf{x}_k^i(p) - \mathbf{x}^{T,i}\|_{\mathbf{Q}}^2 + \|\mathbf{u}_k^i(p)\|_{\mathbf{R}}^2 \right) + \|\mathbf{x}_k^i(N) - \mathbf{x}^{T,i}\|_{\mathbf{P}}^2 \\ &+ \sum_{j|(i,j) \in \mathbb{E}} \left( \sum_{p=0}^{N-1} \|\mathbf{x}_k^i(p) - \mathbf{x}_{k-1}^j(p) - \mathbf{r}^{i,j}\|_{\mathbf{S}}^2 \right), \end{aligned} \quad (6)$$

where  $\|\mathbf{x}\|_{\mathbf{Q}}^2 = \mathbf{x}'\mathbf{Q}\mathbf{x}$  and  $\mathbf{P}$ ,  $\mathbf{Q}$ ,  $\mathbf{R}$ , and  $\mathbf{S}$  are positive definite and symmetric matrices,  $\mathbf{x}^{T,i}$  is the state vector of target (waypoint) of vehicle  $i$ , and  $\mathbf{r}^{i,j}$  is the vector of desired relative position between vehicles  $i$  and  $j$ .

**2.3.1. Delay-Free DMPC Problem.** Assume that the following describes the discrete-time dynamics of the (homogeneous) vehicle  $i$ :

$$\mathbf{x}^i(p+1) = f(\mathbf{x}^i(p), \mathbf{u}^i(p)), \quad \mathbf{x}^i(0) = \mathbf{x}^i(0). \quad (7)$$

Then, the delay-free DMPC problem  $\mathcal{P}^i(k)$  is defined for any vehicle  $i \in \mathbb{V}$  at time step  $k$  as follows.

**Problem 1. Delay-Free DMPC Problem  $\mathcal{P}^i(k)$ .** Calculate

$$J^i(\Gamma^i(k)) = \text{Min}_{\{\mathbf{u}_k^i(\cdot), \mathbf{x}_k^i(\cdot)\}} J^i(\Gamma^i(k), \mathbf{u}_k^i(\cdot)) \quad (8)$$

subject to (for  $p = 0, 1, 2, \dots, N-1$ )

$$\mathbf{x}_k^i(p+1) = f(\mathbf{x}_k^i(p), \mathbf{u}_k^i(p)); \quad \mathbf{x}_k^i(0) = \mathbf{x}^i(k), \quad (9a)$$

$$\mathbf{x}_k^i(p) \in \mathbb{X}^i, \quad \mathbf{u}_k^i(p) \in \mathbb{U}^i, \quad (9b)$$

$$\mathbf{x}_k^i(N) \in \mathbb{X}_f^i, \quad (9c)$$

where  $\mathbb{X}^i$ ,  $\mathbb{U}^i$ , and  $\mathbb{X}_f^i \subseteq \mathbb{X}^i$  denote the set of admissible states, inputs, and final states (terminal region), respectively, for the  $i$ th vehicle.

Given  $\mathbb{E}$  and initial condition of vehicle  $i$ , do:

- (1)  $k \leftarrow 0$ .
- (2) Measure  $\mathbf{x}^i(k)$  and update the information set  $\Gamma^i(k)$  based on (4).
- (3) Solve  $\mathcal{P}^i(k)$  and predict the control and state trajectories  $\mathbf{u}_k^i(\cdot)$  and  $\mathbf{x}_k^i(\cdot)$ .
- (4) Send the predicted state trajectory  $\mathbf{x}_k^i(\cdot)$  to all followers  $j$  (where  $(j, i) \in \mathbb{E}$ ).
- (5) Apply the control action  $\mathbf{u}_k^i(0)$  to vehicle  $i$ .
- (6) Receive the trajectory  $\mathbf{x}_k^j(\cdot)$  from leader(s)  $j$  (where  $(i, j) \in \mathbb{E}$ ).
- (7)  $k \leftarrow k + 1$ . GOTO step 2.

ALGORITHM 1: Delay-free DMPC.

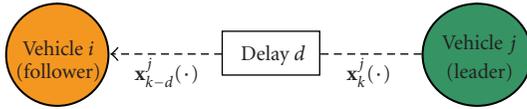


FIGURE 2: Delayed intervehicle communication between two neighboring vehicles.

**2.3.2. Delay-Free DMPC Algorithm.** Each vehicle  $i$  at each sampling period solves the decentralized problem  $\mathcal{P}^i(k)$ . The output of the optimization problem  $\mathcal{P}^i(k)$  is the predicted input and state trajectories of vehicle  $i$ . The generated state trajectory is sent to followers for cooperation purposes, and the first optimal input is applied to the individual vehicle  $i$ . The following algorithm is presented for the online implementation of the delay-free DMPC problem above. The algorithm is formulated for the  $i$ th vehicle as in Algorithm 1.

*Initialization.* For  $k = 0$  at steps 2 and 3, assume  $N_i^j = 0$  (because no information from leader(s) is available).

Algorithm 1 is repeated until the assigned targets are reached. The targets are assumed to be known and assigned to each agent *a priori*.

### 3. Delay-Compensated DMPC (DC-DMPC)

This section develops a new delay-compensated DMPC (DC-DMPC) approach. The safety issues are also discussed.

**3.1. Delayed Cost Function and Information Set.** Assume at time step  $k$  the vehicle  $i$  receives the information from neighbor  $j$  with time-delay  $d$ , where  $d \in \mathbb{N}$  and  $d \geq 1$ , see Figure 2. Then, the term large communication delays denotes the time delays which are larger than the sampling period, that is,  $d > 1$ .

When a vehicle receives the delayed information from neighbors the information set is updated as follows (compare with (5)):

$$\Gamma_D^i(k) = \{\mathbf{x}^i(k), \mathbf{x}_{k-d}^{-i}(\cdot)\}. \quad (10)$$

The *information set*  $\Gamma_D^i(k)$  represents updated information available to the  $i$ th vehicle at time step  $k$ . It implies at time step  $k$  that each vehicle  $i$  has access to its own delay-free

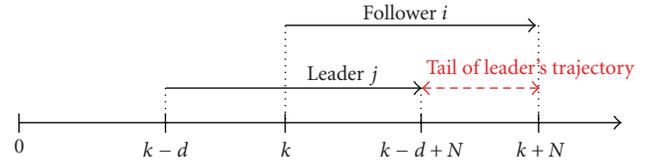


FIGURE 3: Tail of the leader's trajectory in the presence of large communication delay  $d$ .

information and delayed information from neighbors. Then, the decentralized delayed cost function for all  $i \in \mathbb{V}$  at time step  $k$  is defined as follows (compare with (6)):

$$\begin{aligned} J_D^i(\Gamma_D^i, \mathbf{u}_k^i(\cdot)) &= \sum_{p=0}^{N-1} \left( \|\mathbf{x}_k^i(p) - \mathbf{x}^{T,i}\|_{\mathbf{Q}}^2 + \|\mathbf{u}_k^i(p)\|_{\mathbf{R}}^2 \right) \\ &+ \|\mathbf{x}_k^i(N) - \mathbf{x}^{T,i}\|_{\mathbf{P}}^2 \\ &+ \sum_{j|(i,j) \in \mathbb{E}} \left( \sum_{p=0}^{N-d} \|\mathbf{x}_k^i(p) - \mathbf{x}_{k-d}^j(p) - \mathbf{r}^{i,j}\|_{\mathbf{S}}^2 \right). \end{aligned} \quad (11)$$

The communication delay affects the cooperation cost for the periods when the leader's trajectory is not available. In fact, for the tail of the cost function during  $[N-d : N]$  the leader's trajectory is not available due to communication delay to calculate the cost function. More precisely, assume at time step  $k$  the vehicle  $i$  receives the information from neighbor  $j$  with time-delay  $d$ , where  $d \in \mathbb{N}$  and  $d \geq 1$ . Then, the trajectory of neighbor  $j$  for only the time interval  $[k-d, k+N-d]$  is available to vehicle  $i$ , while according to cost function of (6) vehicle  $i$  needs the trajectory of neighbor  $j$  for the entire interval  $[k, k+N]$ . Hence for the portion  $[k+N-d, k+N]$  the trajectory of  $j$  is not available due to the delay (see Figure 3). When the time delay is small this lack of information is not important, but for large communication delays the tail of cost function during  $[k+N-d, k+N]$  becomes large and as shown by simulations in Section 4 it can lead to poor performance and unsafe behaviors (see also [13, 17]). One remedy to this problem is proposed here by estimating the tail of the cost function by including extra decision variables in the cost function.

**3.2. Delay-Compensated Cost Function.** The main idea with the proposed cost function is that it replaces the missing information due to communication loss, with the decision variables. Then, the decision variables are used to approximate the missing information. More specifically, communication losses prevent access to updated leader's trajectory, and then the available delayed neighbor's trajectory is utilized. However, the delayed leader's trajectories do not cover the entire time slot required by follower to calculate its cost function; hence, a delay-compensated cost function is proposed as follows (for the  $i$ th vehicle in the team at time step  $k$ ):

$$\begin{aligned} J_{DC}^i & \left( \Gamma_D^i(k), \mathbf{u}_k^i(\cdot), \mathbf{x}_k^{-i}(N-d:N), \mathbf{u}_k^{-i}(N-d:N) \right) \\ & = J_D^i \left( \Gamma_D^i(k), \mathbf{u}_k^i \right) \\ & \quad + J_{\text{Tail}}^i \left( \mathbf{x}_k^{-i}(N-d:N), \mathbf{u}_k^{-i}(N-d:N) \right), \end{aligned} \quad (12)$$

where  $J_D^i$  is calculated from (11) and

$$\begin{aligned} J_{\text{Tail}}^i & \left( \mathbf{x}_k^{-i}(N-d:N), \mathbf{u}_k^{-i}(N-d:N) \right) \\ & = \sum_{j|(i,j) \in \mathbb{E}} \left[ \sum_{p=N-d+1}^N \left( \left\| \mathbf{x}_k^{j,i}(p) - \mathbf{x}_k^{j,i}(p) - \mathbf{r}^{i,j} \right\|_S^2 \right. \right. \\ & \quad \left. \left. + \left\| \mathbf{x}_k^{j,i}(p) - \mathbf{x}^{T,j} \right\|_Q^2 + \left\| \mathbf{u}_k^{j,i}(p) \right\|_R^2 \right) \right. \\ & \quad \left. + \left\| \mathbf{x}_k^{j,i}(N) - \mathbf{x}^{T,j} \right\|_P^2 \right]. \end{aligned} \quad (13)$$

The subscript "D" stands for "Delay". The delay-compensated decentralized cost function  $J_{DC}^i$  includes two main parts.

- (1) The delayed cost  $J_D^i$  which includes the cost of local vehicle  $i$  and the cost of neighboring vehicles calculated from the delayed information. Literally, this part is calculated using the information available from communication (and measurement sensors for local vehicle).
- (2) The tail cost  $J_{\text{Tail}}^i$  (13) which calculates the cost associated with the neighbors over the tail of the cost function during  $[N-d+1 : N]$  where information is not available due to the communication loss/delay. In fact, the added extra decision variables are used to calculate this part.

Although it is assumed that there is model uncertainty and communication noise, the lack of updated information due to communication loss will impose an uncertainty on the updated leader's trajectory, because, in the event of communication loss, the followers will utilize the leader's delayed trajectory, which may differ from the updated unavailable trajectories. Such uncertainty which is derived from the mismatch between updated and delayed trajectories may lead to poor performance of the decentralized DMPC.

**3.3. Safety Guarantee Using Tube DMPC.** For the particular case of formation control, consider a group of vehicles that are required (1) to keep certain relative positions (local objective) and (2) to visit a set of waypoints (global objective). In general, incorporating the collision avoidance constraint into the optimization problem of DMPC is not trivial because of the nonconvexity nature of the distance constraint. To address this problem, in [18] a hybrid rule-based extension of the decentralized model predictive control (DMPC) is proposed to avoid possible collisions. Also, in [19] a mixed integer linear programming (MLIP) approach is utilized to handle the nonconvex collision avoidance constraint using a DMPC architecture.

The problem becomes even more serious when the large communication delays exist because communication delay leads to a lack of updated information on the trajectory of neighboring vehicles and this can make the formation unsafe and put the team in jeopardy. Hence, a discussion on the collision avoidance is required.

In practice, normally the desired distance in formation control is chosen large enough to ensure safety against overshoots and uncertainties. In this paper another approach which is applicable to both delay-free and delayed situations is proposed. In this approach, instead of using an assumed trajectory for neighboring vehicles, a tube-shaped trajectory is assumed around the trajectory of leader, where the tube radius is the radius of the reachable set of leaders. This way, the leader is put in a protection zone where the followers cannot enter. The radius of tube is a function of maneuverability and communication delay; the smaller the communication delays the thinner the tube. Note that at the end the desired relative distance in the formation is chosen to be larger than the tube radius. This is the main distinction between this work and the cooperative or noncooperative approaches aiming at online estimation of the reachable set. In contrast, here the tube is calculated offline and is used for the online purposes.

The idea of tube MPC was first used to calculate a robust bound on the states for uncertain systems [8, 9].

To avoid conservative reachability sets (thick tubes) the maneuverability of each vehicle is restricted, and then the reachable set of each vehicle is calculated; in fact, the leader imposes an input constraint to its optimization problem, that is, at any time instant the computed input trajectories of leader do not deviate too far from the previous one.

Note that, although restricting the maneuverability can lead to less conservative results, it can be restrictive to application where an agile vehicle is needed. Hence, for applications where, for example, slow moving ground vehicles are used such as automated highway systems [20] or mobile robots [21, 22], this approach can be beneficial. In such application since aggressive maneuvers are not required, restricting the maneuverability is not prohibitive, and hence the proposed approach is well suited.

**3.3.1. Tube Formulation and Calculation.** This section presents a numerical method for offline calculation of the tube for subsystems with generic dynamics. Any other computational or analytical reachability set calculation method

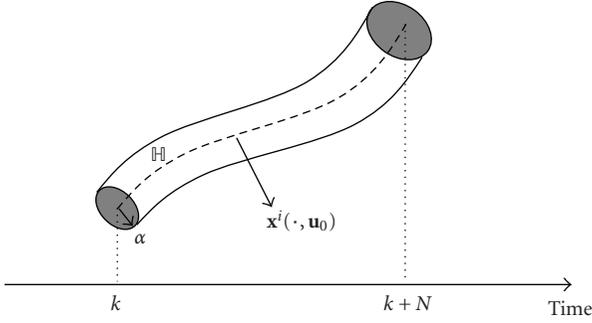


FIGURE 4: A tube around a nominal trajectory, of vehicle  $i$ .

may be used for this purpose. The main advantage of the presented tube calculation algorithm is that it can be computed offline and does not impose significant online computation load. Besides, the available analytical bounds are usually too conservative and are not in the desired format for tube analysis (often a bound on the norm of states is available).

Figure 4 shows the tube  $\mathbb{H}$  around a nominal trajectory  $\mathbf{x}^i(\cdot, \mathbf{u}_0)$ . The tube  $\mathbb{H}$  is formulated as follows (see also [23]):

$$\begin{aligned} \mathbb{H}_k^i = \{ & (p, \hat{\mathbf{x}}^i) \in [0 : N] \\ & \times \mathbb{R}^n \mid |\hat{\mathbf{x}}^i(p, \mathbf{u}) - \mathbf{x}^i(p, \mathbf{u}_0)| < \alpha_k^i, \mathbf{u} \in \mathbb{U} \}, \end{aligned} \quad (14)$$

where  $|\cdot|$  represents the component-wise absolute value of vector and  $\alpha_k^i$  denotes the radius of tube  $\mathbb{H}$  around the trajectory of vehicle  $i$  at time step  $k$ .

The following theorem represents a method for calculating the tube for a generic class of systems. Roughly speaking, tube is a generalization of reachability set; the reachability set is normally calculated around an initial state; however, tube is calculated around a trajectory of state vector. In other words, the deviation from some nominal trajectory over a given time is called tube.

**Theorem 1.** Assume that at time step  $k$  the  $d$  step delayed trajectory of leader  $j$  (i.e.,  $\mathbf{x}_{k-d}^j(\cdot)$ ) is available to vehicle  $i$ . If

$$|\mathbf{u}_k^j(p) - \mathbf{u}_{k-d}^j(p+d)| \leq \beta^j, \quad p = 0, 1, 2, \dots, N-d, \quad (15)$$

then the trajectory of vehicle  $j$  at time step  $k$  belongs to the tube around  $\mathbf{x}_{k-d}^j(\cdot)$ ; the tube is formulated as:

$$\begin{aligned} \mathbb{H}_k^{j,i} = \{ & (p, \hat{\mathbf{x}}_{k-d}^j) \in [0 : N-d] \\ & \times \mathbb{R}^n \mid |\hat{\mathbf{x}}_{k-d}^j(p) - \mathbf{x}_{k-d}^j(p+d)| < \alpha_k^{j,i}(p) \}, \end{aligned} \quad (16)$$

where  $\mathbb{H}_k^{j,i}$  represents the tube around the trajectory of vehicle  $j$ , calculated by vehicle  $i$  at time step  $k$ . Also,  $\alpha_k^{j,i}$  denotes the tube radius and is the solution of the following maximization problem.

**Problem 2.** Given  $\beta^j$ , calculate for all  $p = 0, 1, 2, \dots, N$

$$\begin{aligned} & \alpha_k^{j,i}(p) \\ & = \text{Max}_{\{\hat{\mathbf{x}}_{k-d}^j(\cdot), \hat{\mathbf{u}}_{k-d}^j(\cdot), \mathbf{x}_{k-d}^j(\cdot), \mathbf{u}_{k-d}^j(\cdot)\}} \left[ \sum_{p=0}^{N-d} \left[ f(\hat{\mathbf{x}}_{k-d}^j(p), \hat{\mathbf{u}}_{k-d}^j(p)) \right. \right. \\ & \quad \left. \left. - f(\mathbf{x}_{k-d}^j(p), \mathbf{u}_{k-d}^j(p)) \right] \right] \end{aligned}$$

subject to

$$\hat{\mathbf{x}}_{k-d}^j(p+1) = f(\hat{\mathbf{x}}_{k-d}^j(p), \hat{\mathbf{u}}_{k-d}^j(p)),$$

$$\hat{\mathbf{x}}_{k-d}^j(t_{k-d}) = \mathbf{x}^j(t_{k-d}) \in \mathbb{X},$$

$$\mathbf{x}_{k-d}^j(p+1) = f(\mathbf{x}_{k-d}^j(p), \mathbf{u}_{k-d}^j(p)),$$

$$\mathbf{x}_{k-d}^j(t_{k-d}) = \mathbf{x}^j(t_{k-d}) \in \mathbb{X},$$

$$\hat{\mathbf{x}}_{k-d}^j(p) \in \mathbb{X}; \quad \mathbf{x}_{k-d}^j(p) \in \mathbb{X},$$

$$\hat{\mathbf{u}}_{k-d}^j(p) \in \mathbb{U}; \quad \mathbf{u}_{k-d}^j(p) \in \mathbb{U},$$

$$|\hat{\mathbf{u}}_{k-d}^j(p) - \mathbf{u}_{k-d}^j(p)| \leq \beta^j. \quad (17)$$

*Proof.* At time step  $k$  vehicle  $j$  uses the input trajectory  $\mathbf{u}_k^j(\cdot)$  which yields the state trajectory  $\mathbf{x}_k^j(\cdot)$ , then the solution of the differential equation (8) is calculated as:

$$\mathbf{x}_k^j(p) = \mathbf{x}^j(k) + \sum_{s=0}^{p-1} f(\mathbf{x}_k^j(s), \mathbf{u}_k^j(s)) \quad p = 0, 1, \dots, N. \quad (18)$$

But if vehicle  $j$  uses the assumed input  $\hat{\mathbf{u}}_k^j(\cdot)$ , then the trajectory will be as follows:

$$\hat{\mathbf{x}}_k^j(p) = \mathbf{x}^j(k) + \sum_{s=0}^{p-1} f(\hat{\mathbf{x}}_k^j(s), \hat{\mathbf{u}}_k^j(s)) \quad p = 0, 1, \dots, N. \quad (19)$$

Subtracting (18) from (19) yields

$$\begin{aligned} \hat{\mathbf{x}}_k^j(p) - \mathbf{x}_k^j(p) &= \sum_{s=0}^{p-1} \left[ f(\mathbf{x}_k^j(s), \mathbf{u}_k^j(s)) - f(\hat{\mathbf{x}}_k^j(s), \hat{\mathbf{u}}_k^j(s)) \right] \\ & \quad p = 0, 1, \dots, N. \end{aligned} \quad (20)$$

By finding  $\alpha_k^{j,i}$  the tube  $\mathbb{H}_k^{j,i}$  is determined.  $\alpha_k^{j,i}$  is the upper bound on  $|\hat{\mathbf{x}}_k^j(p) - \mathbf{x}_k^j(p)|$  and then is calculated by solving the following maximization problem.

*Problem 3.* Given  $\mathbf{u}_k^j(\cdot)$ ,  $\mathbf{x}_k^j(\cdot)$  and  $\beta^j$  calculate for all  $p = 0, 1, 2, \dots, N$

$$\alpha_k^{j,i}(p) = \text{Max}_{\{\hat{\mathbf{x}}_k^j(\cdot), \hat{\mathbf{u}}_k^j(\cdot)\}} \left[ \sum_{s=0}^{p-1} \left[ f(\mathbf{x}_k^j(s), \mathbf{u}_k^j(s)) - f(\hat{\mathbf{x}}_k^j(s), \hat{\mathbf{u}}_k^j(s)) \right] \right]$$

subject to

$$\begin{aligned} \hat{\mathbf{x}}_k^j(p+1) &= f(\hat{\mathbf{x}}_k^j(p), \hat{\mathbf{u}}_k^j(p)), \\ \hat{\mathbf{x}}_k^j(0) &= \mathbf{x}^j(0) \quad \hat{\mathbf{x}}_k^j(p) \in \mathbb{X} \quad \hat{\mathbf{u}}_k^j(p) \in \mathbb{U} \\ |\hat{\mathbf{u}}_k^j(p) - \mathbf{u}_k^j(p)| &\leq \beta^j. \end{aligned} \quad (21)$$

In optimization Problem 3 a nominal trajectory  $\mathbf{x}_k^j(\cdot)$  is given as input. In other words the optimal value depends on any nominal trajectory. On the other hand, at any time step DMPC generates a new trajectory which is served as nominal trajectory in Problem 3. Hence, Problem 3 should be modified to be independent of any nominal trajectory and be applicable for a general trajectory  $\mathbf{x}_k^j(\cdot)$ . Hence, considering  $\hat{\mathbf{x}}_k^j(\cdot)$  and  $\hat{\mathbf{u}}_k^j(\cdot)$  as new decision variables in the optimization Problem 3 and shifting the time from  $k$  to  $k-d$  will lead to Problem 2. This way, the tube can be computed offline and used for online applications, and no online computation is required.

**3.3.2. Tube Calculation Algorithm.** The application of Theorem 1 allows calculating the tube  $\mathbb{H}_k^{j,i}$  around the trajectory of each neighboring vehicle. Assume that the control input at each sample time for vehicle  $j$  is bounded as follows:

$$|\mathbf{u}_k^j(p) - \mathbf{u}_{k-1}^j(p+1)| \leq \mu, \quad p = 0, 1, \dots, N-2, \quad (22)$$

where  $\mu$  is a vector with appropriate dimension and is called the *maneuverability* vector. It allows restricting the maneuverability of each vehicle during the time segments when the actual predicted trajectory of vehicle may not be available to followers. The neighboring vehicles by considering such restrictions can compute a bound on the trajectories. Then, if at any time step  $k$  vehicle  $i$  receives the information from leader  $j$  with  $d$  steps time delay, that is,  $\mathbf{x}_{k-d}^j(\cdot)$ , then the results of Theorem 1 are applicable by finding  $\beta^j(\mu, d)$ . To find  $\beta^j(\mu, d)$  the input constraint (22) can be used sequentially as follows (the superscript  $j$  is dropped temporarily):

$$\begin{aligned} -\mu &\leq \mathbf{u}_k(p) - \mathbf{u}_{k-1}(p+1) \leq \mu, \\ -\mu &\leq \mathbf{u}_{k-1}(p+1) - \mathbf{u}_{k-2}(p+2) \leq \mu, \\ &\vdots \\ -\mu &\leq \mathbf{u}_{k-d+1}(p+d-1) - \mathbf{u}_{k-d}(p+d) \leq \mu, \\ &\vdots \\ -d \cdot \mu &\leq \mathbf{u}_k(p) - \mathbf{u}_{k-d}(p+d) \leq d \cdot \mu, \\ p &= 0, 1, 2, \dots, N-d. \end{aligned} \quad (23)$$

Hence,

$$\beta^j = d \cdot \mu, \quad p = 0, 1, 2, 3, \dots, N-d. \quad (24)$$

Therefore, we have

$$\beta^j(p, \mu, d) = \begin{cases} 0, & p = 0, \\ \mu, & p = 1, \\ 2 \mu, & p = 2, \\ \vdots & \vdots \\ (d-1)\mu, & p = d-1, \\ d \mu, & p = d, d+1, \dots, N-d. \end{cases} \quad (25)$$

Then  $\alpha_k^{j,i}$ , the radius of tube  $\mathbb{H}_k^{j,i}$ , is calculated from Problem 2 after calculating  $\beta^j$  from (25). The output of Problem 2 is the trajectory of vector  $\alpha_k^{j,i}(\cdot)$  and will be used in the online DMPC algorithm.

The main benefit of using this procedure for tube calculation is that calculation of  $\alpha_k^{j,i}$  does not impose any online computation load as it can be computed offline. The only parameter which may be unknown prior to mission is the time delay  $d$ . Hence, the tube can be computed for different delays, and the results are given to the DMPC controller as some tabulated data, to be used in online applications. In this way no online computation is imposed.

**3.3.3. Nonconvexity Avoidance.** Using a tube instead of a trajectory in the cost function (12) can lead to nonconvexity for the optimization problem due to the nonconvex nature of the tube. Thus, in order to avoid the nonconvexity, in the cost function (12),  $\mathbf{x}_{k-d}^j(p)$  is not modified (or replaced by tube  $\mathbb{H}_k^{j,i}$ ), instead the desired relative position  $\mathbf{r}^{i,j}(p)$  will be modified as follows:

$$\mathbf{r}^{i,j}(p) \leftarrow \mathbf{r}^{i,j}(p) + \text{sign}(\mathbf{r}^{i,j}(p)) \cdot \alpha_k^{j,i}(p), \quad (26)$$

where  $\alpha_k^{j,i}$  is the tube radius at time step  $k$  and is calculated from Problem 2. In fact, the margin  $\alpha_k^{j,i}(p) \geq 0$  is added to the desired distance to ensure the safety. Since  $\mathbf{r}^{i,j}(\cdot)$  is the relative position vector,  $\alpha_k^{j,i}$  is multiplied by the diagonal matrix  $\text{sign}(\mathbf{r}^{i,j}(\cdot))$  to ensure that adding  $\alpha_k^{j,i}$  increases the magnitude of  $\mathbf{r}^{i,j}(\cdot)$ ; in fact  $\mathbf{r}^{i,j}(\cdot)$  is preserved, and extra margin is added.

**3.4. Delay-Compensated DMPC Problem.** The delay-compensated DMPC (DC-DMPC) problem  $\mathcal{P}_{\text{DC}}^i(k)$  is defined below at time step  $k$  for any  $i$ th vehicle. The outputs of this decentralized optimization problem are (1) the input and (2) the state trajectory of the local vehicle over the prediction horizon and (3) the trajectory of neighboring vehicles during the tail of the cost function.

*Problem 4.* DC-DMPC Problem  $\mathcal{P}_{\text{DC}}^i(k)$ . Given  $\Gamma_D^i(t_k)$  calculate

$$J_{\text{DC}}^i(\Gamma_D^i(k)) = \underset{\{\mathbf{x}_k^i(\cdot), \mathbf{u}_k^i(\cdot), \mathbf{x}_k^{-i}(N-d:N)\}}{\text{Min}} J_{\text{DC}}^i \times \left( \Gamma_D^i(k), \mathbf{u}_k^i(\cdot), \mathbf{x}_k^{-i}(N-d:N), \mathbf{u}_k^{-i}(N-d:N) \right) \quad (27)$$

subject to

(i) for  $p = 0, 1, 2, \dots, N-1$ : (9a)–(9c)

(ii) and for  $p = N-d, \dots, N$  and  $(i, j) \in \mathbb{E}$ :

$$\mathbf{x}_k^{j,i}(p+1) = f(\mathbf{x}_k^{j,i}(p), \mathbf{u}_k^{j,i}(p)); \quad (28a)$$

$$\mathbf{x}_k^{j,i}(N-d) = \mathbf{x}_{k-d}^j(N) \quad (28b)$$

$$\mathbf{x}_k^{j,i}(p) \in \mathbb{X}^j, \quad \mathbf{u}_k^{j,i}(p) \in \mathbb{U}^j \quad (28b)$$

$$\mathbf{x}_k^{j,i}(N) \in \mathbb{X}_f^j, \quad (28c)$$

(iii) and for  $p = 0, 1, 2, \dots, N-2$ :

$$\left| \mathbf{u}_k^i(p) - \mathbf{u}_{k-1}^i(p+1) \right| \leq \boldsymbol{\mu}, \quad (28d)$$

where constraints (9a)–(9c) are the same as in the delay-free DMPC problem  $\mathcal{P}^i(t_k)$  and correspond to the trajectory for calculating the delayed cost function  $J_D^i$ . Constraints (28a) and (28b) correspond to the tail cost function  $J_{\text{Tail}}^i$  (13). Constraint (28c) is the terminal constraint and is the same as (9c) in  $\mathcal{P}^i(t_k)$ . Moreover, constraint (28d) is added for safety guarantee purpose, where  $|\cdot|$  represents the component-wise absolute value of vector and  $\boldsymbol{\mu}$  is a vector with appropriate dimension and called the *maneuverability* vector. Constraint (28d) is called *maneuverability* constraints (such constraint is also known as *compatibility* constraints in similar literatures, [24, 25]). It allows restricting the maneuverability of each vehicle and hence calculating the tube as explained in Theorem 1. Roughly speaking, constraints (28a)–(28c) are added for cooperation performance improvement, and constraint (28d) is added for safety purpose.

*Remark 2.* Expressing equations (28d) in the norm form instead of component-wise absolute value leads to more conservative results in practice.

**3.5. DC-DMPC Algorithm.** Algorithm 2 is presented for the online implementation of the proposed DC-DMPC problem  $\mathcal{P}_{\text{DC}}^i(k)$ . The algorithm is formulated for the  $i$ th vehicle; in fact, all vehicles run this algorithm during the mission simultaneously.

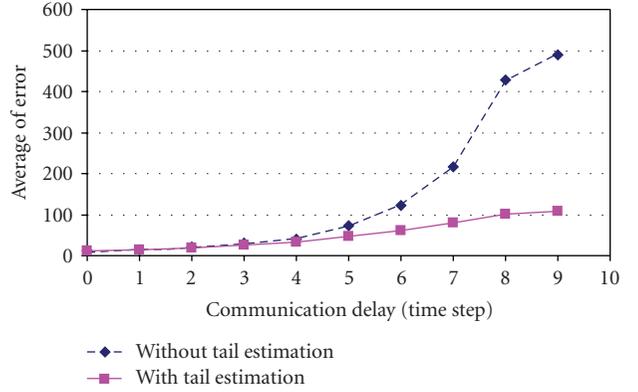


FIGURE 5: Average of error versus communication delay for a triangle formation of three vehicles.

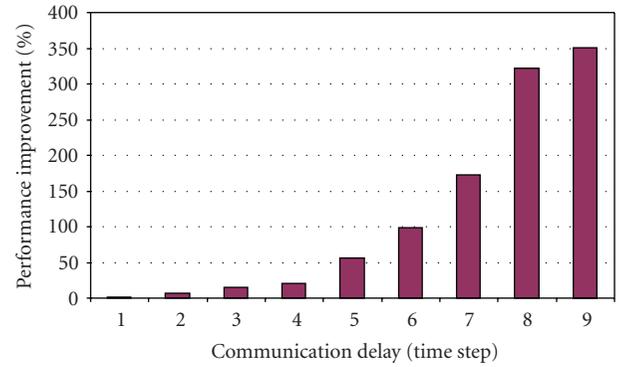


FIGURE 6: Percentage of average of error versus communication delay for a triangle formation of three vehicles.

*Initialization.* For  $k = 0$  to  $k = d-1$  at steps 2, 3, and 4 assume that  $N_k^i = 0$ . For  $k = 0$  to  $k = d-2$  at step 7 assume  $N_k^i = 0$  (because no information from leader(s) is available).

This algorithm is a modified version of Algorithm 1 and handles the large communication delays; it also provides the safety guarantee by executing the step 3 using the tube DMPC approach.

*Remark 3.* In step 2 of Algorithm 2, to measure the time delays, it is assumed that all the vehicles are equipped with synchronous clocks. Then at each sampling time, each vehicle, in addition to its predicted trajectory, broadcasts the time it is calculated the trajectory. Hence, the knowledge on the amount of delay is not required *a priori*.

## 4. Simulation Results

In this section, the proposed approach is tested on the formation problem of a fleet of unmanned vehicles with the following 2DOF dynamics. This dynamics can capture the motion of a mini-hovercraft equipped with two sets of thrusters to control the side and forward motions; also, a

Given  $\mu$ ,  $\mathbb{E}$  and initial condition of vehicle  $i$ , do:

- (1)  $k \leftarrow 0$ .
- (2) Measure  $\mathbf{x}^i(k)$ , calculate delay from leader and update the delayed *information vector* of (10).
- (3) **Calculate the tube of all leaders:**
  - For  $\forall j \in \mathbb{V}$  where  $(i, j) \in \mathbb{E}$ 
    - (i) Calculate  $\beta^j(\cdot, \mu, d)$  from (25).
    - (ii) Calculate  $\alpha_k^{j,i}(\cdot)$  from the data provided by offline solution of Problem 2.
    - (iii) Update  $\mathbf{r}^{i,j}(\cdot) \leftarrow \mathbf{r}^{i,j}(\cdot) + \text{sign}(\mathbf{r}^{i,j}(\cdot)) \cdot \alpha_k^{j,i}(\cdot)$  and update  $\mathbf{r}^{i,j}$  in the cost function (12).
- END
- (4) Solve  $\mathcal{P}_{DC}^i(\mathbf{k})$  and calculate the control action  $\mathbf{u}_k^i(\cdot)$  and  $\mathbf{x}_k^i(\cdot)$ .
- (5) Send the state trajectory  $\mathbf{x}_k^i(\cdot)$  to the followers where  $(j, i) \in \mathbb{E}$ .
- (6) Apply the control action  $\mathbf{u}_k^i(0)$  to individual vehicle  $i$ .
- (7) Receive  $\mathbf{x}_{k-d}^j(\cdot)$  from leaders where  $(i, j) \in \mathbb{E}$ .
- (8)  $k \leftarrow k + 1$ . Goto step 2.

ALGORITHM 2: DC-DMPC.

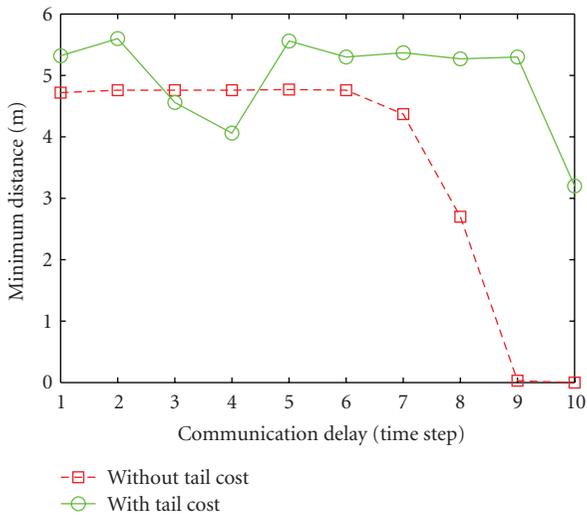


FIGURE 7: Minimum distance between vehicles in the presence of a communication delays.

small damping term is added as the typical resistance of water to hovercraft motion is small:

$$\begin{aligned}
 \dot{x}_1 &= x_2, \\
 \dot{x}_2 &= -0.1x_2 + u_1, \\
 \dot{x}_3 &= x_4, \\
 \dot{x}_4 &= -0.1x_4 + u_2,
 \end{aligned} \tag{29}$$

where  $x_1$  and  $x_2$  represent the components of position vector in  $x - y$  coordinate and  $x_3$  and  $x_4$  are their corresponding velocity components. The input vector is given by  $\mathbf{u} = [u_1, u_2]$ . This dynamics is discretized by sampling time  $h = 0.1$  sec. The MPC problem is solved using optimization toolbox of MATLAB. When Algorithm 1 is used a set of  $2N$  decision variables are used for each vehicle to model the control inputs (because each vehicle has only two control inputs). Once Algorithm 2 is used, two extra sets of decision

variables are used;  $2N + 2d$  decision variables are used. It means the computation grows with communication delay.

In the first simulation example, it is desired to examine the effect of the tail cost added to the cost function. The simulation was run for two cases.

- (1) Using the cost function *without the tail cost*. In this case the control input is set to  $u = 0$  for the tail of the cost function. The extra decision variables for tail cost estimation are not included in the optimization problem.
- (2) Using the cost function *with the tail cost*. In this case the tail of the cost function (12) is estimated using the extra decision variables in the optimization problem  $\mathcal{P}_{DC}^i(k)$ .

The matrix penalties in the cost function are chosen as follows:  $\mathbf{Q} = \mathbf{I}$  (where  $\mathbf{I}$  is identity matrix), and  $\mathbf{R} = 0.1 \mathbf{0I}$  then

$$\mathbf{P} = \begin{bmatrix} 0.8634 & 1.447 & 0.000 & 0.000 \\ 1.4471 & 2.800 & 0.000 & 0.000 \\ 0.0000 & 0.000 & 0.863 & 1.447 \\ 0.0000 & 0.000 & 1.447 & 2.800 \end{bmatrix} \tag{30}$$

and  $\mathbf{S} = 20\mathbf{I}$ . The final penalty matrix  $\mathbf{P}$  is calculated from the Lyapunov equation [15]. The optimization horizon is chosen to be  $N = 10$ . In all cases no disturbances, sensor noise, or model uncertainty is considered in the simulations in order to focus on the effect of the communication delay.

A triangular leader-follower formation of three vehicles is first considered, where the moving vehicles are controlled to form an isosceles triangle with edges of 10, 7.02, and 7.02 m. The actual cooperation cost is chosen as the cooperation performance index of any follower  $i \in \mathbb{V}$  as follows:

$$I^i(k) = \sum_{j|(i,j) \in \mathbb{E}} \left\| \mathbf{x}^i(k) - \mathbf{x}^j(k) - \mathbf{r}^{i,j}(k) \right\|_{\mathbf{S}}^2 \quad \forall i \in \mathbb{V}. \tag{31}$$

This performance index represents the deviation (error) from the desired formation shape; hence, a smaller  $I^i$  is

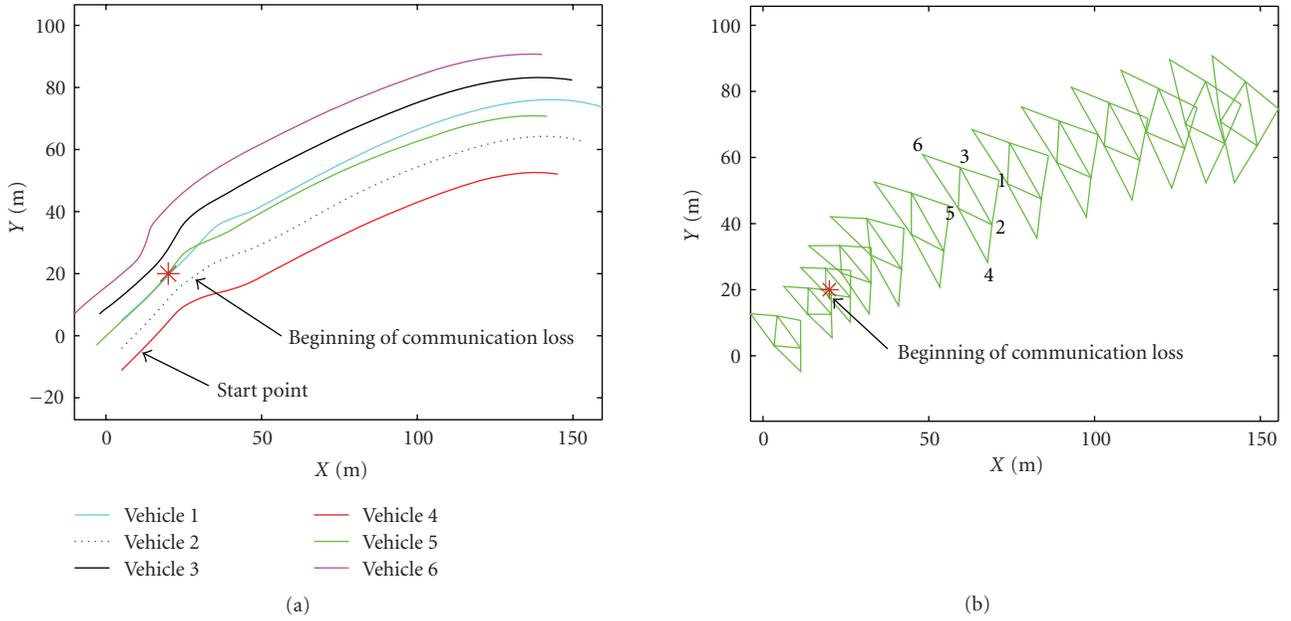


FIGURE 8: Trajectory (a) and formation snapshot (b) of six vehicles in triangular formations when Algorithm 2 is used.

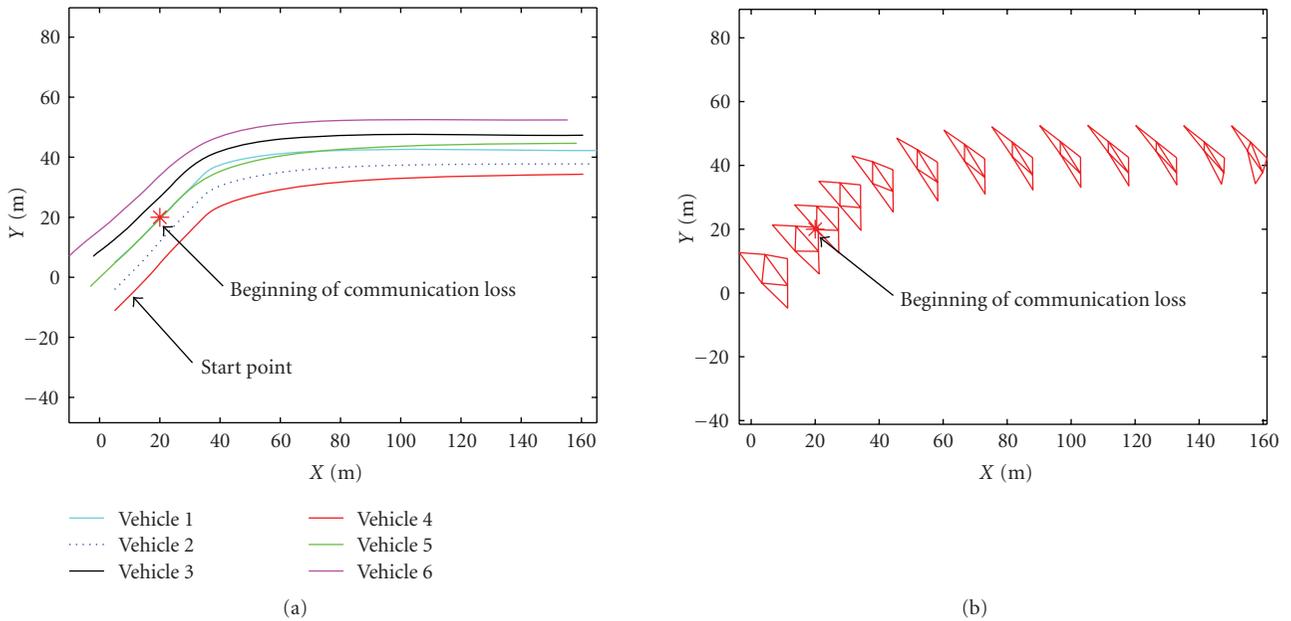


FIGURE 9: Trajectory (a) and formation snapshot (b) of six vehicles in triangular formations when Algorithm 1 is used.

desired. Then, the total error of the group at time step  $k$  is presented as follows:

$$I(k) = \sum_{i \in \mathbb{V}} I^i(k). \quad (32)$$

**4.1. Performance Improvement.** The simulation was repeated for cases with different communication delays, and the results are gathered in Figures 5 and 6. Figure 5 shows the average of the formation error (32) with each point representing a single simulation. It can be seen from

Figure 5 that estimating the tail of the cost function leads to smaller errors. Figure 6 shows the percentage of performance improvement (error reduction) when the tail estimation method is used comparing with the case where no tail estimation is performed. It shows also that for large communication delays the proposed approach can reduce the error by 350%. It can also enhance the stability of the formation; for this particular example, it is seen that if the communication delay is increased to around  $d = 10$  time steps, the formation becomes unstable when using the cost function without the

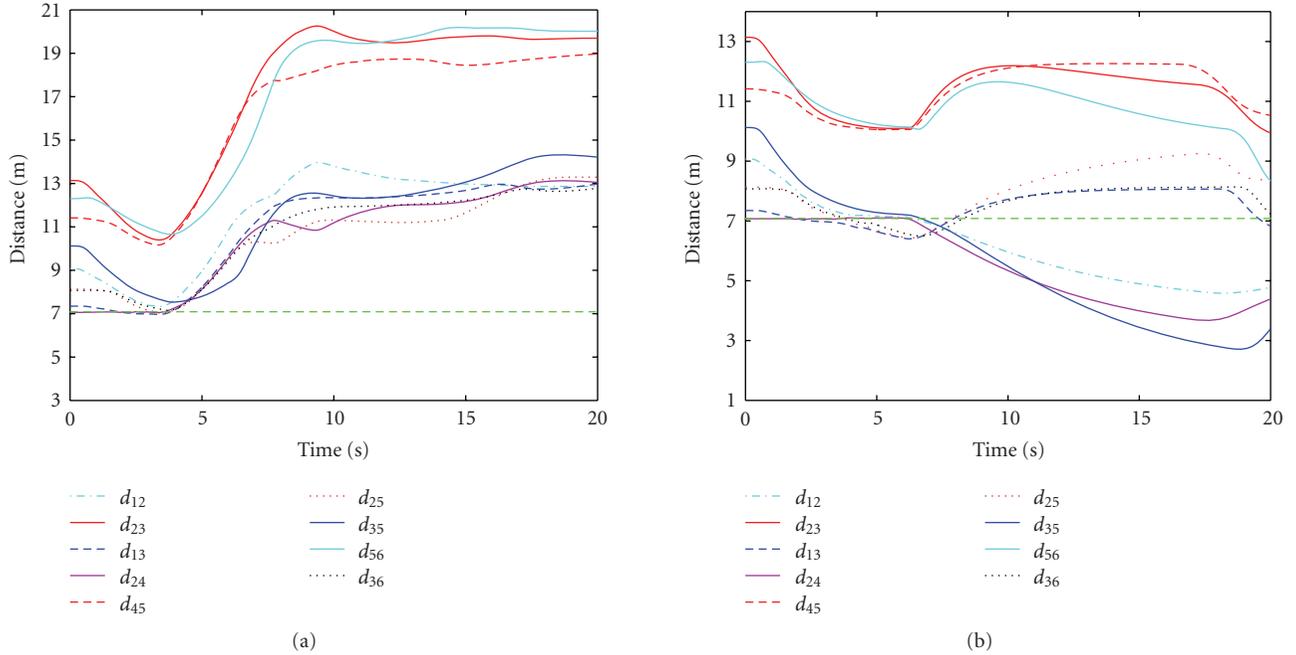


FIGURE 10: Distances between each pair of vehicles for Algorithm 2 (a) and Algorithm 1 (b).

tail cost. However, it is still stable with the proposed cost function including the tail cost. This result is consistent with that of [26, 27] where a final cost is added to the cost function for formation stability, although they did not consider communication delays. The overall trend of the graphs in Figure 5 shows that the error goes up with delay.

Note that, since we assume no uncertainty in the model, the main source of error arises from communication delay and decentralized nature of the problem. The communication delay forces the decentralized controllers to use delayed information instead of updated information, which adds some degree of uncertainty into the problem, leading to error and degraded cooperation performance. Also, the decentralized nature of the problem restricts the decentralized controllers to access all the information about neighboring vehicles intention and control calculation procedure. For example, vehicles do not take into account the interaction graph topology of neighboring vehicles, in order to reduce the computation load. The second source of error does not exist in a centralized framework.

**4.2. Safety Using Tube DMPC.** It is observed that in some simulations in presence of large communication delays although adding the final cost can lead to better performance and stable formation, the vehicles may get too close to each other and collide. For example, the minimum distance among the vehicles for a set of simulations is depicted in Figure 7. The desired distance between each pair of neighboring vehicles is 7.07 m. As seen even for the case where tail cost estimation is used the vehicles may get too close to each other.

Therefore, in the next simulation the proposed Algorithm 2 which includes the tube DMPC is used.

This case involves the triangular formation control of six vehicles. The communication graph topology is set as follows:

$$\begin{aligned} \mathbb{V} &= \{1, 2, 3, 4, 5, 6\}, \\ \mathbb{E} &= \{(2, 1), (3, 1), (4, 2), (5, 4), (6, 3)\}. \end{aligned} \quad (33)$$

The results are shown in Figure 8 through Figure 10. In this case, two sets of way points are considered to be visited by the fleet. At the beginning the intervehicle communication delay is  $d = 0$  but after 3 sec (around point (22, 22)) communication loss begins which leads to  $d = 7$  time step intervehicle communication delay. As seen from Figure 8 the vehicles start to keep a larger distance, and the formation is expanded for safety upon communication loss. Figure 9 shows the same scenario when Algorithm 1 is used.

The distances between each pair of neighboring vehicles are shown in Figure 10 for two cases: (1) Algorithm 1 and (2) Algorithm 2. It is desired that vehicles keep a 7.07 m distance from neighbors. As seen from Figure 10(b) in the case of Algorithm 1, vehicles get too close to each other and may collide. However, Algorithm 2 offers a loose (Figure 8) but safe formation (Figure 10(a)) as the consequence of using tube DMPC for safety.

For this case the minimum desired separation between two neighboring vehicle is 7.07 m. The corresponding required minimum relative distance, when Algorithm 2 is used for a delay of  $d = 7$  time-step, is about 10 m as measured from Figure 10(a). Hence, for this case the added margin due to using tube DRHC is calculated as  $10 - 7.07 \approx 3$  m; then for such relatively large delay, the added margin is about 42.8%, which does not seem conservative. In the presence of model uncertainty and communication noise this margin will grow.

## 5. Conclusions

A new delay-compensated decentralized model predictive controller (DC-DMPC) is proposed that can address communication delays arising from any source including packet losses. The proposed approach provides two key features. The first feature is that the tail of the cost function is estimated for performance improvement in the presence of the large intervehicle communication delays. Simulation results show that this method can lead to 350% improvement in the cooperation performance. The added online computation load depends on the communication delay and should be taken into account for any particular application. The second feature is the development of the tube-based MPC to provide guaranteed formation safety against possible collisions in the presence of large intervehicle communication delays. Together, these results provide a new approach to deal with the large intervehicle communication delays in DMPC problems that ensures safe formations and improved cooperation performance.

## Acknowledgment

This paper was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) through a Strategic Project Grant (STPGP 350889-07) leading by the third author and a Discovery Project Grant by the second author.

## References

- [1] H. A. Azadi, B. W. Gordon, and Y. M. Zhang, "Decentralized receding horizon control for cooperative multiple vehicles subject to communication delay," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 6, pp. 1959–1965, 2009.
- [2] E. Franco, T. Parisini, and M. M. Polycarpou, "Cooperative control of discrete-time agents with delayed information exchange: a receding-horizon approach," in *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC '04)*, vol. 4, pp. 4274–4279, Atlantis, Paradise Island, Bahamas, December 2004.
- [3] E. Franco, T. Parisini, and M. M. Polycarpou, "Stable receding-horizon cooperative control of a class of distributed agents," in *Proceedings of the American Control Conference (ACC '05)*, vol. 7, pp. 4673–4678, IEEE Press, June 2005.
- [4] E. Franco, T. Parisini, and M. M. Polycarpou, "Cooperative control of distributed agents with nonlinear dynamics and delayed information exchange: a stabilizing receding-horizon approach," in *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control (CDC-ECC '05)*, pp. 2206–2211, IEEE Press, December 2005.
- [5] E. Franco, L. Magni, T. Parisini, M. M. Polycarpou, and D. M. Raimondo, "Cooperative constrained control of distributed agents with nonlinear dynamics and delayed information exchange: a stabilizing receding-horizon approach," *IEEE Transactions on Automatic Control*, vol. 53, no. 1, pp. 324–338, 2008.
- [6] A. Richards and J. P. How, "Implementation of robust decentralized model predictive control," in *AIAA Guidance, Navigation, and Control Conference*, pp. 4929–4941, AIAA, August 2005.
- [7] A. N. Venkat, J. B. Rawlings, and S. J. Wright, "Stability and optimality of distributed model predictive control," in *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control (CDC-ECC '05)*, pp. 6680–6685, IEEE Press, December 2005.
- [8] I. Alvarado, D. Limon, T. Alamo, and E. F. Camacho, "Output feedback robust tube based MPC for tracking of piecewise constant references," in *Proceedings of the 46th IEEE Conference on Decision and Control (CDC '07)*, pp. 2175–2180, IEEE Press, December 2007.
- [9] P. Trodden and A. Richards, "Robust distributed model predictive control using tubes," in *Proceedings of the American Control Conference*, pp. 2034–2039, Minneapolis, Minn, USA, June 2006.
- [10] X. Zhang, R. Xu, C. Kwan, L. Haynes, Y. Yang, and M. M. Polycarpou, "Fault tolerant formation flight control of UAVs," *International Journal of Vehicle Autonomous Systems*, vol. 2, no. 3-4, pp. 217–235, 2004.
- [11] M. Innocenti, L. Pollini, and F. Giulietti, "Management of communication failures in formation flight," *Journal of Aerospace Computing, Information and Communication*, pp. 19–35, 2004.
- [12] T. Keviczky, F. Borrelli, and G. J. Balas, "Decentralized receding horizon control for large scale dynamically decoupled systems," *Automatica*, vol. 42, no. 12, pp. 2105–2115, 2006.
- [13] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.
- [14] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
- [15] H. Chen and F. Allgöwer, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," *Automatica*, vol. 34, no. 10, pp. 1205–1217, 1998.
- [16] A. Richards and J. P. How, "Robust distributed model predictive control," *International Journal of Control*, vol. 80, no. 9, pp. 1517–1531, 2007.
- [17] D. Garagić, "On delay-dependent stability of a swarm of networked autonomous vehicles under communication constraints," in *Proceedings of the IEEE Swarm Intelligence Symposium (SIS '05)*, pp. 297–301, IEEE Press, June 2005.
- [18] T. Keviczky, B. Vanek, F. Borrelli, and G. J. Balas, "Hybrid decentralized receding horizon control of vehicle formations," in *Proceedings of the American Control Conference*, pp. 3358–3363, Minneapolis, Minn, USA, June 2006.
- [19] A. Richards and J. How, "Decentralized model predictive control of cooperating UAVs," in *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC '04)*, vol. 4, pp. 4286–4291, Atlantis, Paradise Island, Bahamas, December 2004.
- [20] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang, "Review of road traffic control strategies," *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2043–2065, 2003.
- [21] L. Pallottino, V. G. Scordio, and A. Bicchi, "Decentralized cooperative conflict resolution among multiple autonomous mobile agents," in *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC '04)*, vol. 5, pp. 4758–4763, Atlantis, Paradise Island, Bahamas, December 2004.
- [22] Y. Yoon, H. Kim, J. Shin, T. Choe, and Y. Park, "Communication in distributed model predictive collision avoidance," in *Proceedings of the 1st International Conference on Robot Communication and Coordination*, Athens, Greece, 2007.

- [23] H. K. Khalil, *Nonlinear Systems*, Prentice Hall, Englewood Cliffs, NJ, USA, 2nd edition, 1996.
- [24] W. B. Dunbar and R. M. Murray, "Receding horizon control of multi-vehicle formations: a distributed implementation," in *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC '04)*, pp. 1995–2002, Atlantis, Paradise Island, Bahamas, December 2004.
- [25] W. B. Dunbar and R. M. Murray, "Distributed receding horizon control for multi-vehicle formation stabilization," *Automatica*, vol. 42, no. 4, pp. 549–558, 2006.
- [26] W. B. Dunbar and R. M. Murray, "Model predictive control of coordinated multi-vehicle formations," in *Proceedings of the 41st IEEE Conference on Decision and Control*, vol. 4, pp. 4631–4636, IEEE Press, December 2002.
- [27] D. Gu and E. Yang, "A suboptimal model predictive formation control," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1295–1300, IEEE Press, 2005.

## Review Article

# Vision-Based Tracking of Uncooperative Targets

Suresh K. Kannan,<sup>1</sup> Eric N. Johnson,<sup>1</sup> Yoko Watanabe,<sup>2</sup> and Ramachandra Sattigeri<sup>3</sup>

<sup>1</sup> School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

<sup>2</sup> Department of Systems Control and Flight Dynamics, ONERA, 31055 Toulouse, France

<sup>3</sup> Guided Systems Technologies, Stockbridge, GA 30281, USA

Correspondence should be addressed to Suresh K. Kannan, kannan@gatech.edu

Received 2 July 2010; Revised 6 November 2010; Accepted 28 February 2011

Academic Editor: Yu Gu

Copyright © 2011 Suresh K. Kannan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a summary of a subset of the extensive vision-based tracking methods developed at Georgia Tech. The problem of a follower aircraft tracking an uncooperative leader, using vision information only, is addressed. In all the results presented, a single monocular camera is used as the sole source of information used to maintain formation with the leader. A Kalman filter formulation is provided for the case where image processing may be used to estimate leader motion in the image plane. An additional piece of information, the subtended angle, also available from computer vision algorithm is used to improve range estimation accuracy. In situations where subtended angle information is not available, an optimal trajectory is generated that improves range estimation accuracy. Finally, assumptions on the target acceleration are relaxed by augmenting a Kalman Filter with an adaptive element.

## 1. Introduction

The most commonly used set of sensors required for accurate state estimation of a flight vehicle is a combination of an Inertial Measurement Unit (IMU) and a Global Positioning System (GPS). In most cases, heading information may be estimated when the vehicle is in motion. A magnetometer may be used if an accurate measurement is required even if the vehicle is not in motion, a helicopter for example. The IMU and Magnetometer rely on measuring quantities that are always available. The GPS signal however, is an external signal that is human generated. By its very nature of being a low-power signal, it is susceptible to jamming. Indeed, GPS simulators are available that are able to emulate GPS signals within the confines of a research lab. In the case of flight vehicles flying through terrain such as an urban city or enclosed areas, where GPS signals degrade, the translational variable estimates such as position and velocity also degrade in accuracy. Thus, there is a desire to develop reliable and affordable navigation alternatives that do not require an external signal.

Birds, insects, or, for that matter, most organisms with the ability to see the visual spectrum are able to navigate

(in flight or on the ground), quite reliably using vision information. Highly evolved organisms, especially predators, use a pair of eyes to allow stereo vision ability in order to estimate range to targets. In contrast to the IMU, most animals use relatively low-quality mechanical sensors equivalents, but use vision extensively to determine relevant state information. If a human pilot can visually acquire a target, he or she can usually fly to it; hence an automated system should be able to replicate the same behavior. From as early as the year 2000, the UAV Research Facility (UAVRF) at Georgia Tech has been interested in producing navigation solutions using a single vision camera. In this paper, we consider the specific problem of navigating (following) a target object (leader) using a single vision camera. One of the earliest problems addressed was the estimation, control, and guidance of a glider [1, 2] towards a target window opening, after launch, in order to deliver a simple payload. The only sensor available on the glider was a single camera. Subsequently, under a Multi-University Research Initiative (MURI) program, the Active-Vision Control Systems project, initiated in 2005, a large body of theoretical work, simulation, and flight testing.

Vision-based estimation and navigation methods depend greatly on the application at hand. Some early work on landing an unmanned vehicle using vision have been reported in [3], where multiple views of a landing target from a single camera are used to provide motion estimates in order to land an unmanned helicopter on a moving platform. A more recent result [4] uses natural landmarks and a feature tracking algorithm to provide SLAM solutions for high-altitude UAVs. Both these methods use homography-based techniques to compute vehicle relative translations and rotations. An impressive result is provided in recent works by Chiba University and Universite de Technologie de Compiègne and is reported in [5]. Here, optic flow and inertial measurements are used to compute the position and velocity of a vehicle carrying a single monocular camera. An interesting and useful aspect of [5] is the combined analysis of image processing and controller-plant stability results. In many of these works and related references therein, the approach used is strongly influenced by the task at hand. However, if one were to separate the different phases of the estimation problem, it is the image processing data reduction tasks that caters most directly to the environment the vehicle operates in. Once raw relative motion data is available, an estimator is generally used to fuse the vision-based data with other sensors to maintain an estimate of the vehicle's state.

The focus of this paper is to present tracking and estimation methods to effectively track an uncooperative target using monocular vision only and fly in formation with it. Even though the leader may have its own navigation solution, this information is not used (for practical flight testing purposes, however, in order to setup the initial formation, leader GPS information is indeed used. It is then slowly phased out with vision information being phased in for leader relative position estimation purposes) by the follower aircraft. The goal is to use monocular vision in order to estimate and track the relative position to the target. In this paper, the terms leader and target are used interchangeably. The leader-follower problem appears in many ways. For example, navigating relative to a known landmark (fixed target) or avoiding an obstacle, both, involve estimating the target position and maintaining a safe offset from it. This paper will address only the tracking and estimation part of the problem. The particular image processing algorithms used in our application are described in [6].

The overall approach is to use a vision camera to produce a 2-D image plane representation of the scene. This setup is shown in Figure 1. The 2-D image produced is discrete, has a finite resolution determined by the number of pixels, and can only see a subset of the scene because of a limited field-of-view. This 2-D image is the only source of information on the target, that may be *measured*. If the target undergoes any motion *relative* to the follower, those changes are readily visible in the follower's 2-D image of the scene, as long as the target is in the field-of-view. Image processing techniques may be used to track the target's movement on the image plane. Changes in range from camera to target will not be easily observable due to the difficulty in determining depth perception with monocular vision. The bioinspired solution would be to use stereo vision to improve range estimates.

Using just a single camera, the range estimate will have to be improved using other techniques. It is known that the target in question has certain dimensions. Image processing techniques may be used to compute to angle that the 2-D image of the target subtends on the image plane (essentially, size of the blob) and may be used to improve range estimates. If the target is far away, or small, the finite resolution of the 2-D images may prevent this approach. In such cases, the camera's trajectory relative to the target may be modified. Unlike stereo vision where two perspectives of the scene at the same instant is available, monocular vision can only rely on image sequences and camera motion to improve range estimates. Additionally, sophisticated image processing may be used to leverage any shape/size information that may be available in the 2-D image.

The paper is structured as follows, Section 2 provides a problem statement that illustrates the equations governing the relative dynamics. Image processing techniques are based on [6] and extended to use the Fast Level Set Marching Method [7] for computational speed. Section 3 provides the Extended Kalman Filter formulation for the navigation problem where image processing is able to provide subtended angle information. Simulation and Flight Test results for this case are provided. Next, the scenario where subtended angle information is not available due to lack of image processing or when the target is very far away is considered in Section 4. Here, the problem of minimizing the variance on the range estimation errors is treated as an optimization problem, and the resulting guidance trajectory is compared to the nominal guidance policy. Simulation results are provided. Until Section 4, the Kalman Filter formulation assumes a model for target maneuvering. For example, in Section 3.3 flight test results are described where the target is moving in a circle. This is used as a priori knowledge and used to model the target acceleration characteristics. In order to mitigate this problem of having to assume a target model, Section 5 introduces a Kalman Filter augmented with an Adaptive Element that is able to deal with unmodeled/unknown target accelerations. Simulation results for this case are also provided. In general, the following sections outline the core theory associated with each method and provides simulation and/or flight test results. In each section, the reader is referred to publications where further details may be found. The Extended Kalman Filter (EKF) formulation used for Section 3 is described in the Appendix.

## 2. Problem Formulation

Let  $\mathbf{X}_v$  and  $\mathbf{V}_v$  be a vehicle's position and velocity vectors expressed in an inertial frame (denoted by  $F_I$ ). Suppose that the vehicle dynamics can be modeled as the following simple linear system:

$$\dot{\mathbf{X}}_v(t) = \mathbf{V}_v(t), \quad \dot{\mathbf{V}}_v(t) = \mathbf{a}_v(t), \quad (1)$$

where  $\mathbf{a}_v$  is the vehicle's acceleration input. It is assumed that all the vehicle states are available through its own-ship

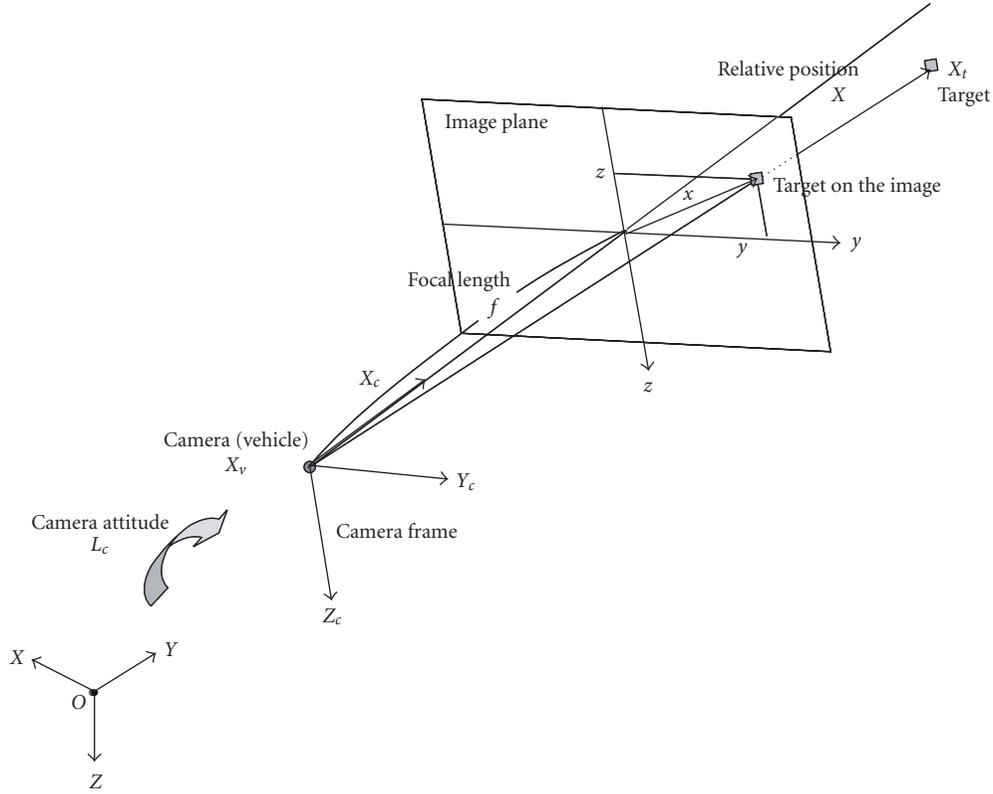


FIGURE 1: Pinhole camera model.

navigation system. The target dynamics are similarly given by

$$\dot{\mathbf{X}}_t(t) = \mathbf{V}_t(t), \quad \dot{\mathbf{V}}_t(t) = \mathbf{a}_t(t), \quad (2)$$

where  $\mathbf{X}_t$ ,  $\mathbf{V}_t$ , and  $\mathbf{a}_t$  are the target's position, velocity and acceleration vectors, respectively. Relative position, velocity and acceleration of the target with respect to the vehicle are defined by

$$\begin{aligned} \mathbf{X}(t) &= \mathbf{X}_t(t) - \mathbf{X}_v(t), \\ \mathbf{V}(t) &= \mathbf{V}_t(t) - \mathbf{V}_v(t), \\ \mathbf{a}(t) &= \mathbf{a}_t(t) - \mathbf{a}_v(t). \end{aligned} \quad (3)$$

Then the relative motion dynamics are formulated as follows:

$$\dot{\mathbf{X}}(t) = \mathbf{V}(t), \quad \dot{\mathbf{V}}(t) = \mathbf{a}(t). \quad (4)$$

Since the target's maneuver is unknown in most cases, we need to apply some model for  $\mathbf{a}_t$  based on its prior knowledge. For example, the following target model is called the Singer model [8]:

$$\dot{\hat{\mathbf{a}}}_t(t) = -\alpha_s \hat{\mathbf{a}}_t(t) + \mathbf{w}(t), \quad \alpha_s > 0, \quad (5)$$

where  $\mathbf{w}(t)$  represents zero mean Gaussian noise.

In this problem, for simplicity, it is assumed that a 2-D passive vision sensor is mounted at the center of gravity of the vehicle (this assumption can be easily removed by including

extra rotational dynamics caused by an offset in the relative motion model.) We also assume that an image processor which is able to detect a target position in an image frame is available. Let  $L_c$  denote a known camera attitude represented by a rotation matrix from the inertial frame  $F_I$  to a camera frame which is denoted by  $F_C$ . A camera frame is taken so that the camera's optical axis aligned with its  $X_c$  axis. Then the relative position expressed in  $F_C$  will be

$$\mathbf{X}_c(t) = L_c(t)\mathbf{X}(t) = \begin{bmatrix} X_c(t) & Y_c(t) & Z_c(t) \end{bmatrix}^T. \quad (6)$$

Assuming a pinhole camera model shown in Figure 1, the target position in the image  $\mathbf{x}(t)$  is given by

$$\mathbf{x}(t) = \begin{bmatrix} y(t) \\ z(t) \end{bmatrix} = \frac{f}{X_c(t)} \begin{bmatrix} Y_c(t) \\ Z_c(t) \end{bmatrix}, \quad (7)$$

where  $f$  is the focal length of the camera. In this paper,  $f = 1$  is used without loss of generality. More target information will be available if the image processor can detect the target's shape, size, contours, or other characteristics in addition to its center position  $\mathbf{x}$ . The vision-based navigation objective is to estimate the unknown target states from the image processor outputs and the known vehicle and camera states.

### 3. Vision-Based Formation Flight

In this section, an Extended Kalman Filter (EKF) is designed to estimate the relative state of the leader aircraft with respect

to the follower using vision information. More details on the EKF are provided in the Appendix. The EKF is implemented and tested in an image-in-the-loop 6 DoF multiple UAV flight simulation and in actual flights of UAVs. An own-ship navigation filter and a flight guidance and control system have already been developed and implemented [9]. The image processor used is developed for realtime target tracking by Ha et al. [10, 11]. In flight experiments, closed-loop vision-based formation flights of two UAVs have been successfully achieved [12]. Estimation results obtained in both simulations and flight tests are illustrated in this section.

### 3.1. Navigation Filter Design

**3.1.1. Process Model.** In the EKF design for this application, an estimation state vector is taken as follows:

$$\mathbf{x} = \left[ \mathbf{u}^T \quad \dot{\mathbf{u}}^T \quad \frac{1}{r} \quad \frac{\dot{r}}{r} \quad b \right]^T, \quad (8)$$

where  $\mathbf{u}$  is an unit vector pointing from the follower to the leader and  $r$  is a range between the two aircraft. They are defined by

$$\mathbf{u} = \frac{\mathbf{X}}{\|\mathbf{X}\|}, \quad r = \|\mathbf{X}\|, \quad (9)$$

where  $\mathbf{X}$  is the relative position vector. In bearing-only tracking problems, it is very common to use an inverse of range  $1/r$  instead of the range itself to reduce the nonlinearity associated with its dynamics [13]. Moreover, the unit vector is chosen as an estimation state, instead of bearing and elevation angles which are commonly used [13], to avoid a singularity.  $b$  in (8) is a constant wingspan of the leader airplane. When using the random walk model for the target's velocity, the process model is written as follows:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{u}} \\ -\left(\|\dot{\mathbf{u}}\|^2 - \frac{1}{r} \mathbf{u} \cdot \mathbf{a}_v\right) \mathbf{u} - 2\frac{\dot{r}}{r} \dot{\mathbf{u}} - \frac{1}{r} \mathbf{a}_v \\ -\frac{1}{r} \frac{\dot{r}}{r} \\ \|\dot{\mathbf{u}}\|^2 - \frac{1}{r} \mathbf{u} \cdot \mathbf{a}_v - \left(\frac{\dot{r}}{r}\right)^2 \\ 0 \end{bmatrix} + \mathbf{w} \quad (10)$$

$$= \mathbf{f}(\mathbf{x}, \mathbf{a}_v) + \mathbf{w},$$

where  $\mathbf{a}_v$  is the follower's acceleration input and  $\mathbf{w}$  is the process noise.

**3.1.2. Measurement Model.** It is well-known that the range information is not always observable when only a 2-D image position of a target center is measured from a single camera [14]. To guarantee range observability, the line-of-sight from the camera to the target needs to be in motion [15]. However, in the formation flight application, the objective is to maintain a constant offset from the leader, leading to poor range estimates. To overcome this observability issue, an angle that is subtended by the leader's size (e.g., wingspan)

is introduced as an additional measurement. The subtended angle  $\alpha$  is defined by

$$\alpha = 2 \tan^{-1} \frac{b}{2r}. \quad (11)$$

In defining this subtended angle, there are some assumptions made regarding the target's orientation with respect to the camera. The most common configuration for pursuit is with the camera directly behind the target. This results in the wing span being the most convenient parameter to represent the size of the target. Hence, the difference between the camera and target heading angles is assumed to be small. If for example the heading of the target is offset 90 degrees from the camera heading, it is the fuselage length that will be visible to any image processing. Additionally, as long as the difference between the camera and target pitch angles is small, there will be no ambiguity between pitch and roll orientation of the target. The image processor developed in [10] tracks the target's contour in images, and can thus extract the target size. Figure 2 shows an example of the image tracking result. The image processor processes raw images captured by the onboard camera, and then outputs, in image coordinates, the center and the two wing-tips of the leader airplane. Let  $\mathbf{x}_c$  be an image coordinate of the center position, and let  $\mathbf{x}_l$  and  $\mathbf{x}_r$  be those of the left and right wing-tip positions, respectively. Since a pinhole camera model is assumed, the subtended angle measurement can be calculated by using  $\mathbf{x}_l$  and  $\mathbf{x}_r$  as follows:

$$\alpha = 2 \tan^{-1} \frac{\|\mathbf{x}_l - \mathbf{x}_r\|}{2}. \quad (12)$$

The measurement vector in the EKF is chosen as

$$\mathbf{z} = \left[ \mathbf{u}_c^T \quad \alpha \right]^T, \quad (13)$$

where  $\mathbf{u}_c$  is the unit vector expressed in the camera frame  $F_C$ . The unit vector is chosen as a measurement instead of the two angles for the same reason as choosing it as an estimation state. The measurement vector  $\mathbf{z}$  can be expressed as a nonlinear function of the estimation state  $\mathbf{x}$ . The measurement model is given by

$$\mathbf{z}_k = \begin{bmatrix} L_{c_k} \mathbf{u}_k \\ 2 \tan^{-1} \frac{b_k}{2} \left(\frac{1}{r}\right)_k \end{bmatrix} + \boldsymbol{\nu}_k = \mathbf{h}(\mathbf{x}_k) + \boldsymbol{\nu}_k, \quad (14)$$

where  $L_{c_k}$  is the known camera attitude at the time step  $t_k$  and  $\boldsymbol{\nu}_k$  is a measurement error. The EKF prediction and update procedures discussed in the Appendix are applied to the process and measurement models given in (10) and (14).

**3.1.3. State Constraint Enforcement.** Since the unit vector  $\mathbf{u}$  defined in (9) always satisfies  $\|\mathbf{u}\| = 1$ , the estimates of  $\mathbf{u}$  and its changing rate  $\dot{\mathbf{u}}$  in the state vector used in the EKF design have the following constraints:

$$\|\mathbf{u}\| = 1, \quad \mathbf{u}^T \dot{\mathbf{u}} = 0. \quad (15)$$



FIGURE 2: Image tracking result: (a) original image taken by the onboard camera, (b) center and two wing-tip positions of the leader airplane are detected.

These constraints are naturally satisfied in the EKF prediction procedure (A.3), but not in the EKF update procedure (A.9). Therefore, the constraints (15) need to be enforced after the EKF update at each time step as follows:

$$\begin{aligned} \hat{\mathbf{u}}_k &= \frac{\hat{\mathbf{u}}_k}{\|\hat{\mathbf{u}}_k\|}, \\ \hat{\mathbf{u}}_k &= \hat{\mathbf{u}}_k - \left( \hat{\mathbf{u}}_k^T \hat{\mathbf{u}}_k \right) \hat{\mathbf{u}}_k. \end{aligned} \quad (16)$$

Calise [16] proposed another method to enforce the state constraints in the EKF design. In his work, the Kalman gain was computed by minimizing the augmented Lagrangian that includes the estimation error and the state constraint. This results in simply adding the correction term to the original EKF update laws (see (A.8), (A.9)). However, this approach is not used in this work to avoid using a singular measurement error covariance matrix.

**3.1.4. Leader's State Estimation.** Once the estimation state is updated by using a measurement at  $t_k$  in the EKF, the leader's estimated position and velocity can be calculated from the updated estimate  $\hat{\mathbf{x}}_k$  and from the known follower's state as shown below

$$\begin{aligned} \hat{\mathbf{X}}_{t_k} &= \frac{\hat{\mathbf{u}}_k}{\left( \hat{1}/r \right)_k} + \mathbf{X}_{v_k}, \\ \hat{\mathbf{V}}_{t_k} &= \frac{\hat{\mathbf{u}}_k}{\left( \hat{1}/r \right)_k} + \frac{\hat{\mathbf{u}}_k}{\left( \hat{1}/r \right)_k} \cdot \left( \frac{\hat{r}}{r} \right)_k + \mathbf{V}_{v_k}. \end{aligned} \quad (17)$$

Since a nonaccelerating target is assumed in the EKF design, the leader's acceleration is estimated by  $\hat{\mathbf{a}}_{t_k} = \mathbf{0}$ . These estimated leader states are fed back to the guidance system to maintain the formation.

**3.2. Simulation Results.** Figure 3 shows a 6-DOF multi-airplane flight simulation in a formation flight configuration Figure 3(a) and the airplane model used in this simulation Figure 3(b). The airplane is the MURI research airplane with

a wingspan of 15.7 (ft) and overall length of 11 (ft). Flight control is achieved using an adaptive neural network-based controller that generates actuator signals based on the navigation system outputs and commanded position, velocity, and attitude [17]. The follower airplane has a camera, whose position, orientation and field of view generates simulated images. The synthetic images are processed and provide the locations of the leader's center and the two wing-tips. The synthetic image and the image processor outputs are shown in the left bottom window of the simulation interface in Figure 3.

In simulation, the leader airplane flies straight with a constant speed of 100 (ft/sec). The follower aircraft is commanded to maintain a given position *relative* to the leader. The formation is maintained by using the estimation result. Figure 4 compares the estimated (in blue) and the true (in green) relative positions and velocities. For example, at the start of the time scale (20 s) in Figure 4, the follower is commanded to keep the leader 100 ft ahead ( $X = 100$  ft), stay directly behind ( $Y = 0$  ft), and fly at the same altitude ( $Z = 0$  ft) as the leader.

The vision-based estimation results match the corresponding true states well. There is a bias in the range ( $X$  position) estimation, that is, difference between the two solid lines. Even though the range estimation accuracy is improved after the vehicle's lateral motion due to the relative position command changes, a small bias remains. This is due to a measurement bias in the leader's size. However, the estimation is sufficiently accurate to achieve closed-loop vision-based formation flight. Formation breakup happens when the image processor can no longer track the target. Depending on the image processor, this happens when the target leaves or partially leaves the visual field of the camera. This may occur if  $X$  estimates are inaccurate enough where the leader overshoots the target or lags so far behind the target that the image processor is no longer able to segment the target contours. The higher likelihood of formation breakup is in the orthogonal directions, where the  $Y$  and  $Z$  relative position estimates have enough errors where the target leaves the camera's field of view. Subsequent to validation of the



FIGURE 3: (a) 6 DoF Multi-airplane Flight Simulation Interface, (b) MURI research airplane.

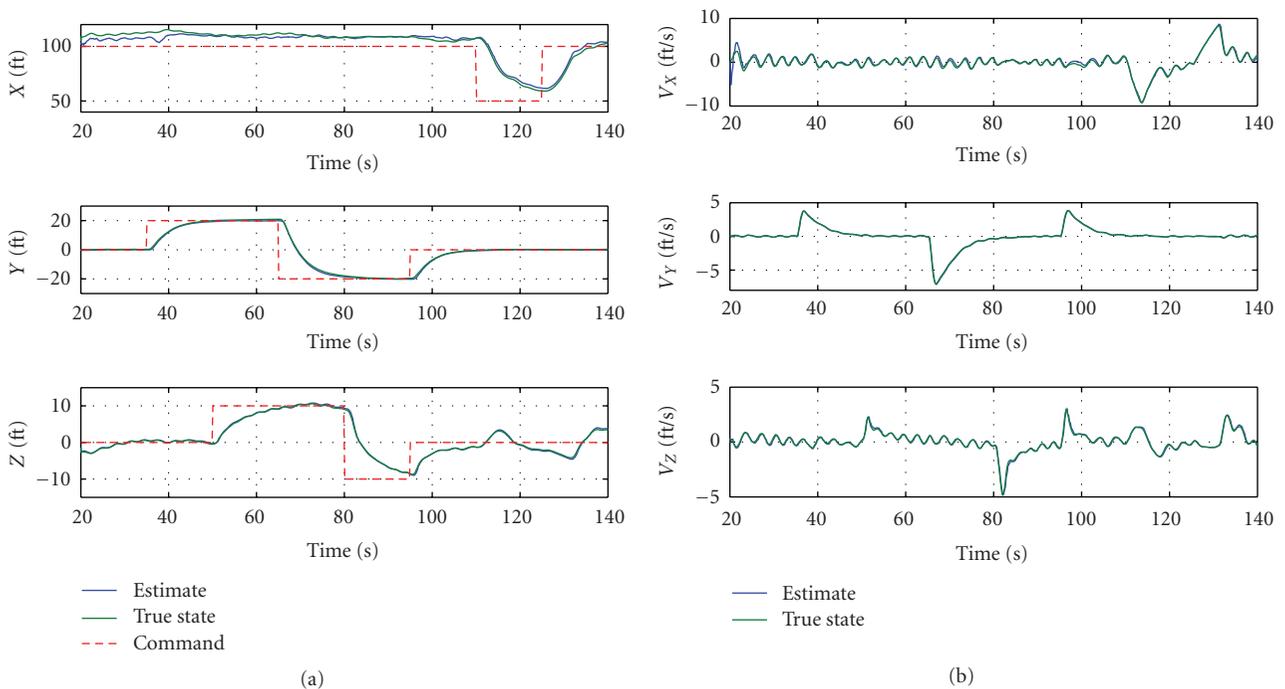


FIGURE 4: Estimated versus true relative position and velocity.

EKF-based estimation performance in simulation, flight tests were conducted and is discussed in the next section.

### 3.3. Flight Test Results

**3.3.1. Platform Aircraft.** The simulation-tested algorithms from the previous sections were implemented on two different types of UAVs, a rotary-wing aircraft, the GTMax, and a fixed-wing aircraft, the GTYak. Initially, the GTMax helicopter (based on Yamaha's RMax helicopter) was used as a follower aircraft. In later tests, the fixed-wing aircraft, the GTYak (scale model of the Yak-54), was deployed as a follower. For the vision-based guidance and control tests, the UAVs were equipped with cameras fixed to their bodies. The real-time image processor was implemented and executed within the flight computers of the GTMax and the GTYak.

The GTEdge fixed-wing aircraft was deployed as the leader for all tests. The GTEdge is a 33% scale model of the aerobatic Edge 540T airplane and is highly maneuverable with vertical hover capabilities [18]. All three aircraft of both types (fixed-wing and rotary-wing) use the same adaptive trajectory tracking controller described in Johnson and Kannan [17] for flight control purposes. Table 1 summarizes the physical specifications of the three UAVs used in the flight tests.

**3.3.2. EKF Modification for Circular Target Maneuvers.** During flight tests, the GTEdge was commanded to fly in a circle with a constant ground speed  $V_t = 65$  (ft/s) and a constant angular rate  $\omega_t = 0.1$  (rad/s). Unlike the straight level flight case used in the simulations, a circling motion was chosen because of a limited flight test field size. Since the leader's motion does not satisfy the zero acceleration assumption,

TABLE 1: Some specifications of UAV's used for formation flights.

Name	GTMax (follower) Tamaha RMax	GTyak (follower) Yak-54	GTEdge (leader) Edge540T
Model			
Wingspan (ft)	—	8.50	8.75
Rotor diameter (ft)	10.2	—	—
Overall length (ft)	11.9	7.90	7.80
Empty weight (lbs.)	157	40	35
Engine (cc)	246	100	100

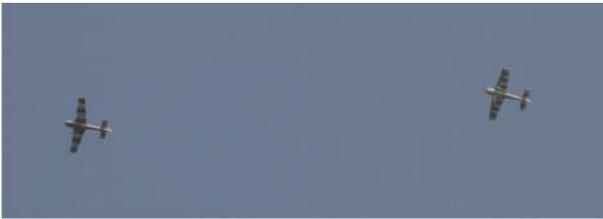


FIGURE 5: GTYak and GTEdge in formation.

the EKF design discussed in Section 3.1 was slightly modified. Suppose that it is known that the leader is flying in a circle at a constant altitude. In the modified EKF design, the leader's lateral acceleration  $a_{lat}$  is added as an extra estimation state of the EKF. Since  $a_{lat}$  should be constant in a circling motion, the dynamics can be modeled by

$$\dot{a}_{lat} = 0 + w_{a_{lat}}, \quad (18)$$

where  $w_{a_{lat}}$  is zero mean Gaussian noise. By assuming that the leader's lateral acceleration is perpendicular to the velocity vector in the horizontal  $X$ - $Y$  plane, the leader's acceleration is estimated by

$$\mathbf{a}_t = \frac{a_{lat}}{\sqrt{V_{X_t}^2 + V_{Y_t}^2}} \begin{bmatrix} -V_{Y_t} \\ V_{X_t} \\ 0 \end{bmatrix}, \quad (19)$$

where the leader's velocity vector  $\mathbf{V}_t = [V_{X_t} \ V_{Y_t} \ V_{Z_t}]^T$  is given in (17). In the process model (10),  $\mathbf{a}_v$  should be replaced by  $\mathbf{a}_v - \mathbf{a}_t$ .

**3.3.3. Formation Flight Results.** The first sustained closed-loop vision-based formation flight between the GTMax and the GTEdge was achieved in June, 2006. This may have been the first time automated formation flight based on vision only has ever been accomplished. Results pertaining to this particular flight test between a rotary-wing and fixed-wing aircraft are available in [12, 19].

Following these tests, it was concluded that a follower vehicle which is comparable in performance to the leader would allow flights at higher speeds and accommodate more maneuvering. Hence the GTYak was chosen to act as the leader in future experiments. The closed-loop vision-based formation flight between two fixed-wing airplanes, the GTYak and the GTEdge, was achieved in July, 2007. In this flight test, the GTEdge flew in a circle with  $V_t = 75$  (ft/sec) and  $\omega_t = 0.1$  (rad/sec). Since the GTYak has more maneuverability than the GTMax, a higher speed was used. The GTYak was commanded to hold a relative position of  $\mathbf{X}_{com} = [80 \ 0 \ -20]^T$  (ft) from the leader. Figure 5 shows the GTYak and the GTEdge flying in formation. The image processor keeps track of every time it loses lock on the target, even briefly. This is shown in the Figure 6(a) along with image processor outputs. Although the image processor sometimes failed to capture the leader airplane, even though it was in view, it was able to recover and lock onto the leader after several frames of no lock. The right plot of Figure 6 shows estimated and true relative positions. Figure 7 compares the vision-based (solid-blue) and GPS-INS-based estimates (dashed-green) of velocity and acceleration. The vision-based estimates agree well with the GPS-INS solutions and is sufficient to maintain formation.

#### 4. Optimal Trajectory for Range Observability

There exist cases where in (12) it may not be possible to measure  $\alpha$ , the subtended angle, reliably. The monocular cameras used for vision have finite resolution. At large values of the range  $r$ , of the follower from the target, the image of the target on the image plane will not have enough pixel width to reliably compute subtended angle information. At these large ranges,  $\mathbf{x}_l$  and  $\mathbf{x}_r$  are close to each other and will result in a large variance in  $\alpha$  measurements. Another situation where  $\alpha$  may not be available is when sufficient processing power is not available to compute  $\alpha$ . In either case, the inability to measure the subtended angle results in poor estimates of the range  $r$ . This then becomes a "Bearings-Only" [13] tracking problem where we want to estimate depth from image sequences [20]. Keeping in mind

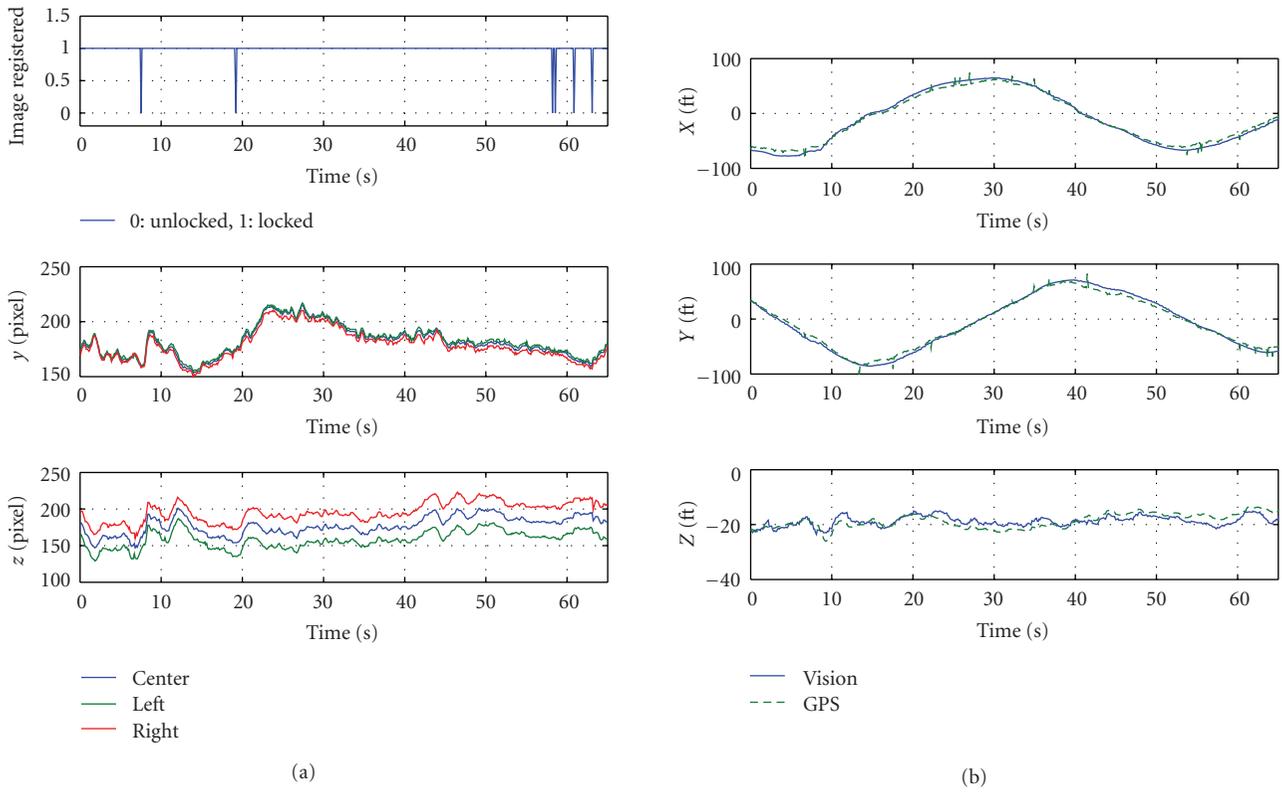


FIGURE 6: (GTyak-GTEdge) (a) image processor outputs, (b) vision-based versus GPS-based estimated relative position.

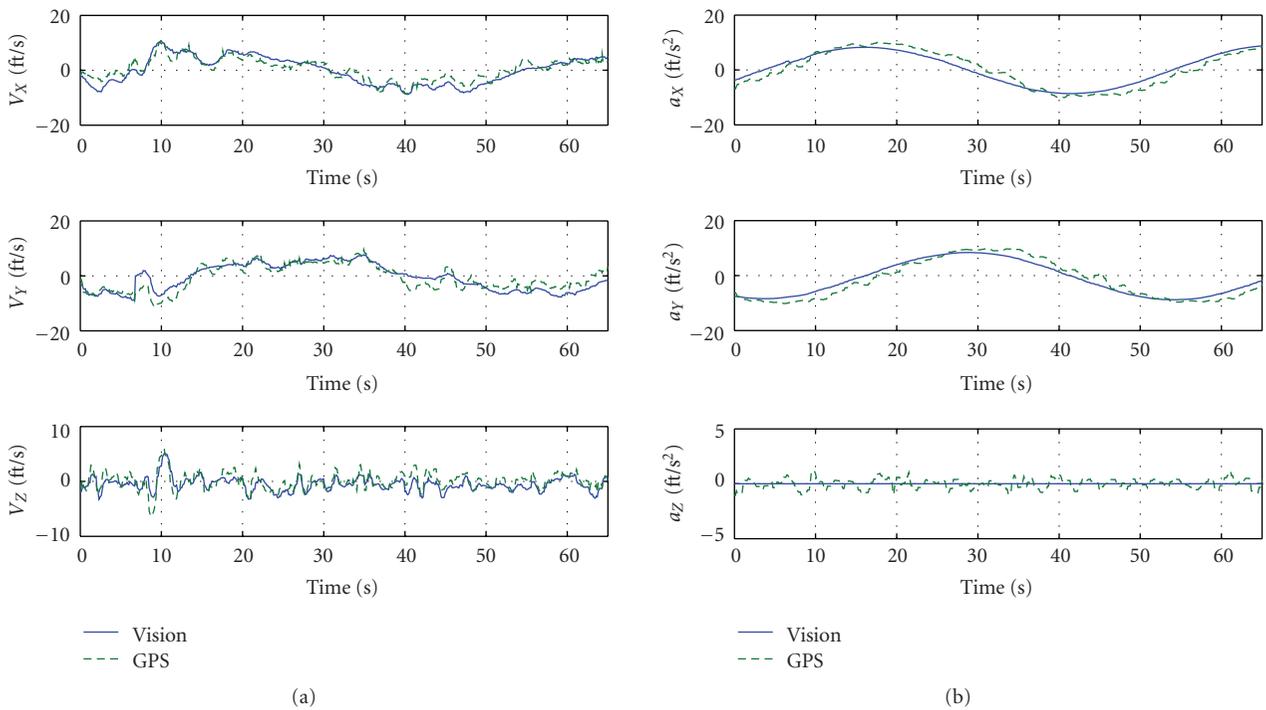


FIGURE 7: (GTyak-GTEdge) Vision-based versus GPS-based estimated relative velocity (a), and acceleration (b).

that we have used the unit vector  $\mathbf{u}$  to parameterize any bearing information to avoid singularities, we are interested in improving the accuracy of the range estimation.

A rigorous evaluation of the observability of depth information may be computed using the extended output Jacobian-based observability condition put forth in the works of Mariottini [21, 22]. Some of the motivation for using a nonlinear observability condition arises from the geometry of the problem being nonlinear where a linearization of the equations is nonobservable [23]. However, intuitively,  $\hat{\beta}$ , the bearing to target, has to be nonzero in order for range to be observable. We can come to the same conclusion by examining  $P_k$ , the error covariance matrix of our Kalman Filter implementation described in the Appendix. The objective is to maximize  $P_k^{-1}$ , which may be achieved by maximizing the term corresponding to the range estimation error in the term  $H_k^T R_k^{-1} H_k$ . This particular term, if expanded, is given by

$$\left(\frac{\hat{u}_{y_k}^-}{\sigma_{\xi_y}}\right)^2 + \left(\frac{\hat{u}_{z_k}^-}{\sigma_{\xi_z}}\right)^2. \quad (20)$$

We also desire to limit the magnitude of control in  $y$  and  $z$  directions, that is, minimize  $a_y, a_z$ . The optimization problem may then be stated as

$$\min_{a_y, a_z} \frac{1}{2} \int_0^{t_f} \left\{ -\left(\frac{\hat{u}_{y_k}^-}{\sigma_{\xi_y}}\right)^2 - \left(\frac{\hat{u}_{z_k}^-}{\sigma_{\xi_z}}\right)^2 + K_y^2 a_y^2 + K_z^2 a_z^2 \right\} dt \quad (21)$$

subject to the camera dynamics given by (1), since it is attached to the vehicle and here we assume it is not rotating with the follower's body frame. Here,  $t_f$  represents the terminal time when the vehicle reaches the destination.

In order to derive an analytical solution to the optimization, the following assumptions are made.

- (i) The camera motion is limited to the  $X$ - $Y$  plane, and the  $Z$  position is held constant. This allows  $\hat{u}_{z_k}^-$  to be assumed constant and  $a_z = 0$ . Thus the second and fourth terms of the performance index in (21) may be eliminated.
- (ii) To further simplify the problem,  $\dot{u}_x$  may be assumed constant and  $a_x = 0$ .

This reduces the problem to having just two states  $u_y$  and  $\dot{u}_y$ . The boundary conditions are given by the initial and terminal states

$$\begin{aligned} u_y(0) &= u_{y0}, \\ \dot{u}_y(0) &= \dot{u}_{y0}, \\ u_y(t_f) &= u_{yf}, \\ \dot{u}_y(t_f) &= \dot{u}_{yf}. \end{aligned} \quad (22)$$

The analytical solution for this problem may be obtained by solving the Euler-Lagrange equations [24]. With the Hamiltonian defined as

$$H = -\frac{1}{2} \frac{u_y^2}{\sigma_{\xi_y}^2} + \frac{1}{2} K_y^2 a_y^2 - \lambda_1 \dot{u}_y + \lambda_2 a_y, \quad (23)$$

the resulting Euler-Lagrange equations may be expressed as

$$\begin{aligned} \dot{\lambda}_1 &= -\frac{\partial H}{\partial u_y} = \frac{u_y}{\sigma_{\xi_y}^2}, \\ \dot{\lambda}_2 &= -\frac{\partial H}{\partial \dot{u}_y} = \lambda_1, \end{aligned} \quad (24)$$

$$\frac{\partial H}{\partial a_y} = K_y^2 a_y + \lambda_2 = 0,$$

where  $\lambda_1$  and  $\lambda_2$  are Lagrange multipliers. A differential equation for the follower's lateral acceleration  $a_y$  can be derived and is given by

$$\frac{d^4 a_y}{dt^4} = \frac{1}{(\sigma_{\xi_y}^2 K_y)^2} a_y. \quad (25)$$

The solutions for  $u_y(t)$ ,  $\dot{u}_y$ , and  $a_y(t)$  is given by

$$\begin{aligned} u_y(t) &= -\frac{\sigma_{\xi_y}}{K_y} (A \sin \omega t + B \cos \omega t - C e^{\omega t} - D e^{-\omega t}), \\ \dot{u}_y(t) &= \frac{\sqrt{\sigma_{\xi_y} K_y}}{K_y} (A \cos \omega t - B \sin \omega t - C e^{\omega t} + D e^{-\omega t}), \\ a_y(t) &= -\frac{1}{K_y^2} (A \sin \omega t + B \cos \omega t + C e^{\omega t} + D e^{-\omega t}), \end{aligned} \quad (26)$$

where the constants  $A, B, C, D$  may be found by satisfying the boundary conditions given by (22), and the parameter  $\omega$  is given by

$$\omega = \frac{1}{\sqrt{K_y \sigma_{\xi_y}}}. \quad (27)$$

Small values of  $K_y$  give larger lateral maneuvers, resulting in better estimates for the range information. In practical applications, too large a lateral maneuver can quickly result in the target moving out of the view of the camera. If this is a situation where the target is so far away that the subtended angle cannot be used, small values of  $K_y$  are acceptable because larger lateral deviation will not necessarily move the object out of the camera's field of view. However, if this is a situation where the target is close, and the image processor is such that subtended angle is not available,  $K_y$  will have to be increased as the range decreases. In general, we want to keep  $K_y$  as high as possible resulting in less maneuvering in the follower. Thus, the variance of the range estimation error may be used to schedule  $K_y$  by decreasing  $K_y$  as the variance increases.

Figure 8 shows the estimation and guidance results with and without the optimal guidance policy. In both cases, a step command is issued in the  $x$ -axis and  $y$ -axis, from an initial commanded relative position of [100, 10, 0] (ft) to [50, 0, 0] (ft). In Figure 8(b) the guidance policy is only turned on from  $t = 20$  s to  $t = 60$  s. The  $y$ -axis relative position plot in Figure 8(b) shows the meandering lateral

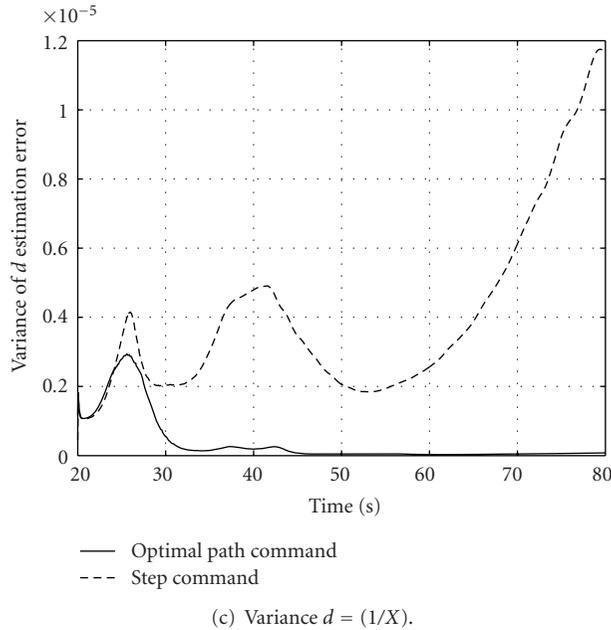
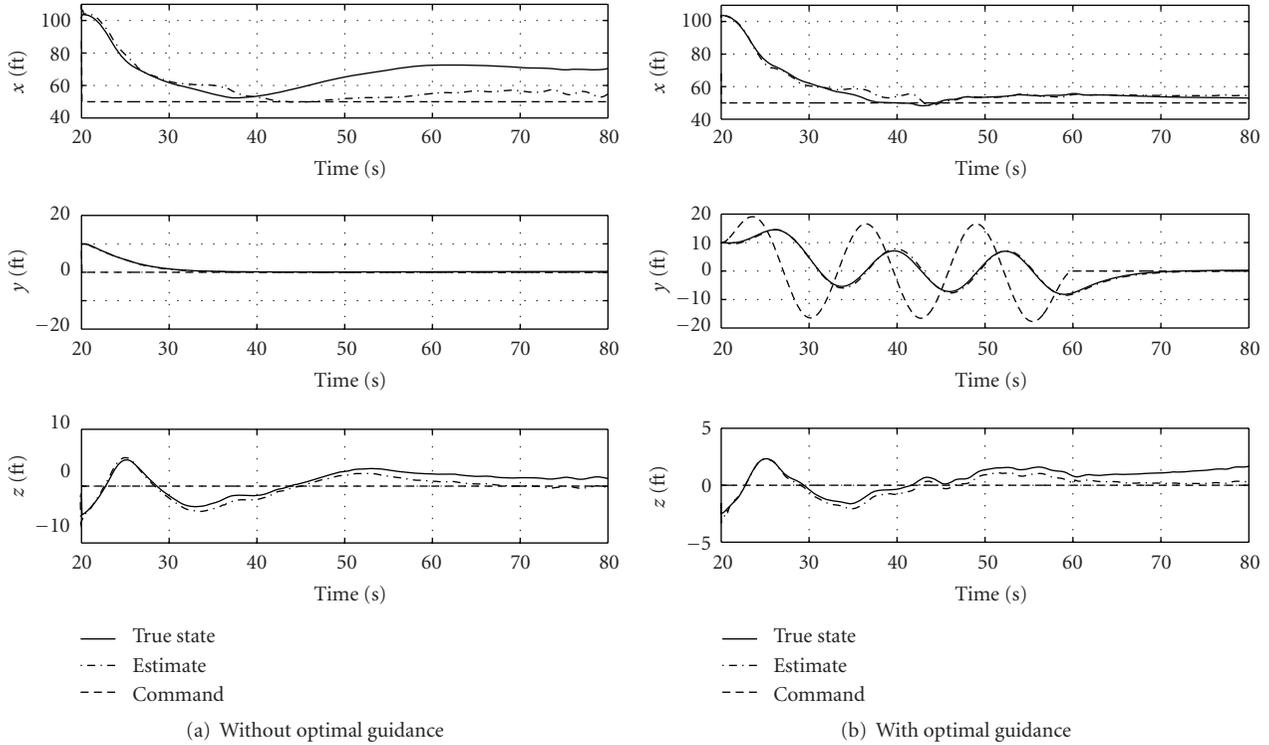


FIGURE 8: Optimal trajectory for range estimation.

path generated due to the optimal policy. In Figure 8(a), the lack of an optimal policy results in a large 20 ft steady state error between the true state of the aircraft and the estimated state of the aircraft along the  $x$ -axis. This estimation error results in a large error between the command and the true state. In contrast, Figure 8(b), has smaller estimation errors and hence smaller command tracking errors. Finally, Figure 8(c) shows a plot of the variance in estimation of

$d = 1/X$  and conclusively illustrates the advantages of the optimal policy in reducing estimation error.

### 5. Adaptive State Estimation

One of the major challenges for target tracking arises from target motion uncertainty. This uncertainty refers to the fact that an accurate dynamic model of the target being tracked

is generally not available to the tracker. In addition, any measurements of the target being tracked are corrupted by noise and time delays. A Kalman filter or one of its several variants is usually used as the target state estimator, but its performance may be seriously degraded unless the estimation error due to unknown target maneuvers and other uncertainties is accounted for. Some of the most popular approaches to handling the unknown target maneuvers in the target state estimator design are based on the so-called model-based filtering techniques [25]. The models may: (1) approximate the nonrandom target maneuver as a random process with certain properties, or (2) describe typical target trajectories by some representative motion models. Examples of the former include the simple white-noise acceleration model [26], the slightly more sophisticated Markov process based Singer model [8], and the more complex interacting multiple model technique [27]. A comprehensive survey of models may be found in Li and Jilkov [25]. When there is some a priori knowledge of the target maneuver, for example, if it is known that the target is maneuvering in a circle, circular motion models like (19) can be used for designing the target state estimator. In general, for the model-based approaches to target state estimation, filter performance may not be satisfactory when the target maneuver does not comply with the model, and every approach can be defeated with a suitably chosen target maneuver. Neural Network-(NN-) based adaptive estimation and filtering techniques for state estimation design have been proposed to compensate for the modeling errors that arise due to nonlinearities and unmodeled dynamics [28, 29]. In these approaches, a nominal time-varying estimator is augmented with the output of an adaptive NN that is trained online with the residuals of the nominal estimator and with delayed values of available system measurements as inputs.

Using an adaptive approach can eliminate the need to have apriori knowledge of the target maneuvering such as those given by (5) and (19).

**5.1. Problem Formulation.** Consider the following bounded single-input-single-output (SISO) nonlinear system:

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + Bg(\mathbf{x}(t), \mathbf{z}(t)), \quad (28)$$

$$\dot{\mathbf{z}}(t) = \mathbf{f}_z(\mathbf{x}(t), \mathbf{z}(t)), \quad (29)$$

$$y(t) = C\mathbf{z}(t), \quad (30)$$

with initial conditions  $\mathbf{x}(0) = \mathbf{x}_0$  and  $\mathbf{z}(0) = \mathbf{z}_0$ . Here  $\mathbf{x} \in Dx \subseteq \mathcal{R}^{n_x}$  and  $\mathbf{z} \in Dz \subseteq \mathcal{R}^{n_z}$  are the states of the system.  $\mathbf{x}$  represents the states that have been modeled, and  $\mathbf{z}$  represents unknown/unmodeled states.  $Dz, Dx$  are compact sets and  $\mathbf{f}_z(\mathbf{x}, \mathbf{z}) : \mathcal{R}^{n_x} \times \mathcal{R}^{n_z} \rightarrow \mathcal{R}^{n_z}$  is unknown, but bounded, and represents the unmodeled dynamics.  $g(\mathbf{x}, \mathbf{z}) : \mathcal{R}^{n_x} \times \mathcal{R}^{n_z} \rightarrow \mathcal{R}$  is unknown, but is uniformly bounded and continuous. It is essentially through  $g(\mathbf{x}, \mathbf{z})$  that the unmodeled dynamics  $\mathbf{f}_z$  couples into the system dynamics. The dimension of the  $\mathbf{z}$  dynamics need not be known, but must be bounded. That is,  $n_z$  is unknown but bounded. The output is represented by  $y \in \mathcal{R}$ . The matrices  $(A, B, C)$  are known, and the pair  $(A, C)$  is assumed to be observable.

The function  $g(\mathbf{x}, \mathbf{z})$  acts as the unknown system input or disturbance to the nominal linear system given by the matrices  $(A, B, C)$ . In the context of target tracking,  $\mathbf{f}_z(\mathbf{x}, \mathbf{z})$  represents unknown/unmodeled target dynamics. The objective is to design a state estimator to estimate the states  $\mathbf{x}$  of the system in (28) with bounded estimation error in the presence of the unknown system input  $g(\mathbf{x}, \mathbf{z})$ . We start by approximating  $g(\mathbf{x}, \mathbf{z})$  using a neural network parametrization given by

$$g(\mathbf{x}, \mathbf{z}) = \mathbf{W}^T \boldsymbol{\sigma}(\bar{\boldsymbol{\mu}}) + \epsilon(\boldsymbol{\mu}), \quad (31)$$

where  $\mathbf{W}$  is the *ideal* weights that results in a function approximation error  $\epsilon$ . We assume that  $\|\mathbf{W}\|_F \leq W^*$ ,  $\|\epsilon(\bar{\boldsymbol{\mu}})\| \leq \epsilon^*$ , and  $|\bar{\boldsymbol{\mu}}| \leq \mu^*$ .  $\boldsymbol{\sigma}(\bar{\boldsymbol{\mu}}) = [\sigma_1(\bar{\boldsymbol{\mu}}), \dots, \sigma_N(\bar{\boldsymbol{\mu}})]^T$  is a vector of sigmoidal functions.  $W^*$  is a bound on the Frobenious norms of the ideal, unknown weights  $\mathbf{W}$ .  $\epsilon^*$  is a bound on the Neural Network function approximation error  $\epsilon$ .  $N$  is the number of neurons.  $\bar{\boldsymbol{\mu}}$  is the input vector given by

$$\bar{\boldsymbol{\mu}}(y(t), d) = [1, \Delta_d^{(0)} y^T(t) \cdots \Delta_d^{n-1} y^T(t)], \quad (32)$$

where

$$\begin{aligned} \Delta_d^{(0)} y^T(t) &= y^T(t) \\ \Delta_d^{(k)} y^T(t) &= \frac{\Delta_d^{(k-1)} y^T(t) - \Delta_d^{(k-1)} y^T(t-d)}{d} \end{aligned} \quad (33)$$

$k = 1, 2, \dots$

$d > 0$  represents a time delay. The sigmoidal functions are smooth and uniformly bounded that is,  $|\sigma_i(\bar{\boldsymbol{\mu}})| \leq 1$ . The above set of equations originating with (31) essentially states that a continuous function of the states of (28) that is,  $\mathbf{x}, \mathbf{z}$  can be approximated to an arbitrary degree of accuracy by a Neural Network over a compact domain by using inputs  $\bar{\boldsymbol{\mu}}(y(t), d)$  that are a finite sample of the output ( $y$ ) history of the system.

**5.2. Adaptive Element augmented Kalman Filter.** One may design a time-varying filter to estimate the states of (28) as follows:

$$\begin{aligned} \hat{\mathbf{x}}(t) &= A\hat{\mathbf{x}}(t) + K(t)(y(t) - \hat{y}(t)) + Bv_{\text{ad}}, \quad \hat{\mathbf{x}}(0) = \hat{\mathbf{x}}_0, \\ \hat{y}(t) &= C\hat{\mathbf{x}}(t), \end{aligned} \quad (34)$$

where the Kalman gain  $K(t)$  may be obtained through the following matrix differential Riccati equation [30],

$$\begin{aligned} \dot{P}(t) &= AP(t) + P(t)A^T - P(t)C^T R^{-1} CP(t) + Q, \\ K(t) &= P(t)C^T R^{-1}, \end{aligned} \quad (35)$$

where  $P(0) = P_0 > 0$ ,  $Q = Q^T > 0$ ,  $R = R^T > 0$  are design constants. The solution  $P(t)$  is bounded, symmetric, positive definite, and continuously differentiable. In (34), the term  $Bv_{\text{ad}}$  represents the Neural Network augmentation to

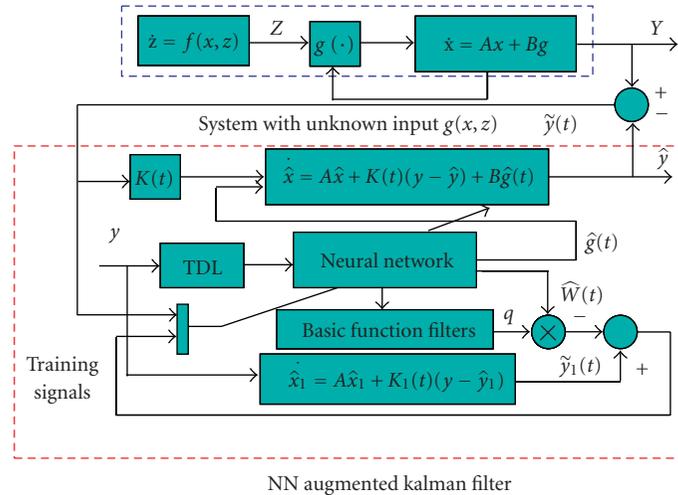


FIGURE 9: Kalman filter augmented with an adaptive element.

the Kalman Filter.  $v_{ad}$  represents the output of the Neural Network and is given by

$$v_{ad} = \hat{\mathbf{W}}(t)^T \sigma(\bar{\mu}), \quad (36)$$

where  $\hat{\mathbf{W}}(t)$  is the estimate of the weight vector  $\mathbf{W}$  and  $v_{ad}$  is designed to approximate the bounded disturbance  $g(\mathbf{x}, \mathbf{z})$ . The difference between the output  $y(t)$  and our estimate of the output  $\hat{y}(t)$ , that is, the residual signal  $\tilde{y}(t) = y(t) - \hat{y}(t)$  may be used as a training signal to train the Neural Network. A second training signal is usually derived to improve the effectiveness of the NN adaptation performance. Further details on the second training signal and the Neural Network weight update laws may be found in [28]. Figure 9 illustrates a block diagram of the system. The TDL block represents the delay line give by (32).

**5.3. Simulation Results.** The detailed equations for this particular rendering of the estimation problem for target tracking may be found in [29]; the results, however, are presented here. In contrast to using an estimate for the target acceleration such as that given by (19), with the adaptive element augmentation, the target  $\mathbf{a}_t$  need not be assumed apriori and is simply treated as unmodeled dynamics  $\mathbf{f}_z(\mathbf{x}, \mathbf{z})$  which results in an unknown input  $g(\mathbf{x}, \mathbf{z})$  that the adaptive element can approximate.

In this scenario, the leader aircraft is flying in a circle in the horizontal plane at a constant heading rate. The follower aircraft is tasked with maintaining specified separation distances along the  $x$ ,  $y$ , and  $z$  axes of the follower body-fixed frame. The follower is first put into the desired formation using only GPS-communicated data of the leader inertial position, velocity, and acceleration. The leader GPS data is communicated at about 5 Hz and is filtered to produce leader state estimates at the rate required (50 Hz) by the follower aircraft guidance and flight control algorithms. Once the leader aircraft is at the desired separation distance, the image processing and target state estimation algorithms

are switched on. The update rate of the image processing in simulation 10 Hz. The estimates of the leader position, velocity, and acceleration from using the vision-based target state estimator are blended in with the corresponding GPS estimates to produce the leader state estimates that are used in the guidance and flight control algorithms for formation keeping. The formation separation commands for the results shown below are given by  $[dx, dy, dz]_{commanded} = [60, 15, 10]$  ft.

Figure 10(a) shows various flags used during the simulation and serves to indicate the method in which this scenario was tested. The IP flag, if 1, indicates the image processing system returned a valid result. If 0, it generally implies that the target was not in view. The use of vision information in the navigation solution was slowly blended into information available from the leader's navigation solution which is communicated to the follower. Hence a zero for useVision indicates a fully cooperative leader and a one indicates a completely uncooperative target. The NN switch flag indicates where the adaptive element is active. At  $t = 140$  s, the vision estimates are blended in (at 50%) and the reliance on vision information is slowly increased, where at about  $t = 290$  s, only vision information is used to estimate the leader's position, velocity, and acceleration. In doing so, Figure 10(b) illustrates the ability of the follower to maintain formation at the commanded  $[dx, dy, dz]$ . Figures 10(c) and 10(d) show the estimates of the leader's velocity and error in those estimates respectively. Similarly, Figures 10(e) and 10(f) show the estimates, of the leader's acceleration and error in those estimates respectively. In general, the ability of the follower to track the desired offset command is sufficient to keep formation with 15 ft in each axis.

Note that an initially cooperative target, that is, using the leader's GPS position (useVision = 0), is only assumed in order to position the leader within the follower's field of view before the experiment may begin. This assumption does not affect the ability of the Neural Network to improve

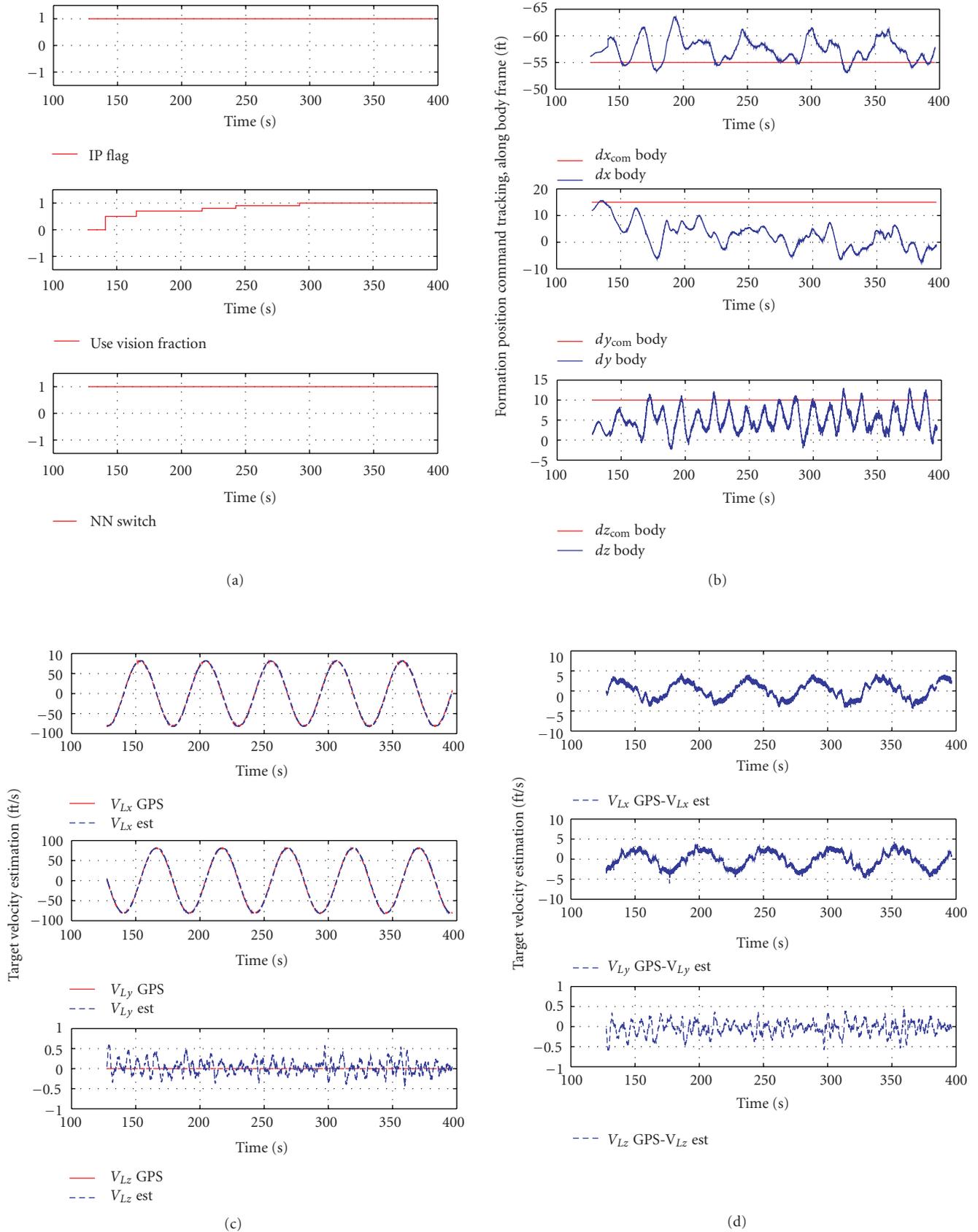


FIGURE 10: Continued.

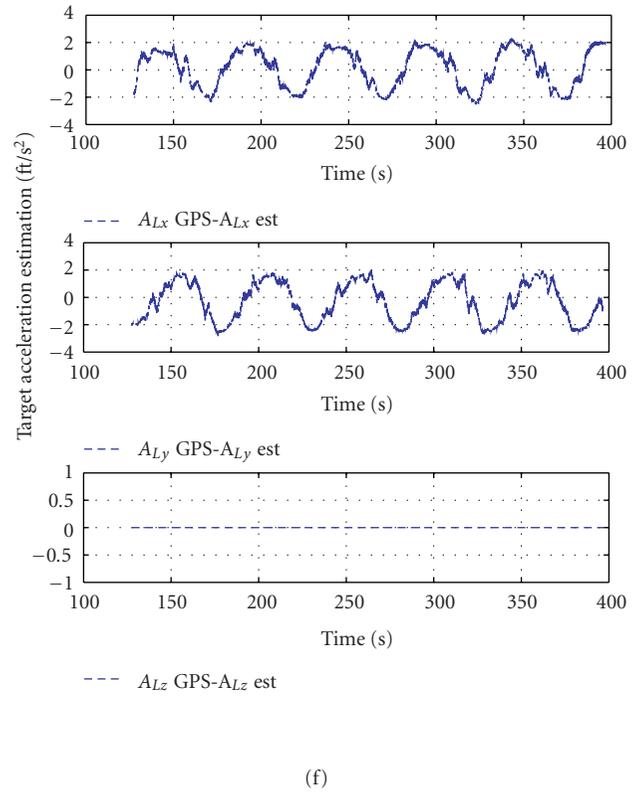
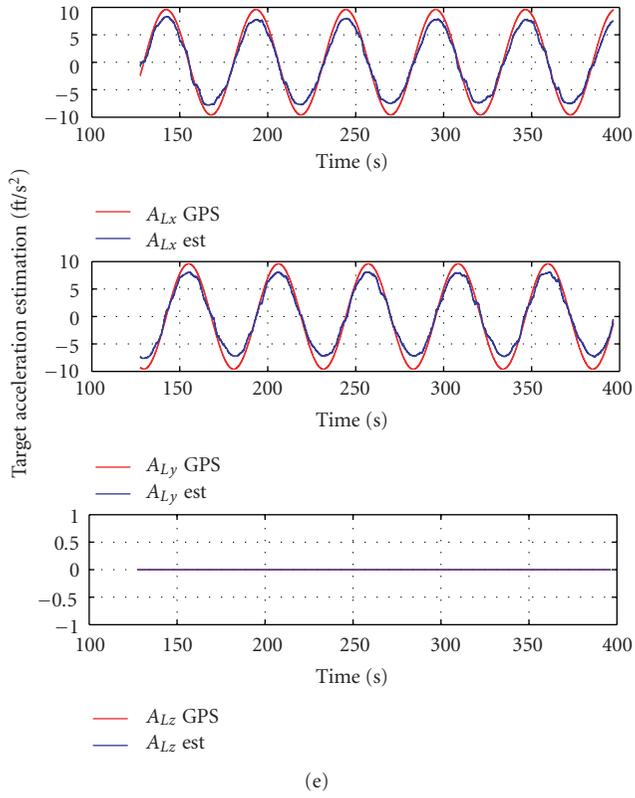


FIGURE 10: Adaptive estimation simulation results.

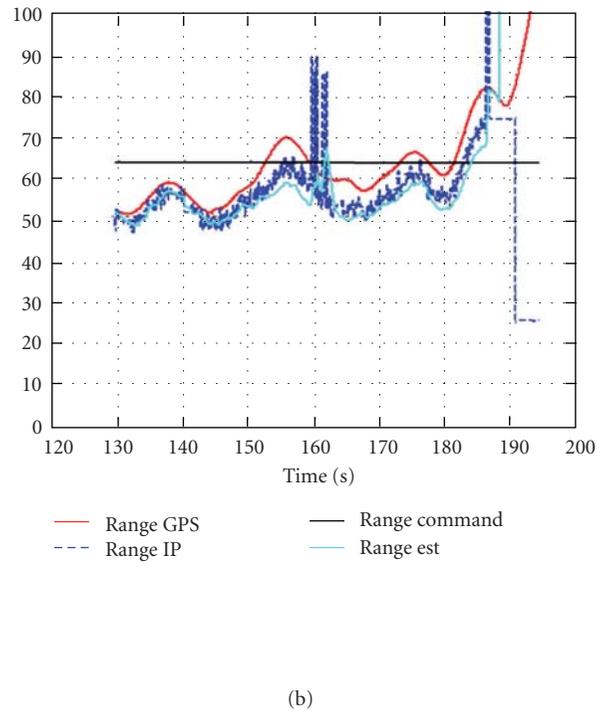
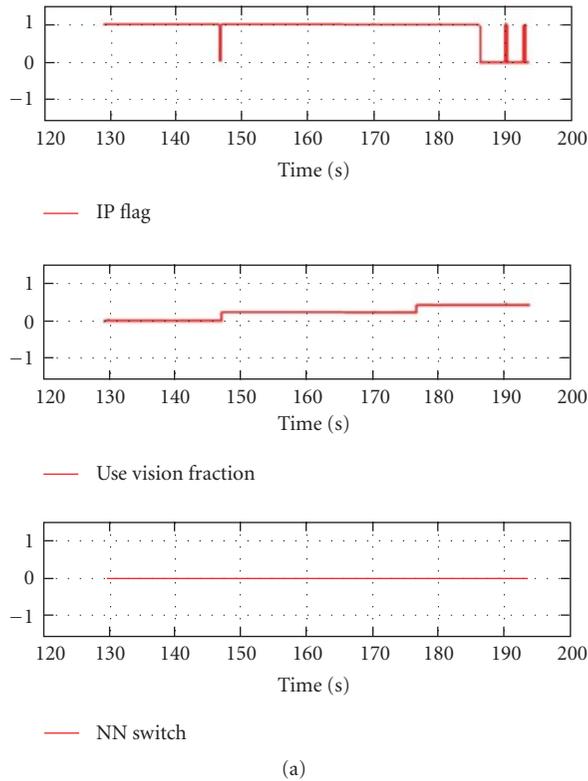


FIGURE 11: Range estimation without the adaptive element.

tracking results. Other methods such as external radar-based information may be used in a real situation where the target is truly uncooperative.

In order to demonstrate the effect of adaptation on this problem, the adaptive element was switched off leaving just the Kalman filter to estimate the leader's position and velocity. In this case only the flags (Figure 11(a)) and the range estimation error (Figure 11(b)) are shown. Even at a low blending in of vision information (20%), the IP flag temporarily goes to zero indicating a loss of vision tracking. As the vision information is further blended in, the range estimates diverges and the leader aircraft drifts out of the field-of-view, indicated by the repeated zeroing of the IP flag, eventually resulting in divergence of the range estimate. This clearly indicates the requirement of an adaptive element when no a priori information on target maneuvers is assumed.

## 6. Conclusions

It is shown that even though the range observability is problematic when using monocular vision, especially when target subtended angle information is not available, it is practical to generate follower maneuvers that improve range observability. A combination of flight test and simulation results are presented for vision-based tracking of an uncooperative target when subtended angle information is available. Initially, a target maneuver model which reflects the circular maneuvers performed in simulation and flight tests is assumed to be available. The achievement of the closed-loop formation flight verified the estimation performance of the vision-based relative navigation filter design. When the requirement for a target maneuver model is removed, the lack of a model was found to be detrimental to leader state estimation performance, and formation flight was not possible. By treating the target as unmodeled dynamics, an adaptive element was augmented with a Kalman Filter allowing leader states to be estimated. The adaptive element was found to be critical in maintaining formation geometry.

## Appendix

### Extended Kalman Filter

Since the 2-D vision-based measurement (7) is a nonlinear function with respect to the 3-D relative state, an extended Kalman filter (EKF) is applied to estimate the relative states from the measurement. The Kalman filter is a recursive solution to the least squares method for a linear filtering problem [31]. Since the filter was introduced by Kalman in 1960, it has been the subject of extensive research and application particularly in the area of autonomous navigation [32]. The EKF is an extension of the standard linear Kalman filter so that it can be applied to nonlinear systems by linearizing the system about the predicted estimate at each time step [30, 33, 34]. Even though the convergence of its estimate cannot be theoretically proven, good estimation performance of the EKF has been demonstrated in many practical applications.

A general formulation of the EKF is presented in this section. Consider the following nonlinear system:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) + \mathbf{w}(t), \quad (\text{A.1})$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \boldsymbol{\nu}_k, \quad (\text{A.2})$$

where  $\mathbf{x}$  is a state vector,  $\mathbf{u}$  is a system input,  $\mathbf{w}$  is a random zero-mean process noise,  $\mathbf{z}_k$  is a measurement at time step  $k$ , and  $\boldsymbol{\nu}_k$  is a discrete measurement noise. Let  $\hat{\mathbf{x}}_k^-$  and  $\hat{\mathbf{x}}_k$  be the predicted and updated estimates of  $\mathbf{x}(t_k)$ , and let  $P_k^-$  and  $P_k$  be their estimated error covariance matrices. A state estimate at  $t_{k-1}$  is propagated to the next time step  $t_k$  through a first-order Euler integration of the original nonlinear system (A.1),

$$\hat{\mathbf{x}}_k^- \simeq \hat{\mathbf{x}}_{k-1} + \mathbf{f}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1})(t_k - t_{k-1}). \quad (\text{A.3})$$

Then the system dynamics and the measurement model ((A.1)-(A.2)) are linearized about  $\hat{\mathbf{x}}_k^-$  and discretized as follows:

$$\mathbf{x}_k = \Phi_k \mathbf{x}_{k-1} + \Gamma_k \mathbf{u}_{k-1} + \mathbf{w}_{k-1}, \quad (\text{A.4})$$

$$\mathbf{z}_k = H_k \mathbf{x}_k + \boldsymbol{\nu}_k,$$

where

$$\Phi_k = e^{F_k(t_k - t_{k-1})},$$

$$\Gamma_k = \int_{t_{k-1}}^{t_k} e^{F_k(t_k - s)} G_k ds,$$

$$\mathbf{w}_{k-1} = \int_{t_{k-1}}^{t_k} e^{F_k(t_k - s)} \mathbf{w}(s) ds,$$

$$F_k = \left. \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_k^-, \mathbf{u}=\mathbf{u}_{k-1}}, \quad (\text{A.5})$$

$$G_k = \left. \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_k^-, \mathbf{u}=\mathbf{u}_{k-1}},$$

$$H_k = \left. \frac{\partial \mathbf{h}(\mathbf{x}_k)}{\partial \mathbf{x}_k} \right|_{\mathbf{x}_k=\hat{\mathbf{x}}_k^-}.$$

Now the linear discrete Kalman filtering algorithm can be applied to the linearized system (A.4). The predicted and updated error covariance matrices and the Kalman gain are calculated by

$$P_k^- = \Phi_k P_{k-1} \Phi_k^T + Q_k, \quad (\text{A.6})$$

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}, \quad (\text{A.7})$$

$$P_k = (I - K_k H_k) P_k^-, \quad (\text{A.8})$$

where  $Q_k$  and  $R_k$  are covariance matrices of the discrete process noise  $\mathbf{w}_k$  and measurement noise  $\boldsymbol{\nu}_k$ . Finally, the updated state estimate is obtained by

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K_k (\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-)). \quad (\text{A.9})$$

Since a camera's field of view is limited and the image processor may sometimes fail to capture the target, the vision-based measurement is not always available. When this happens, only the EKF prediction procedure ((A.3) and (A.6)) is performed. The absence of a measurement corresponds to having a measurement with an infinitely large noise. When  $R_k = \infty$  in (A.7), the Kalman gain  $K_k$  becomes zero. It results in  $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^-$  and  $P_k = P_k^-$ , and nothing will be changed in the EKF update procedure (see (A.8) and (A.9)).

## References

- [1] A. A. Proctor, E. N. Johnson, and T. B. Apker, "Vision-only control and guidance for aircraft," *Journal of Field Robotics*, vol. 23, no. 10, pp. 863–890, 2006.
- [2] C. DeWagter, A. Proctor, and E. N. Johnson, "Vision-only aircraft flight control," in *Proceedings of the 22nd Digital Avionics Systems Conference*, Indianapolis, Ind, USA, October 2003.
- [3] O. Shakernia, R. Vidal, C. Sharp, Y. Ma, and S. Sastry, "Multiple view motion estimation and control for landing an unmanned aerial vehicle," in *Proceedings of the International Conference on Robotics and Automation*, Washington, DC, USA, May 2002.
- [4] F. Caballero, L. Merino, J. Ferruz, and A. Ollero, "Vision-based odometry and SLAM for medium and high altitude flying UAVs," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 54, no. 1–3, pp. 137–161, 2009.
- [5] F. Kendoul, K. Nonami, I. Fantoni, and R. Lozano, "An adaptive vision-based autopilot for mini flying machines guidance, navigation and control," *Autonomous Robots*, vol. 27, no. 3, pp. 165–188, 2009.
- [6] J.-C. Ha, C. Alvino, G. Pryor, M. Niethammer, E. Johnson, and A. Tannenbaum, "Active contours and optical flow for automatic tracking of flying vehicles," in *Proceedings of the American Control Conference (AAC '04)*, pp. 3441–3446, July 2004.
- [7] J. A. Sethian, "A fast marching level set method for monotonically advancing fronts," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 93, no. 4, pp. 1591–1595, 1996.
- [8] R. A. Singer, "Estimating optimal tracking filter performance for manned maneuvering targets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 6, no. 4, pp. 473–483, 1970.
- [9] E. N. Johnson and S. K. Kannan, "Adaptive trajectory control for autonomous helicopters," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 3, pp. 524–538, 2005.
- [10] J. Ha, C. Alvino, G. Pryor, M. Niethammer, E. Johnson, and A. Tannenbaum, "Active contours and optical flow for automatic tracking of flying vehicles," in *Proceedings of the American Control Conference (AAC '04)*, pp. 3441–3446, Boston, Mass, USA, July 2004.
- [11] J. Ha, E. N. Johnson, and A. Tannenbaum, "Real-time visual tracking using geometric active contours for the navigation and control of UAVs," in *Proceedings of the American Control Conference (ACC '07)*, pp. 365–370, July 2007.
- [12] E. N. Johnson, A. J. Calise, Y. Watanabe, J. Ha, and J. C. Neidhoefer, "Real-time vision-based relative aircraft navigation," *Journal of Aerospace Computing, Information and Communication*, vol. 4, no. 4, pp. 707–738, 2007.
- [13] V. J. Aidala and S. E. Hammel, "Utilization of modified polar coordinates for bearings-only tracking," *IEEE Transactions on Automatic Control*, vol. 28, no. 3, pp. 283–294, 1983.
- [14] L. Matthies, T. Kanade, and R. Szeliski, "Kalman filter-based algorithms for estimating depth from image sequences," *International Journal of Computer Vision*, vol. 3, no. 3, pp. 209–238, 1989.
- [15] Y. Watanabe, E. N. Johnson, and A. J. Calise, "Optimal 3-D guidance from a 2-D vision sensor," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pp. 319–328, August 2004.
- [16] A. J. Calise, "Enforcing an algebraic constraint in extended Kalman filter design," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pp. 1879–1891, Indianapolis, Ind, USA, August 2007.
- [17] E. N. Johnson and S. K. Kannan, "Adaptive flight controller for an autonomous unmanned helicopter," in *Proceedings of the AIAA Guidance, Navigation and Control Conference*, 2002.
- [18] E. N. Johnson, M. A. Turbe, A. D. Wu, S. K. Kannan, and J. C. Neidhoefer, "Flight test results of autonomous fixed-wing UAV transitions to and from stationary hover," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pp. 5144–5167, August 2006.
- [19] E. N. Johnson, Y. Watanabe, J. Ha, A. J. Calise, and A. R. Tannenbaum, "Image processor, estimation, guidance, and flight test of vision-based formation flight," in *Proceedings of the 3rd International Symposium on Innovative Aerial/Space Flyer Systems*, 2006.
- [20] E. W. Frew and S. M. Rock, "Trajectory generation for constant velocity target motion estimation using monocular vision," in *Proceedings of the AIAA Guidance, Navigation and Control Conference*, 2003.
- [21] G. L. Mariottini, F. Morbidi, D. Prattichizzo et al., "Vision-based localization for leader-follower formation control," *IEEE Transactions on Robotics*, vol. 25, no. 6, pp. 1431–1438, 2009.
- [22] G. L. Mariottini, S. Martini, and M. B. Egerstedt, "A switching active sensing strategy to maintain observability for vision-based formation control," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 2009.
- [23] A. Bicchi, D. Prattichizzo, A. Marigo, and A. Balestrino, "On the observability of mobile vehicles localization," in *Proceedings of the IEEE Mediterranean Conference on Control and Automation*, Sardinia, Italy, June 1998.
- [24] E. Bryson and Y. Ho, *Applied Optimal Control*, Taylor & Francis, London, UK, 1975.
- [25] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking, part I: dynamic models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333–1364, 2003.
- [26] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*, Wiley-Interscience, New York, NY, USA, 2001.
- [27] H. A. P. Blom and Y. Bar-Shalom, "The interacting multiple-model algorithm for systems with Markovian switching coefficients," *IEEE Transactions on Automatic Control*, vol. 33, no. 8, pp. 780–783, 1988.
- [28] R. J. Sattigeri and A. J. Calise, "Neural network augmented kalman filtering in the presence of unknown system inputs," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, Colo, USA, August 2006.

- [29] R. J. Sattigeri, E. N. Johnson, A. J. Calise, and J.-C. Ha, "Vision-based target tracking with adaptive target state estimator," in *Proceedings of the AIAA Guidance Navigation and Control Conference*, Hilton Head, SC, USA, August 2007.
- [30] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, John Wiley & Sons, New York, NY, USA, 1997.
- [31] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME- Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.
- [32] G. Welch and G. Bishop, "An introduction to the kalman filter," Tech. Rep. TR 95-041, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 2004.
- [33] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Artech House, Norwood, Mass, USA, 2004.
- [34] P. Zarchan and H. Musoff, *Fundamentals of Kalman Filtering: A Practical Approach*, AIAA, Reston, Va, USA, 3rd edition, 2004.