Fall 2022

Uncrewed Aircraft Systems for Autonomous Infrastructure Inspection

Michail Kalaitzakis

Uncrewed Aircraft Systems for Autonomous Infrastructure Inspection

by

Michail Kalaitzakis

Mechanical Engineering Diploma
National Technical University of Athens, 2011

Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy in

Mechanical Engineering

College of Engineering and Computing

University of South Carolina

2022

Accepted by:

Nikolaos Vitzilaios, Major Professor

Michael Sutton, Committee Member

Austin Downey, Committee Member

Ioannis Rekleitis, Committee Member

Dimitris Rizos, Committee Member

Cheryl L. Addy, Interim Vice Provost and Dean of the Graduate School

Dedication

To my family

Acknowledgments

There are many who supported me throughout my Ph.D. journey. First, I would like to thank my supervisor, Dr. Nikolaos Vitzilaios. Thank you for your guidance and support even beyond your role as my academic advisor. Special thanks should also go to my dissertation committee members, Dr. Michael Sutton, Dr. Austin Downey, Dr. Ioannis Rekleitis and Dr. Dimitris Rizos. I was fortunate to work closely with you in many projects and I learned a lot from all of you. Thank you for serving in my committee and for your valuable inputs, comments and suggestions.

I am very grateful to my lab mates in the Unmanned Systems and Robotics Lab. Brennan, Sabrina, Bhanu, Sanim, Keith thank you for all your support and for making the endless hours in the lab fun and interesting. I am also thankful to all the brilliant colleagues with whom I collaborated through these years. Marios, Nare, Alberto, Jason, Sharmin, Bharat and Ian. Thank you for all your help and expertise.

Lastly, I would like to thank my family, to whom this is dedicated. Thank you for being there for me always, even when I am doing thing that might seem strange. Christina, thank you for being the inspiration for this journey and for being with me in every step along the way.

Abstract

Uncrewed Aircraft Systems (UAS) are becoming increasingly popular for infrastructure inspections since they offer increased safety, decreased costs and consistent results, compared to traditional methods. However, there are still many open challenges before fully autonomous, reliable, and repeatable UAS inspections. While a UAS platform has increased mobility and can easily approach hard to reach areas, it has limited range and payload capacity and is susceptible to environmental disturbances. Therefore, current operations are limited to Visual Line of Sight (VLOS) manual inspections that usually result in just a qualitative (visual) assessment of the structure.

The objective of this work is to propose solutions to these limitations in an effort to improve the effectiveness of UAS as an autonomous inspection platform. First, a heterogeneous marsupial robotic system is proposed as a solution to the limited range and flight time of UAS. The proposed system uses an Autonomous Surface Vehicle (ASV) to ferry the UAS close to the area of interest, where the latter can perform an inspection. Combining these two different platforms in a single system takes advantage of the individual strengths resulting on a platform that has the reach and high point of view of a UAS but has the range and operation time of the ASV. The proposed system was extensively tested over a six-month period in field deployments at Lake Murray and at the Congaree River, SC, USA, to validate its capabilities.

As a solution to go beyond visual UAS inspections, a UAS equipped with a Stereo Digital Image Correlation (StereoDIC) system is proposed. StereoDIC is a non-contact non-destructive evaluation method that can accurately measure displacements,

strains, strain rates, and geometry profiles. StereoDIC has become a method of choice in experimental mechanics with most studies performed in controlled lab environments with controlled lighting and stationary cameras positioned in the appropriate distance from the measured object.

A prototype is built and tested in a lab setting to investigate its effectiveness and understand the challenges that might arise from the deployment of such a system. A comparative study using a stationary StereoDIC system validates the accuracy of the measurements while the acquisition of measurements showing the onset and evolution of defects and the dynamic response of the structure during a harmonic oscillation verifies the ability of the system to produce a quantitative assessment.

Finally, using the lessons learned from the lab experiments, a new, upgraded, StereoDIC enabled UAS is developed for outdoor deployment and infrastructure inspection. To allow safe field deployments, the new system features a state estimation framework enabling operation in GNSS degraded environments while also estimating any external disturbances acting on the platform. These disturbances are utilized by the controller to make the platform adaptable to challenging weather conditions. The new system was deployed over an eight-month period at a railroad bridge in Columbia, SC. Initial data were collected that guided the investigations of effective speckle pattern applications. Experimental results from bridge measurements, while loaded from crossing trains, are presented and discussed.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

The American Society of Civil Engineers (ASCE), in their 2021 comprehensive assessment [1], rate America's overall Infrastructure with a C- while bridges are rated with a C. Out of the more than 600.000 bridges in the United States, 42% are 50 years old or older, while 7.5% are structurally deficient. Since the cost of maintaining a bridge is only a fraction of the cost of building a new one, preventive maintenance is the new priority. As such, a systematic inspection using Nondestructive Evaluation (NDE) techniques, is of critical importance.

The principal NDE technique used today is visual inspections from bridge inspectors. A 2001 report on the reliability of visual inspections for highway bridges from the Federal Highway Administration, found that the results from routine inspections have a significant variability [66]. Factors such as inspector stress levels, comfort with heights, visual and color acuity as well as traffic and light conditions and the structure accessibility can affect the reported condition of the bridge. The challenging conditions and the subjective nature of this type of inspections leads to highly inconsistent evaluations.

Recently, the Minnesota Department of Transportation conducted a study on utilizing Unmanned Aircraft Systems (UAS) (commonly referred to as aerial drones) to improve the quality of bridge inspections [60]. Their findings showed that drone inspections can produce improved deliverables by supplementing the traditional in-

spection report with substantial amounts of data that can be compiled into 3D reconstructions of the bridge or high-quality maps of areas of interest. Presenting the inspection results in a graphical manner, makes them easier to be reviewed and understood by owners and engineers. Another important finding of this report is that UAS inspections can significantly reduce the inspection costs. Traditional bridge inspections usually require the use of man lifts or Under Bridge Inspection Vehicles (UBIV). The cost of purchasing or renting such equipment is significantly higher than that for drones. Moreover, the use of UBIVs requires traffic control that adds to the cost of the inspection. The average cost savings for UAS inspections were reported at 40%. Finally, the report recognizes the dramatic improvement of safety for the public and inspection personnel when UAS are used instead of traditional techniques.

Past studies, as the ones presented above, have clearly shown the many advantages of drone inspections over traditional methods. As a result, UAS infrastructure inspections is a rapidly growing industry with a market value of $45 billion [88]. Apart from bridge inspections [41, 86], UAS are used for inspecting power lines [58], gas pipe lines [2], railroads [9], as well as in energy efficiency surveys for buildings [74]. Companies like Skydio [5] offer inspection products that combine UAS with image processing software that can scan a structure and generate a map or a 3D reconstruction. However, there are still many areas for improvement. In an extensive survey of UAS civil applications [88], the authors identify the main challenges and areas of improvement in the field of UAS inspections:

- UAS have limited power, range, flight time and payload capacity thus limiting their application envelope.
- UAS rely on Global Navigation Satellite Systems (GNSS) for their positioning, making it challenging to deploy in GNSS degraded or denied environments such as indoors or close to large structures.

- Lack of research attention on the cooperation of multiple platforms that could widen the inspection scope.
- Advance data collection to achieve more efficient inspections.

The results of the UAS visual inspections are qualitative and are used primarily to identify large scale defects and damage to a structure. However, there are vision-based methods that can provide quantitative results. Point tracking, Structure from Motion, and Phase-Based Video Motion Processing are some of such methods that have been employed in inspections. Stereo Digital Image correlation is a vision-based method used to measure full-field 2D and 3D coordinates [85]. Using the measured coordinate fields, it can calculate quantities such as displacements, strains, strain rates, and geometry profiles. Using StereoDIC for structural inspections will make possible to identify and measure microscopic defects as well as measure the structure's response under different loading conditions.

1.2 Contributions

This dissertation proposes solutions to the main challenges faced in autonomous UAS inspections. The aim is to enable UAS to reach remote areas and extend their range of operation, allow safer navigation on GNSS degraded environments and push towards quantitative assessments. Specifically, the contributions of this work are:

- The development and deployment of a heterogeneous marsupial robotic system comprised by an Autonomous Surface Vehicle (ASV) and a UAS for inspection of structures adjacent to water bodies [47]. This work addresses the limited range and flight time of a UAS and adds to the research field of using multiple platforms to improve the inspection scope.
- An experimental comparison between the state-of-the-art fiducial markers for pose estimation along with the development and public release of a Robot Oper-

ating System (ROS) package for a novel fiducial marker [50, 51]. The motivation for this work is to improve the position estimation of UAS in GNSS denied or degraded environments.

- The development of a semi-autonomous UAS, equipped with a StereoDIC system for infrastructure inspection and assessment [49, 48]. The proposed system has been tested in lab experiments to validate its effectiveness on producing accurate displacement and deformation results.

- The development and field deployment of an updated StereoDIC enabled semi-autonomous UAS. The updated version improves on all the limitations of the prototype, while the most important change is the development of a state and disturbance observer and a model predictive position controller. The updated control framework allowed the system to be deployed in the field and acquire data even under challenging wind conditions.

1.2.1 Publications

A list of the publications produced related to this dissertation is presented here in reverse chronological order:

1. **Michail Kalaitzakis**, Nikolaos Vitzilaios, Dimitrios C. Rizos, and Michael A. Sutton, *"Drone-Based StereoDIC: System Development, Experimental Validation and Infrastructure Application"*, Experimental Mechanics, April 2021

2. **Michail Kalaitzakis**, Brennan Cain, Sabrina Carroll, Anand Ambrosi, Camden Whitehead, and Nikolaos Vitzilaios, *"Fiducial Markers for Pose Estimation"*, Journal of Intelligent & Robotic Systems, March 2021

3. **Michail Kalaitzakis**, Sabrina Carroll, Anand Ambrosi, Camden Whitehead, and Nikolaos Vitzilaios, *"Experimental Comparison of Fiducial Markers for*

Pose estimation", 2020 International Conference on Unmanned Aircraft Systems (ICUAS), Athens, Greece, September 2020

4. **Michail Kalaitzakis**, Brennan Cain, Jason Moulton, Ioannis Rekleitis, and Nikolaos Vitzilaios, *"A Marsupial System for Marine Surveying and Inspection"*, Journal of Field Robotics, May 2020

5. **Michail Kalaitzakis**, Sreehari Rajan Kattil, Nikolaos Vitzilaios, Dimitris Rizos, and Michael Sutton, *"Dynamic Structural Health Monitoring using a DIC-enabled drone"*, 2019 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, June 2019

A list of publications produced not directly related to this dissertation follows in reverse chronological order:

1. Kazi R. I. Sanim, **Michail Kalaitzakis**, Bhanuprakash Kosaraju, Zechariah Kitzhaber, Caitlyn English, Nikolaos Vitzilaios, Michael Myrick, Michael Hodgson and Tammi Richardson, *"Development of an Aerial Drone System for Water Analysis and Sampling"*, 2022 International Conference on Unmanned Aircraft Systems (ICUAS), Dubrovnik, Croatia, June 2022

2. Marios Xanthidis, **Michail Kalaitzakis**, Nare Karapetyan, James Johnson, Nikolaos Vitzilaios, Jason M. O'Kane, and Ioannis Rekleitis, *"AquaVis: A Perception-Aware Autonomous Navigation Framework for Underwater Vehicles"*, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, September 2021

3. Brennan Cain, **Michail Kalaitzakis**, and Nikolaos Vitzilaios, *"MK-RRT*: Multi-Robot Kinodynamic RRT* Trajectory Planning"*, 2021 International Conference on Unmanned Aircraft Systems (ICUAS), Athens, Greece, July 2021

4. Sabrina Carroll, **Michail Kalaitzakis**, and Nikolaos Vitzilaios, *"UAS Sensor Deployment and Retrieval to the Underside of Structures"*, 2021 International Conference on Unmanned Aircraft Systems (ICUAS), Athens, Greece, July 2021

5. Bharat Joshi, Sharmin Rahman, **Michail Kalaitzakis**, Brennan Cain, James Johnson, Marios Xanthidis, Nare Karapetyan, Alan Hernandez, Alberto Quattrini Li, Nikolaos Vitzilaios, and Ioannis Rekleitis, *"Experimental Comparison of Open Source Visual-Inertial-Based State Estimation Algorithms in the Underwater Domain"*, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, November 2019

6. Jason Moulton, Nare Karapetyan, **Michail Kalaitzakis**, Alberto Quattrini Li, Nikolaos Vitzilaios and Ioannis Rekleitis, *"Dynamic Autonomous Surface Vehicle Controls, Aiding Disaster Response and Enabling Responsible Port Growth"*, 12th Conference on Field and Service Robotics, Tokyo, Japan, August 2019

Chapter 2

Background

The routine inspection of civil infrastructure such as buildings, bridges and tunnels is critical to assess their condition and ensure their proper maintenance. The standard inspection methods have safety concerns for both the inspectors and the public. Moreover, studies have shown that the resulting assessments have increased variability and thus are not always reliable [36, 72]. Finally, there are increased direct and indirect costs associated with traditional inspections as well as disruption of the normal operation of the structure.

To increase safety, lower costs and move towards quantitative measurements, robot inspections have been increasingly studied in the past years. While with a first glimpse robots look like an ideal platform for inspections, there are still many challenges towards achieving fully autonomous repeatable robotic inspections. Many different platforms have been used in robotic inspections, from ground vehicles [38], to underwater [53] and aerial [88].

Aerial robots, and especially rotor-crafts, have the best mobility. On the other hand, aerial platforms are the ones most susceptible to environmental disturbances. This is the main reason why in most cases of UAS inspections, the system is not autonomous but rather remotely operated [56]. UAS are primarily used in non-contact visual inspections, while there is active research in aerial manipulators that are able to physically interact with their environment [79, 70] and UAS used for sensor deployments [91, 39, 16].

2.1 UAS Infrastructure Inspection

Since the initial stages of research in UAS inspections, the safe navigation in a GNSS degraded environment was of great concern. Since the platform was already equipped with a camera for the inspection, vision-based localization and control were explored. In [14], the authors proposed calculating the position of the drone by tracking a set of features in consecutive images and finding the homography transformation. The system, based on a helicopter UAS, was then used to collect images of the face of a building that were then used to create a rectified map. The same system equipped with infrared cameras was later used in [23], to assess the energy efficiency of buildings.

A similar approach was explored in [65]. The authors developed a visual servoing framework based again on the calculated homography of a flat surface. The proposed controller imposed orientation limits on the platform, nevertheless it was successfully deployed on a helicopter UAS for a bridge inspection. The UAS acquired images were presented to bridge inspectors for the assessment of the bridge. The authors also present images where cracks are detected on the structure walls.

Apart from mappings and reconstructions of the inspected area, crack detection is one of the primary findings of UAS inspections. Crack detection on images is usually based on gradients. The Sobel and Prewitt edge detection algorithms have been used for identifying cracks on images [104], [45]. More advanced methods, capable of removing noise from shadows and connecting crack segments have also been developed [103]. In [54], a visual method to detect fatigue cracks from videos was proposed. The method tracks the motion of multiple features in an series of images while the inspected structure is under a changing load. Then by grouping neighbouring features and calculating the difference in their motion, the authors can separate features that are around a crack. If the camera is static, the specimen does not have out of plane motion and the pixel magnification is known, the method can also measure the size

of the crack as well as how much it changes due to the altering load.

In [25], a small UAS is used to take multiple images of the face of a building. The images are then processed with an image stitching algorithm to produce a map of the building. For damage detection, the images are processed using edge detection and other filtering techniques to highlight the cracks in the images. The authors acknowledge that the crack detection algorithms used are far from perfect and they may lead to false positives since an edge might be registered as a crack.

In [98], a method to identify cracks in areas susceptible to damage. The method proposed first identifies specific areas of interest that, by prior knowledge, are more susceptible to damage, narrowing down the search area. To autonomously detect the defective areas, an object detection algorithm is developed. The algorithm considers the expected shape and orientation of a defect, to reduce the generated false-positives. The method was tested using a steel beam with multiple bolts and the defects that were searched for were cracks that originated specifically from the holes of the bolts.

Recently however, there have been studies that either track the progression of static defects through time, or make dynamic measurements when a structure is under load using vision-based techniques. In [24] a low-cost commercial UAS is used to for quantitative evaluations. The onboard camera was used to acquire images and a point tracking method along with a set of markers was used to identify the target structure. By tracking these markers, the size, shape and deformation of the structure was measured. The method was tested in both static situations like crack detection and in dynamic scenarios like the deformation of a beam under load. When the results were compared with the measurements from a static DIC system, significant error levels were noted. Another significant remark was that when the test was repeated with the drone resting in front of the specimen, the measurements were more accurate compared to when the vehicle was flying during the data acquisition process.

A method to make displacement measurements using a commercial UAS is presented in [99]. The proposed method starts by taking measurements of the target displacement while the drone hovers in place. Then, static features from the background are used to calculate the camera motion. Finally, the results from the first two phases are combined to get the absolute displacement of the target. The method was testes in lab using a servo-hydraulic simulator reproduce revenue-service train traffic of a pin-connected steel truss bridge. The root-mean-square error of the process was $2.14mm$. The authors list camera specifications like the limited frame rate and the rolling shutter effect of the CMOS sensor and the out of plane motion of the structure as reasons for the low accuracy of the measurements.

2.2 Digital Image Correlation

Digital Image Correlation (DIC) [85] is based on image template matching to track motions on the surface of a deforming object. By tracking the surface displacements, DIC can build up full field 2D and 3D deformation vector fields and strain maps. The template matching algorithms used are highly optimized, taking into account the physics of the deformation processes. Since the motions expected in engineering applications are microscopic, the DIC measurements are required to have a high enough resolution with minimal systematic errors. Since its inception in the early 1980s, the method has matured and today there is software that can efficiently obtain sub-pixel resolutions that allow measurements down to one part per million of the field of view.

To perform DIC, a unique correspondence between features in two images must be found. Consequently, a surface that produces images with high contrast, rich in texture with non-repeating features is needed. While some surfaces may naturally display these properties, it is customary practice to prepare the area of interest by applying a random speckle pattern. The pattern applied needs to adhere with the

surface and to deform with it.

DIC can be performed using a single imaging sensor to acquire in-plane deformations. However, even small out of plane motions can introduce significant errors in the measurements. Combining stereo-vision imaging with DIC concepts can mitigate these effects, allowing StereoDIC to acquire three dimensional measurements. StereoDIC has been extensively used in a wide field of applications ranging from material studies and aircraft wing loading to deformation measurements of small soft tissue.

2.3 UAS and Digital Image Correlation

To the best of our knowledge, the first ever mention of deploying a DIC system using a UAS was in [63]. DIC is compared to other techniques used in lab settings to calculate displacements and deformation, such as laser shearography and speckle interferometry, for their applicability in field deployments. DIC is deemed the most suitable method due to the simplicity and low cost of the setup. The authors propose the deployment of the DIC system on small flying vehicles to reach hard to access areas as a cheaper and safer alternative to setting up scaffolding.

The first study with a drone deployed DIC system is [75]. A UAS equipped with a custom-made camera system is proposed to track the progression of cracks on bridges. Measurements on the target locations were obtained at regular time intervals, while the same defects were measured by experts using calipers. The overall results presented, were comparable in accuracy with the measurements taken with dial calipers and thus outperformed any results from visual inspection. The most significant challenge faced during the deployments was piloting the UAS and holding it in position for the measurements. The authors observed that the addition of the vision system increased flight instabilities that in addition to being deployed to a GNSS degraded environment, made it challenging to handle. As a result, some of the measurements acquired were obtained while the imaging system was held in place by

hand rather than while it was airborne.

To increase the system stability and payload capabilities, the authors in [17], utilized an octocopter with increased payload capacity. As a result, their platform was able to hover in front of the target and acquire the desired data. To simplify the system and make it usable by a single operator, they added a LiDAR along with a microcontroller to trigger the cameras when the platform was at the desired distance from the target. In the experimental setup, a simple target was deformed while using a ruler to confirm that the DIC measurements were in reasonable agreement with an independent metric.

In [20], a UAS is used in an experimental setup to get vibration measurements of a bridge model. To achieve this, the authors correct the images using a homography-based method in order to remove the effect of the drone motion. Then, they use DIC to calculate the displacement time history of the nodes of the model bridge which they later use to find the vibration frequency. To find the homography transformation needed to rectify the images, the authors place a fixed square target in front of the model bridge. By tracking the four corners of the target, a transformation is calculated that can rectify the image, removing the effect of the drone motion. The authors compare the results acquired from the UAS with that from a static DIC to validate their results.

Chapter 3

A Heterogeneous Marsupial Robot for Inspections close to Water Bodies

3.1 Introduction

One major weakness of UAS is that they have limited power and payload capacity that in turn limits their range, flight time and range of sensors that they can carry. Other autonomous platforms, such as Unmanned Ground Vehicles (UGV) and Autonomous Surface Vehicles (ASV) have superior range and payload capacity and thus can reach even remote locations. However, such platforms have a limited view of their environment and are restricted to only specific types of terrains. One way to overcome the individual limitations of each platform is to combine them together. This chapter addresses the problem of the UAS limited range by proposing a marsupial system composed of a ASV and a UAS to inspect remote structures close to water bodies.

In a marsupial system, a smaller "weaker" robot is placed on a larger robot to increase the capabilities of the overall system. They were first introduced by Anderson, McKay, and Richardson in [3], where a large UGV would carry and deploy a smaller UGV to reach areas the larger one could not reach. The term "Marsupial Robot" was coined by Murphy *et al.* [69] to describe a system of a parent and child robot. A survey of marsupial systems is presented in [42], where the distinct roles of the members, their use in various environments, and the advantages they present are discussed.

In our case, we use an ASV to ferry a UAS in areas of interest and once there, the UAS is deployed to perform the inspection and collect the data. After the inspection is completed, the UAS returns and docks to the ASV to end their mission or visit the next area of interest. This setup takes advantage of the increased range of the ASV extending the total mission time as well as the higher point-of-view of the UAS and its ability to reach all areas of interest. When the system is deployed, the following measurements are obtained: the position of the ASV, the position of the UAV, and the relative pose between the two from a vision based tracker. By taking advantage of the extra information, we can increase the accuracy of the system and its robustness in the event of GNSS signal loss; this process is termed Cooperative Localization (CL) [76, 77].

The work presented here was published in [47]. The remainder of this Chapter is organized as follows. Section 3.2 details the system design and the two platforms used. Section 3.3 presents the system's Cooperative Localization framework, while Section 3.4 presents the system control including the high-level autonomy, designed safety features and inspection missions and the UAS flight controller. Finally, Section 3.5 presents results from the CL method, the flight controller and more importantly results from two sample infrastructure inspections performed by the proposed system.

3.2 System Design

The two members of the marsupial system are shown in Figure 3.1. The ASV is the AFRL Jetyak [67] and the UAV is the DJI Matrice 100[1]. The AFRL Jetyak is an ASV based on the Mokai Es-Kape[2], a 3.6 meters long boat with a seven horsepower, four stroke engine [67]. Its operation time can vary between 4 and 18 hours depending

[1] `https://www.dji.com/matrice100`

[2] `http://www.mokai.com/mokai-es-kape/`

Figure 3.1: A view of the UAV-ASV marsupial system where the UAV follows the ASV at a safe distance during field trials.

on the speed of the boat and the maximum payload capacity is 163kg. The system is augmented with a Pixhawk PX4 micro-controller and an RFD900+ radio. This allows for remote and autonomous operations of the vehicle as well as telemetry broadcasting. The Jetyak system architecture can be seen in Figure 3.2. For the purpose of this project, a $1.2m \times 1.2m$ landing platform was designed and built in our lab. The platform is attached to the top of the Jetyak behind the mast. The platform features a quick release mechanism as well as a folding surface, for easy transportation and quick installation on the boat.

The DJI Matrice 100 is a development-focused platform for autonomous aircraft research. The onboard computer used is a DJI Manifold, based on the NVidia Tegra TK1. A Zenmuse Z3 gimbal camera is attached for data gathering and visual position estimation, and an RFD900+ radio is used for communications with the Jetyak and the Ground Control Station (GCS). All the peripheral devices, as well as the UAV's N1 Flight Control System, are connected to the onboard computer. The computer is able to access the sensor data and send high level commands to the drone using the DJI Onboard SDK. The overall architecture of the UAV system, with its individual components can be seen in Figure 3.3. The system is designed in a highly modular

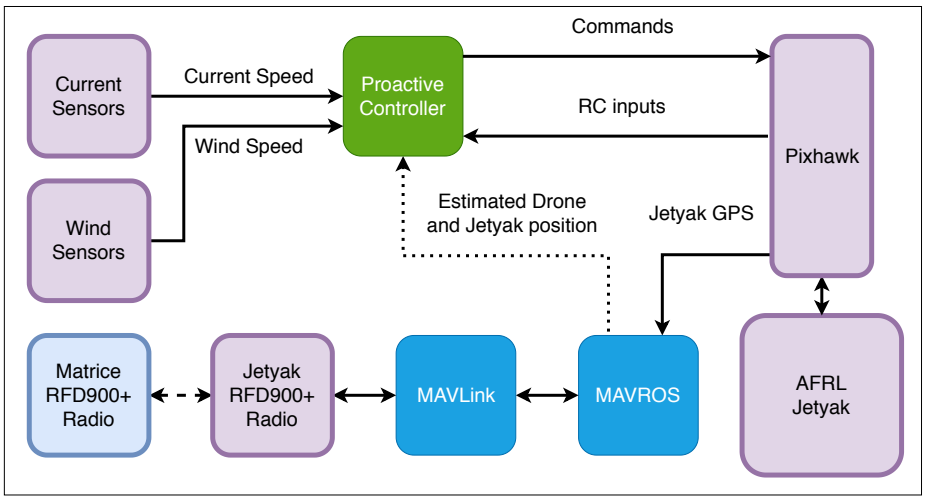


Figure 3.2: Block diagram of the ASV system architecture. The blue blocks are open-source ROS layers, the green blocks are the developed components, the magenta blocks are the hardware peripherals of the ASV and the cyan one are hardware peripherals of the UAV. The dotted line denotes the use of the cooperative localization data in the Jetyak controller.

way, that allows wrappers to be written for each hardware component, such as the gimbal camera or the flight control system. This increases the interoperability of the system.

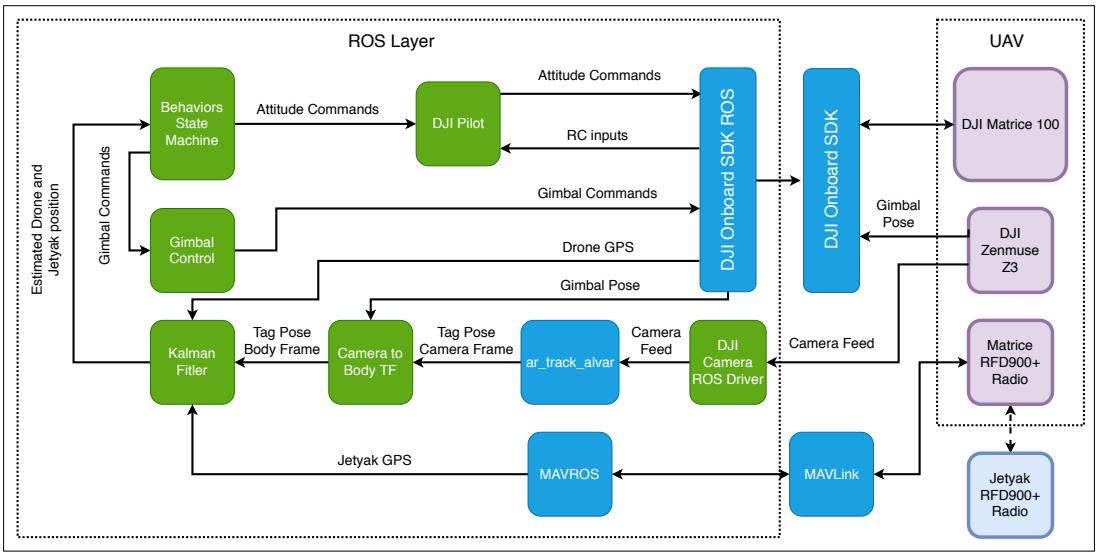


Figure 3.3: Block diagram of the developed UAV system architecture. The blue blocks are open-source ROS layers, the green blocks are components developed for this project, the magenta blocks are the hardware peripherals of the UAV and the cyan one are hardware peripherals of the ASV.

The two platforms and the GCS communicate over MAVLink. The connection allows messages to be exchanged either directly from one node to the other, or indirectly through another node. ROS is used as a middle-ware to develop the systems for both platforms as well as the communications between them. Radio broadcasts can be accessed using MAVROS, a ROS package that provides the communication drivers for MAVLink.

3.3 Cooperative Localization Framework

Missions in marine and especially in freshwater environments require operating close to vegetation and large structures. In such environments, GNSS positioning like GPS, Galileo, etc., can be unreliable or even unavailable. Moreover, on long missions the quality of the GNSS positioning is expected to change. At the same time, these environments are visually challenging with the vegetation generating extreme lighting variations between shadows and direct sunlight resulting in lens flares and overexposed images, while the water is generally featureless and there is an increased chance of reflections on the water surface [18]. To ensure the safety of the autonomous landing and to increase the effectiveness of navigating in such environments, a robust localization framework is needed that will be able to detect and reject outliers as well as to adapt to the changing quality of the sensor measurements.

Our main motivation for cooperative localization is to get an accurate relative position between the two platforms that is crucial for the precision landing maneuvers. Cooperative localization can also be used to assist navigation in GNSS degraded or denied environments. For example, when there is a need to traverse under a large bridge, the drone can be deployed to a position with a better GNSS signal and to provide a more accurate position estimation to the Jetyak by maintaining a visual tether. Once the quality of the GNSS signal has been restored on the Jetyak side, the drone can fly over the bridge and return to land on the Jetyak.

3.3.1 Vision-based localization

Both the UAV and ASV, have a built-in algorithm that fuses the GNSS position and the inertial sensor measurements to get a more accurate positioning of the platform. However, since the two systems use different sensors and are expected to operate in different regions at times, localization based solely on this data is unreliable for precision maneuvers. Visual position estimation methods such as Visual Odometry (VO) are often used to increase the accuracy of the localization framework. In marine environments, however, VO is challenging and visual methods that use specific features and patterns are more appropriate. The most common approach is fiducial marker pose estimation. Fiducial markers can increase the accuracy and robustness of a localization system by providing an easily recognizable feature with embedded fault detection. They have been extensively used in the literature to provide reliable position estimation for challenging maneuvers such as autonomous landing [93, 55, 4, 102].

An extensive experimental comparison of the four markers (ARTag[94, 27], AprilTag [71, 95], ArUco [32], and STag [10]) that represent the state-of-the-art was performed. These markers are evaluated on their accuracy, detection rate and computational cost in several scenarios that include simulated noise from shadows and motion blur. Different marker configurations, including single markers, planar and non-planar bundles and multi-sized marker bundles were also considered. The results of this study are presented in Chapter 4.

Following the example of [4], a bundle of six ARTag markers [27] of various sizes is placed on the ASV. Specifically, a set of four ARTAg markers, $52cm \times 52cm$ each, are placed on the landing platform while a set of two markers, $13cm \times 13cm$, is attached at the base of the mast. The landing platform bundle is used to provide a visual tether over long distances while the mast bundle is used when the drone flies in close proximity to the Jetyak. ARTag was chosen over other fiducial markers like April-

Tag [71, 95] for their efficient and fast tracking performance. Our experimentation found that in the deployed hardware onboard the drone, ARTag tracking is providing position measurements at $13Hz$ while AprilTag tracking has a lower rate of $5Hz$.

Visual position estimation using a independently controlled gimballed camera

The most important subsystem of the UAV is the gimballed camera. The camera is being used to gather data during infrastructure inspections and surveys and most importantly it is used to provide an estimate of the relative position between the two platforms. While in most cases where cameras are used to provide position measurements the cameras are rigidly attached on the platform [80, 93, 26], in our case the camera can move independently of the platform.

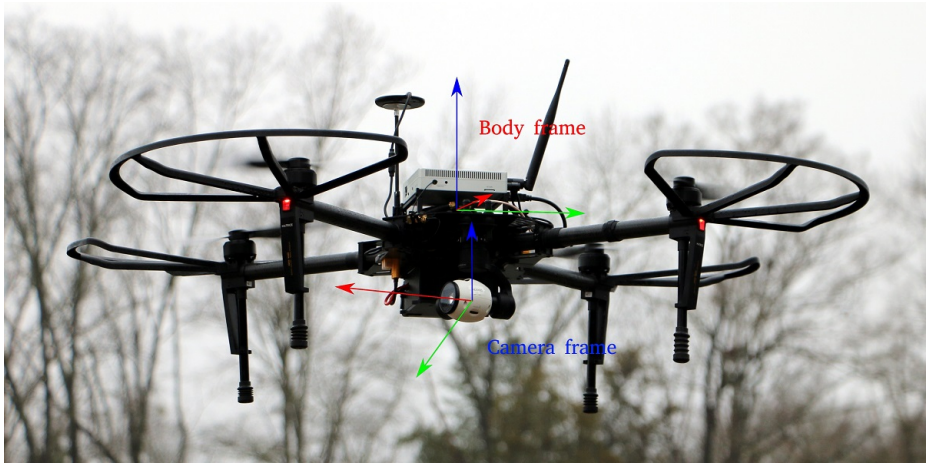


Figure 3.4: The body and camera coordinate systems shown on the UAV. Since the camera is mounted on a gimbal it is free to rotate relative to the body.

Having the camera able to move independently of the platform, provides the advantage of being to able to track the target even in situations where the drone is not facing the boat. Moreover, when the drone flies close to the platform, even small movements could lead to tracking loss with a stable camera. With a moving camera, the gimbal can rotate in order to keep the target always in the camera's field of view and thus maximizing the rate of data from the visual position estimation algorithm.

By tracking the relative orientation of the gimbal frame to the body frame, we can transform the position measurements from the camera frame to the body frame. The UAV used and the gimballed camera with the two coordinate systems are shown in Figure 3.4.

In most cases where gimbals are used in UAVs, their role is to compensate for the motion of the UAV and to keep the camera's orientation. In our case, the gimbal is independently controlled in order to keep a point of interest in frame. Two different control scenarios were implemented. First, if the target is already in frame, the gimbal is rotated as to always keep the target in the center of the image. When the target is not in the line of sight of the camera but its position is known, the gimbal moves so that the projection of the target aligns with the center of the image. In the case of tracking the Jetyak when it is not in the line of sight of the drone, the position of the boat from the cooperative localization framework is used to control the gimbal.

3.3.2 Framework design

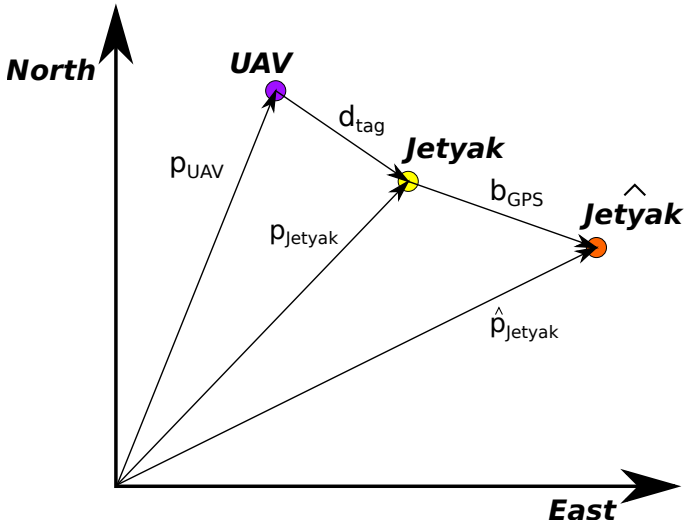


Figure 3.5: The scheme used in the cooperative localization framework. The purple dot is the estimated position of the drone in the local ENU system while the yellow dot is the estimated position of the Jetyak in the same system. The orange dot is the measured position of the Jetyak before it is corrected by subtracting b_{GPS}.

A cooperative localization method based on a Kalman Filter is used throughout all the experiments in this work. The designed framework fuses the measured position of the two platforms, the measured velocity of the drone and the relative position of the platforms acquired from a visual positioning method. As described above, the Jetyak features a bundle of six ARTag markers that are detected and tracked to provide the relative position of the two platforms. During the system setup, it was noted that there was a significant offset between the measured position of the Jetyak and the measured position of the drone even when the sensors were placed next to each other. Furthermore, the quality of the GNSS measurements of the UAV were superior to that of the Jetyak both indoors and outdoors. As a result, an extra state was added to the filter to track the offset on a local East-North-Up (ENU) frame.

Figure 3.5 shows the localization scheme used with:

$$p_{UAV,GPS} = p_{UAV} + w_{UAV} \tag{3.1}$$

$$p_{Jetyak,GPS} = \hat{p}_{Jetyak} + w_{Jetyak} \tag{3.2}$$

$$p_{Jetyak} = \hat{p}_{Jetyak} - b_{GPS} \tag{3.3}$$

where $p_{UAV,GPS}$ is the measured position of the drone with p_{UAV} the true position and w_{UAV} the measurement noise. The measured position of the Jetyak is $p_{Jetyak,GPS}$, with $\hat{p}_{Jetyak}$ the Jetyak position before it is corrected, $w_{Jetyak,GPS}$ the sensor noise, p_{Jetyak} the corrected position and b_{GPS} the offset. The state of the filter is:

$$x = \begin{bmatrix} p_{UAV}^T & v_{UAV}^T & \hat{p}_{Jetyak}^T & v_{Jetyak}^T & b_{GPS}^T \end{bmatrix}^T \tag{3.4}$$

where v_{UAV} is the 3-dimensional velocity of the drone on the same ENU frame and v_{Jetyak} is the 2-dimensional velocity of the Jetyak. Since the Jetyak can only move on the surface of the water, only minor changes in its altitude are expected due to any waves. Thus, its velocity in the z-axis is assumed to be zero. While the AR tag measurements provide information about both the position and the orientation of the

tag, only the position measurements were used. The attitude of the two platforms was estimated by using only their onboard sensors. The measurement from the AR tag tracking is finally defined as:

$$d_{tag} = \hat{p}_{Jetyak} - p_{UAV} - b_{GPS} + w_{tag} \tag{3.5}$$

with w_{tag} the noise of the tag measurement.

Outlier Rejection and Adaptive Measurement Covariance Matrix

To operate safely in marine and freshwater environments and close to structures, the localization framework must be robust to outliers. Outliers are expected in the GNSS position measurements in GNSS degraded environments but more frequently in the marker measurements. Reflections and occlusions in the image usually result in erroneous measurements that may cause the filter to diverge or the controller to fail. To reject any outlying sensor measurement, a chi-squared test is implemented [19, 40, 21]. Using the innovation of the sensor measurement the chi-squared test is:

$$\chi^2 = (z_i - \hat{z}_i) S_i^{-1} (z_i - \hat{z}_i) \tag{3.6}$$

where z_i is the sensor measurement, $\hat{z}_i$ is the predicted measurement, $S_i = HPH^T + R$ is the innovation covariance matrix for a sensor with observation matrix H, R is the measurement noise covariance matrix and P is the *a priori* covariance matrix of the Kalman Filter. If the value of χ^2 is larger than a predefined confidence level, then the measurement is rejected and the filter update is skipped.

It is usually assumed that the measurement uncertainty remains constant during the operation of the system. In our case, however, the uncertainties are expected to change significantly as the mission progresses and the surrounding environment changes. For example, GNSS localization accuracy is expected to degrade close to large structures and the tag measurement accuracy reduces with the distance between the platforms.

In order to deal with the time-varying uncertainty of measurements, an adaptive filter must be used. Covariance Matching (CM), proposed in [64], is one of the approaches used in adaptive filters. CM uses the innovation sequence v_i to make the residuals consistent with their theoretical covariances. On a sample of N measurements, the innovation covariance matrix can be approximated by its sample covariance as:

$$S_i \approx \frac{1}{N} \sum_{i=1}^{N} v_i v_i^T \tag{3.7}$$

and since $S_i = HPH^T + R$, the estimated measurement covariance matrix is approximated as:

$$R_i \approx \frac{1}{N} \sum_{i=1}^{N} v_i v_i^T - HPH^T \tag{3.8}$$

where N must be chosen empirically for each sensor to provide some statistical smoothing. Finally, it is important to note here that a nominal value for R_i should be measured in optimal conditions and Eq. (3.8) must only be used to increase R_i.

3.4 System Control

In this section the different control schemes used will be discussed. First, the state machine for the high level autonomy of the system will be explained. Then, a brief introduction to the Jetyak controls will be presented and finally the UAV flight controller will be discussed in detail.

3.4.1 Behaviors State Machine

The high level behaviors of the system are designed as a state machine, shown in Figure 3.6. Seven main behaviors are implemented: **Ride**, **Take-Off**, **Follow**, **Leave**, **Return**, **Land**, and **Hover**. Secondary behaviors are also implemented to handle specific functions such as waypoint following and spiral search. Transitioning between the states can be either automatic or manual after a user request. The behaviors and their functionality are:

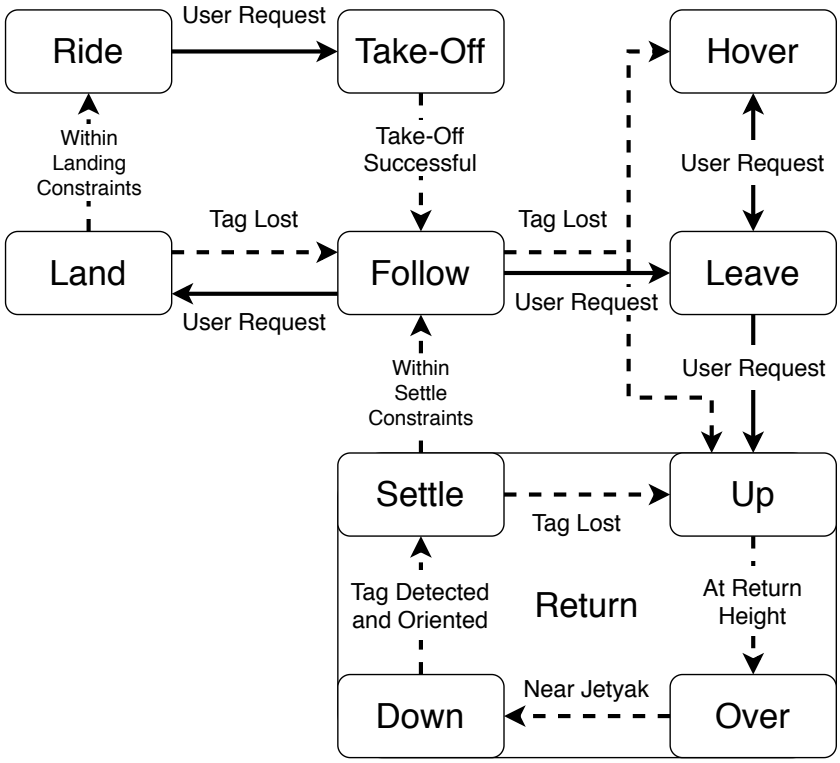


Figure 3.6: State machine diagram of the behaviors. Dashed lines signal automatic transitions with conditions. Solid lines show where user input is required for transitions. Higher level behaviors may also generate user requests.

(i) **Ride**. This is the initial state and is used while the drone is ferried by the Jetyak. In this state, the UAV is on stand-by, waiting for a request. The motors are ensured to be off to allow for a smooth ride on the Jetyak and to conserve energy. The colocalization framework is normally operating in this state to provide the ASV an augmented positioning if needed.

(ii) **Take-Off**. Every mission starts with a **Take-Off** request. Once the **Take-Off** state is requested, the drone controller starts the motors and the UAV starts ascending. Once a certain altitude is reached, the behavior state automatically switches to **Follow**. In the event of loss of the visual tether during the UAV's ascent the state will first switch to **Follow** and the safety features designed for that state will be utilized.

(iii) **Follow**. In the **Follow** state, the UAV controller aims to maintain a visual tether with the ASV while flying behind it in a safe distance. Moreover, at this position, the UAV can provide a bird's eye view of the Jetyak's surroundings as shown in Figure 3.7. Through in-lab trials using a Motion Capture (MoCap) system it was determined that the position of $2m$ above the ASV and $1.5m$ behind has the best marker reliability. In this position, the ARTag bundle at the mast and two out of the four markers in the landing pad are visible. As a safety feature, if the UAV is unable to track the ARTag bundle for more than 10 seconds, then the state is either switched to **Hover** or to **Return**, depending on the configuration.

(iv) **Leave**. This is an umbrella state that includes all the activities that the UAV completes when it does not maintain a visual tether with the Jetyak. Since there is no visual tether, the cooperative localization framework is updated using only the individual GNSS data and thus the two platform states become independent. In this state, the drone can be requested to move and operate in a remote location. Using waypoint following control, the drone can fly in patterns and gather data from areas of interest either for mapping or monitoring purposes. A spiral and a quadrilateral coverage pattern trajectory generator are included in this waypoint follow method. When a mission is completed, the state automatically switches to **Hover** where the UAV remains in stand-by until it receives another request.

(v) **Return**. When the UAV is in the **Leave** state, the **Return** state can be requested to move the drone back in a position close to the Jetyak, where a visual tether can be established. To ensure a smooth transition between the states, the **Return** state maintains an internal state machine as seen in Figure 3.6. Initially, the ***Up*** state moves the UAV to an altitude of $10m$ that should provide

a safe path for the return flight. Meanwhile, the gimbal controller moves the camera to point towards the estimated Jetyak position so that a visual tether can be established as soon as possible. Once in a safe altitude, the state switches to ***Over*** where the UAV moves toward the estimated Jetyak location. Once the UAV approaches the area over the Jetyak, the state switches to ***Down*** where the UAV slowly descents to an altitude of $3m$ above the ASV where both tag bundles should be visible. Then, the state switches to ***Settle*** and the controller corrects the UAV's orientation to match the Jetyak's heading while it moves the drone to a specified position behind the Jetyak. During the final descent and positioning, a visual tether needs to be established in order to proceed to the next steps of the **Return** state, otherwise the state is switched back to ***Up*** and the process repeats. Finally, once the UAV has descended to a safe altitude above the ASV and a visual tether is established, the state transitions to **Follow**.

(vi) **Land**. This state is used for the docking procedure since it transitions from the **Follow** to the **Ride** state. The drone controller moves the UAV from the **Follow** position to a safe position above the landing pad. The UAV starts descending when the drone is within the landing constraints: (i) the UAV velocity is within 15 cm/sec of the velocity of the Jetyak, while (ii) it is inside the volume of a pyramid, centered at the center of the landing platform. The apex of the pyramid is at 20 cm over the platform while the base of the pyramid is a square with a 80 cm side. Once the drone is resting safely on the Jetyak then the controller switches off the motors and the state switches to **Ride**. A landing can only be requested from the **Follow** position. If the UAV is in any other state, the **Return** state needs to be requested first to move the UAV to the **Follow** position. For safety reasons, if the visual tether is broken for more than 3 seconds the state is automatically switched back to **Follow**.

(vii) **Hover**. This is a stand-by state where the UAV maintains its position and altitude while it is away from the ASV. This state is used as a safety measure when the tag is lost in the **Follow** state so that the drone will hold its position and altitude until further action can be taken. Moreover, it can also be used in the case that there is a need to hold position in order to monitor a point of interest.

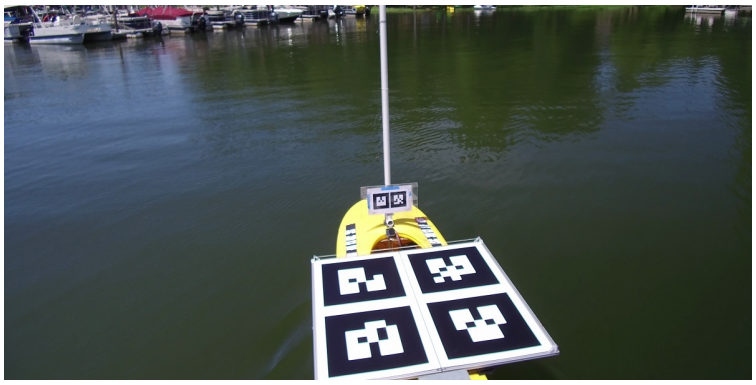


Figure 3.7: A bird's eye view image of the ASV and its surrounding environment. Captured from the drone camera while following the Jetyak.

The operational cycle of the proposed system is graphically represented in Figure 3.8, where the concept of operations and the different phases of flight are shown and explained for a sample mission.

3.4.2 Jetyak control

The Jetyak utilizes a Proportional-Integral-Derivative (PID) controller in order to follow a trajectory specified by waypoints. A controller that dynamically updates the waypoints in order to compensate for disturbances from wind and current was presented in [68]. A similar approach can be used in GNSS degraded environments where the Jetyak will correct the waypoints according to the position reported by the cooperative localization framework.

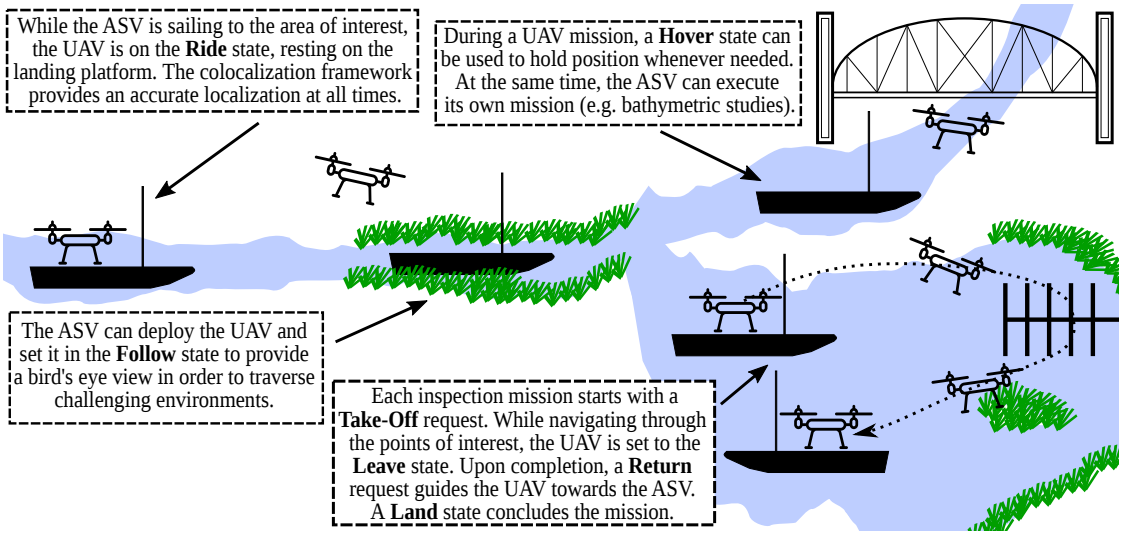


Figure 3.8: Concept design highlighting the characteristics of the UAV-ASV marsupial system. For each phase of the sample mission the relevant high level behavior state is provided.

3.4.3 UAV flight control

For such a system to operate safely and continuously, the most critical part is an effective and repeatable docking and separation procedure. In the proposed system the separation and docking procedures are autonomous taking-off and landing on a moving platform. While taking-off is a trivial process for Vertical Take-off and Landing (VTOL) vehicles, autonomous landing is considered one of the most challenging parts of flight, and it requires precise sensing and accurate control [12, 33, 62].

To control the position of the UAV during all mission stages, a Linear Quadratic Regulator (LQR) was designed. In order to evaluate the performance of our controller, a PID controller was also implemented to be used as a baseline. To make the comparison fair, a significant amount of time was invested in tuning both controllers while both of them were tested in the field and were successful on landing the UAV on the Jetyak.

The drone allows for different types of control inputs. On the horizontal axis, the commands can either be velocities, roll and pitch angles or roll and pitch rates. On

the vertical axis, the control input is the velocity along the axis and for rotations around the vertical axis, the command is the yaw rate.

The tuning of the controllers started on the hardware-in-the-loop (HIL) simulator provided by DJI and then gradually moved to indoor flights using a MoCap system to finally reach field trials where they successfully managed to land the UAV on the platform. The two controllers are described in the following subsections while the results from their comparison are presented in Section 3.5.

LQR Controller

To implement the LQR, a model of the system under control is needed. A high-level position controller was designed on top of the built-in attitude controller. Nevertheless, a model for the closed loop attitude dynamics needs to be derived. Data from the onboard sensors along with the corresponding RC commands were used to derive second order models of the roll and pitch dynamics. The State-Variable Filters (SVF) approach along with the Generalized Poisson Moment Functions (GPMF) approach were used to estimate the model parameters [31]. The structure used for the model is:

$$\begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -b_{\dot{q},q} & -b_{\dot{q},\dot{q}} \end{bmatrix} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} + \begin{bmatrix} 0 \\ c_{\dot{q},q_r} \end{bmatrix} q_r \tag{3.9}$$

with q representing either the roll (ϕ) or the pitch (θ) angle and q_r the reference signal. By linearizing the system around the stable state of hovering, the horizontal position dynamics are modeled as second order system. Around the hovering point, the only force acting on the drone is the gravity. Assuming that there is some damping relative to the velocity and using the small angle approximation we have:

$$\begin{bmatrix} \dot{p} \\ \ddot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\lambda_p \end{bmatrix} \begin{bmatrix} p \\ \dot{p} \end{bmatrix} + \begin{bmatrix} 0 \\ \pm g \end{bmatrix} q \tag{3.10}$$

where p represents either the x axis ($+g$, $q = \theta$) or y axis ($-g$, $q = \phi$). The only unknown in the model is the parameter λ_p that captures the damping of the system.

Table 3.1: Model parameters found through model identification.

Parameter	Roll (ϕ)	Pitch (θ)
$b_{\dot{q},q}(sec^{-2})$	45	46
$b_{\dot{q},\dot{q}}(sec^{-1})$	7.2	7.4
$c_{\dot{q},q^r}(sec^{-2})$	42.5	44
$\lambda_p(sec^{-1})$	0.05	0.05

To estimate this parameter, data from the MoCap system were used in a non-linear least-squares optimization algorithm.

The parameters for both roll and pitch angles are shown in Table 3.1. As for the altitude and yaw control, a second order model for the vertical position z and heading ψ was derived using the vertical velocity u_z and yaw rate u_ψ as inputs:

$$\begin{bmatrix} \dot{\psi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -b_{\dot{\psi},\dot{\psi}} \end{bmatrix} \begin{bmatrix} \psi \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 \\ c_{\dot{\psi},\psi^r} \end{bmatrix} u_\psi \tag{3.11}$$

$$\begin{bmatrix} \dot{z} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -b_{\dot{z},\dot{z}} \end{bmatrix} \begin{bmatrix} z \\ \dot{z} \end{bmatrix} + \begin{bmatrix} 0 \\ c_{\dot{z},z^r} \end{bmatrix} u_z \tag{3.12}$$

PID Controller

For the PID case, the selected control inputs were the velocities in the horizontal and vertical axes and the yaw rate. The input to the controller is the position error on the three axes as well as the desired orientation of the drone. Since the control inputs are in the body frame, the position error must be also expressed in the same frame. Thus, the orientation of the drone is used to rotate the body frame to the world frame before calculating the position error.

As discussed in Section 3.4.1, there are three different states where the drone needs to be in a controlled flight, **Follow**, **Land** and **Leave**. Each of these states requires a different control approach. For example, in the **Follow** state, the drone needs to be able to keep up with a moving boat so it needs to be more aggressive

while in the **Return** state a smooth flight is more preferable because it will produce better quality surveys and inspections. For that reason, a gain scheduling approach was followed with three different sets of gains for the three different states. This is however not the case for the LQR controller, since the LQR was able to handle all cases with the same tuning.

3.5 Experimental Design and Results

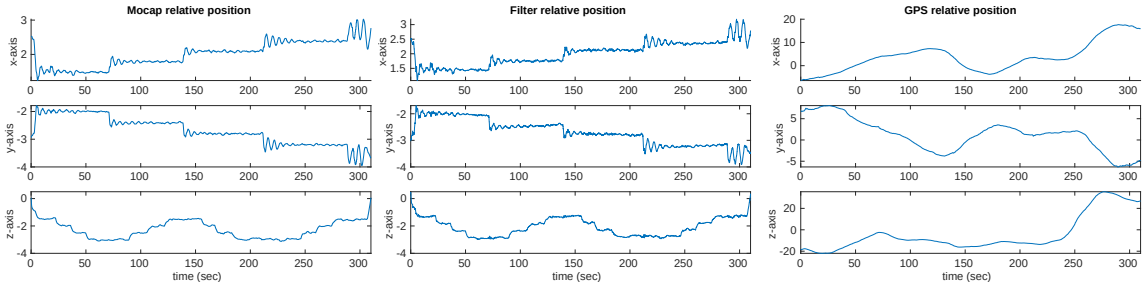
The proposed system and its subsystems were thoroughly tested. In this section, the experimental setup for each test will be discussed and the results will be presented. First, the cooperative localization and controller comparison experiments in the lab and at the field will be presented. Then, a number of field experiments will be discussed that demonstrate the system's effectiveness for surveying and inspection.

3.5.1 Cooperative localization experiments and results

The cooperative localization framework was tested in the lab using an OptiTrack MoCap system to provide an accurate ground truth. Moreover, since the experiment was done indoors, we were able to emulate a GNSS degraded environment. The aim of this experiment was to test the accuracy of the framework in finding the relative position between the two platforms in a GNSS degraded environment.

For this experiment, the Jetyak was static and the drone was flying in different positions while maintaining a visual tether. Specifically, using the drone position from the MoCap system, the drone was flying to a number of pre-specified positions of different altitudes and distances from the tag to see how the distance may affect the tag measurements and the whole cooperative localization framework. The testing range for the altitude was from 1.5m to 3.5m and for the horizontal distance was from 2.5m to 4.5m. During the experiment, the standard deviation for the Jetyak position

reported by the GNSS was 5.66m in the x-axis, 3.11m in the y-axis, and 17.94m in the z-axis.



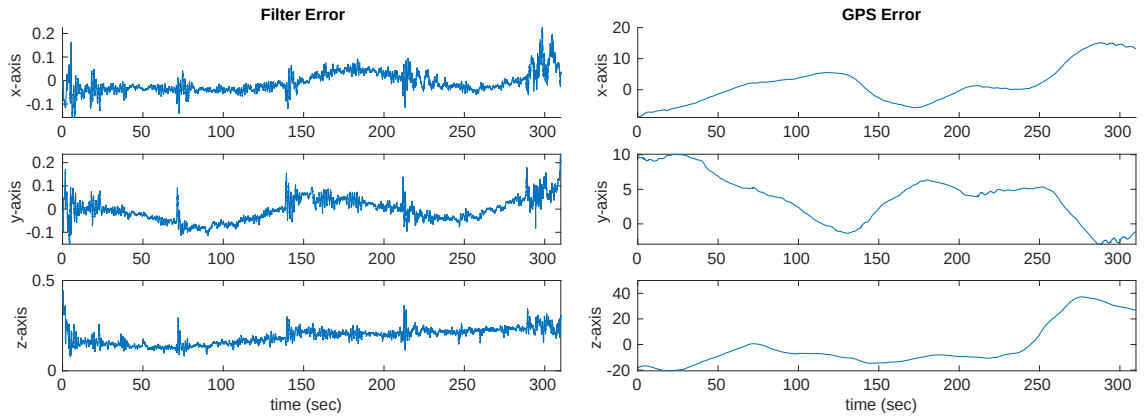
(a) (b) (c)

Figure 3.9: Relative position of the two platforms in a GNSS degraded environment according to (a) the OptiTrack Motion Capture system, (b) the cooperative localization framework, and (c) using only the GNSS.

Figure 3.9 shows the relative position of the two platforms according to the MoCap system, the proposed filter, and using just the GNSS measurements. It is clear that in such an environment, using only the GNSS position would not be able to provide an accurate enough relative position for an autonomous landing. On the other hand, the filter results have a significantly higher accuracy. Moreover, as it visible in Figure 3.10, the error does not significantly change with the distance at least at the range that it was tested.

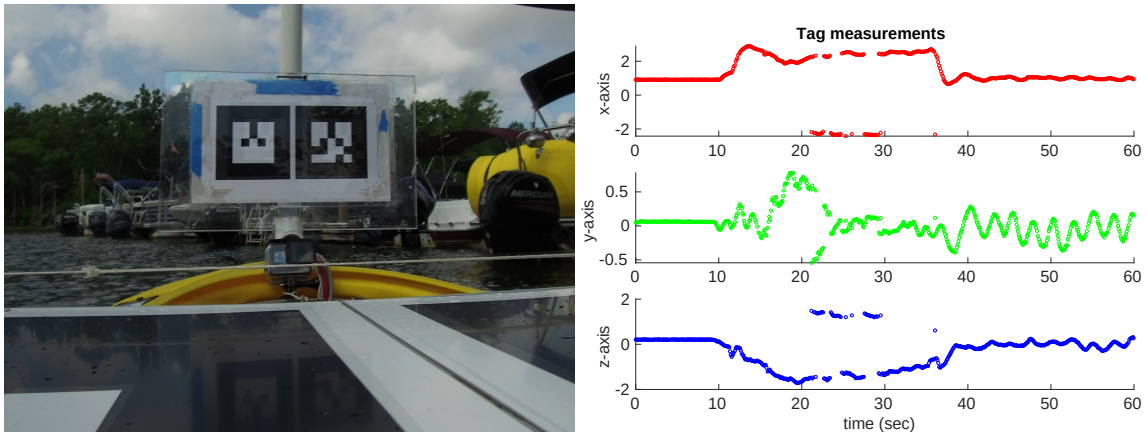
While the GNSS positioning usually has a slow drift, the tag measurements are more prone to mis-detections that could cause aggressive changes in the relative position of the two platforms. Such aggressive changes in the relative position, especially during the approach and landing maneuvers, may lead to the drone crashing. Reflections are a common source of mis-detections especially in the freshwater domain. Figure 3.11 shows the mast tag being reflected on the platform during the approach maneuver and the results on the tag detection measurements.

Using the proposed outlier rejection method, such measurements do not affect the filter, ensuring the safe operation of the system. Figure 3.12 shows the cooperative localization framework with and without the outlier rejection method. As it can



(a) (b)

Figure 3.10: The position error according to (a) the filter results and (b) the GNSS results.



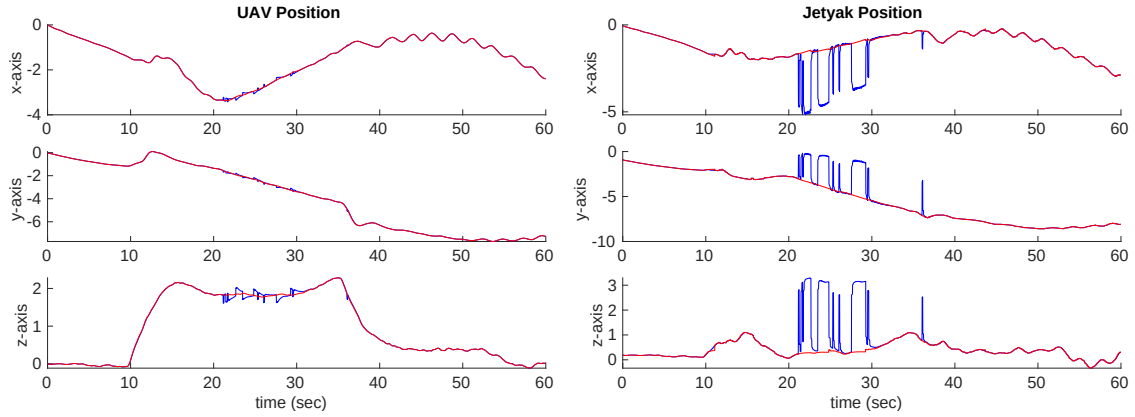
(a) (b)

Figure 3.11: (a) Reflections of the mast tag on the platform is a cause of mis-detections; (b) While the drone tries to approach the platform and land, the reflections on the platform may cause outliers in the tag measurements as seen around the 20 and 30 second mark of the experiment.

be seen in the figure, the positions of both platforms are affected, with the Jetyak position being affected the most with an error of a few meters. The reason why the UAV position is more robust to outliers is that the UAV GNSS measurements are more certain than that of the Jetyak.

As an additional test, in one of the field experiments we purposefully took the drone camera out of focus, as seen in Figure 3.13, to test the cooperative localization framework and also the designed fail-safe capabilities of the high-level controller. The

focus of the camera was fixed so that it could clearly see the markers while close to the landing platform so that the landing procedure would not be compromised. On the other hand, in the follow position, the markers are blurry and the vision based localization was not able to track the tags for extended periods of time while it was producing less accurate data. While the drone was initially able to follow the Jetyak, when 10 seconds passed without a new tag measurement then, as described in Section 3.4.1, the drone initially switches to **Hover** before it switches to **Return**. For the **Return** behavior, the drone controller uses the platform positions estimated from the localization framework. Thus, if the framework was not able to discard outliers and adapt to any erroneous measurements provided by the camera, the system would not be able to continue its operation. However, the system was able to manage this challenging situation, switch between the different behavior states and finally land safely on the platform.

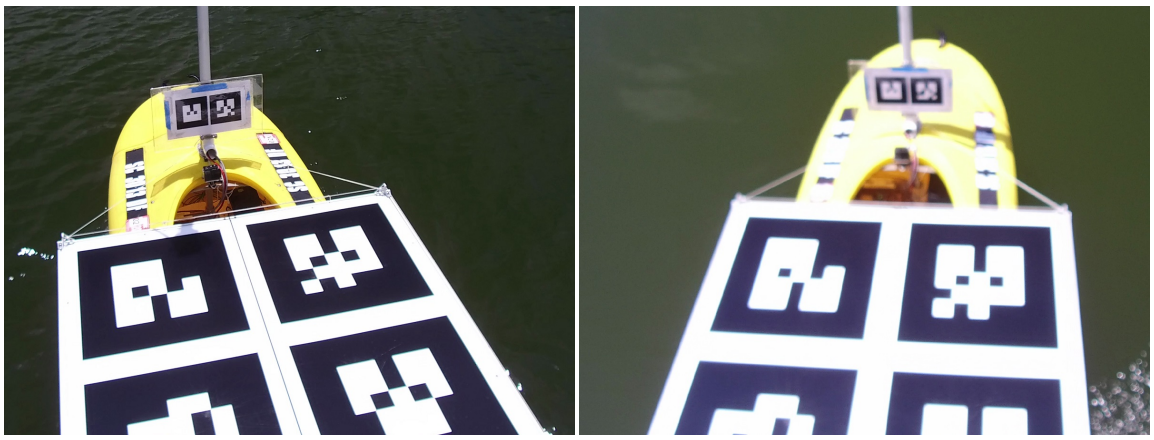


(a) (b)

Figure 3.12: (a) The estimated UAV position and (b) the estimated Jetyak position; blue line is without outlier rejection and red line is with outlier detection.

3.5.2 Controller experiments and results

As described in Section 3.4, both of the developed controllers were tested in the field to make sure that they were able to successfully land the drone on a moving platform.

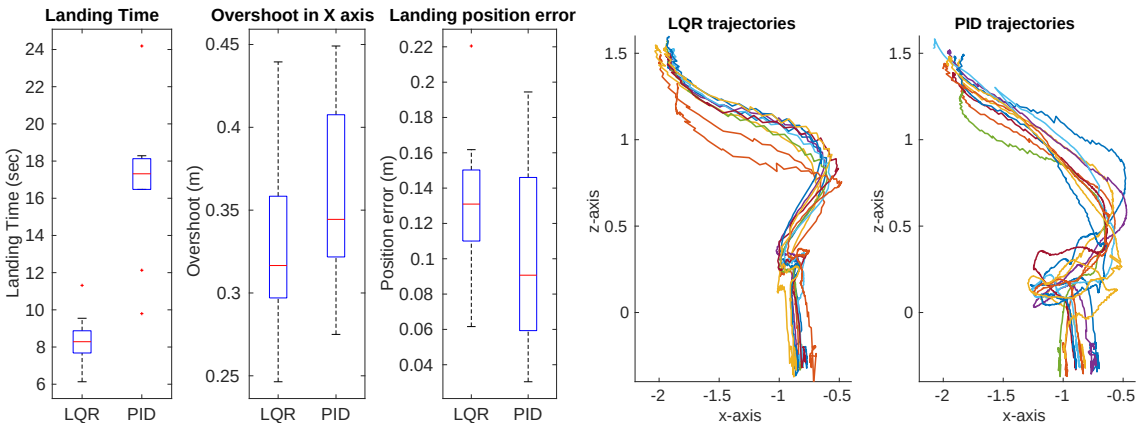


(a) (b)

Figure 3.13: A cropped view of the Jetyak from the drone with the camera (a) in focus and (b) out of focus.

Initially, both controllers were tuned using a HIL simulator provided by DJI, then in the lab using a MoCap system and finally in the field. Since the most challenging aspect of any mission is the landing, the controllers were judged by their effectiveness on this task. While the controllers were tuned to land the drone on a moving platform, a static landing was used for the comparison. The reason for this is that in a static landing there are less uncontrolled variables that might affect the results. In the field, the wind and current conditions might change during the comparison affecting the results and leading to wrong conclusions. Furthermore, in the lab there is not enough space to perform a landing in a moving platform.

The three metrics used to compare the controllers are the time it took to land the drone from the moment of the request until the drone was resting on the platform with the motors off, the maximum overshoot and finally the error on the landing position. The missions that the system undertakes are time critical, meaning that the drone should use its limited flight time effectively. Moreover, in cases where a mission needs to be abruptly terminated, the drone needs to return and dock on the boat as fast as possible. As described in Section 3.2, the Jetyak is equipped with a mast that holds the communication hardware and is used to host other mission specific sensors such



(a) (b)

Figure 3.14: (a) Box and Whiskers plots showing the landing time, overshoot and final error for the 10 field landings; (b) The different trajectories from the follow position to the landing position.

as an anemometer or a lidar. As a result, the maximum overshoot in the landing process has an increased significance in order to minimize the chance of the drone crashing on the mast. Finally, the position error shows how close to the center of the platform is the drone when it lands. It is worth noting here that the drone is trying to land in an area around the center of the platform which is selected to be a $15cm \times 15cm$ square.

As mentioned earlier, the controllers were compared in all three domains: simulation, lab and field trials. In all three domains, the drone starts on the platform, takes-off and flies to the follow position that is 1.1 meters back and 1.6 meters up and after 20 seconds it starts the landing procedure. For the simulation results, a total number of 30 landings were executed for each controller in 6 different wind conditions. Initially the wind is set to zero and then it is increased to $3m/s$ and $4m/s$ in both the lateral and longitudinal directions. In the last case, the wind is set to $3m/s$ in both directions. The choice of the selected wind speeds was made because it was observed that the controllers were unable to land the drone with wind speeds of $5m/s$ or more in the longitudinal direction. For the lab experiment, the MoCap system was used to provide the position of the platform and the drone during five consecutive landings.

Table 3.2: Landing results for the three different test types showing the mean values along the experiments for all three metrics.

Controller	Experiment	Landing time(sec)	Overshoot(m)	Landing Error(m)
LQR	Simulation	8.373	0.103	0.146
	MoCap	9.576	0.344	0.088
	Outdoors	8.382	0.331	0.133
PID	Simulation	17.032	0.281	0.020
	MoCap	15.406	0.552	0.128
	Outdoors	16.833	0.355	0.101

Finally, two sets of five consecutive landings were done at the field, where the position of the two platforms was provided by the cooperative localization framework.

Table 3.2 shows the results for both controllers in all three domains. In all cases, the LQR controller was able to land the drone significantly faster than the PID controller. Moreover, the overshoot is lower for the LQR in all cases. The PID controller shows smaller errors on the landing position in most cases but both controllers were able to land inside the predefined area. Figure 3.14 shows the box and whiskers plots for the 10 field landings along with the trajectories followed in the indoor landings. One can notice that the LQR had a more consistent behavior between the different trials. Finally, it is worth mentioning that the measured overshoot on the static landings was higher than the one observed on landings in a moving platform especially for the PID. As mentioned before, the two controllers were tuned to land on a moving platform and as a result they had a more aggressive response.

Switching between the **Take-Off** to **Follow** and finally to **Land** states was thoroughly tested on the field with a moving platform. The first field deployments aimed to test the separation and docking mechanism of the system and its safety. Moreover, to test the robustness of the system, the **Take-off**, **Follow** and **Land** sequence was repeated in different weather conditions. The system was able to complete the sequence even when flying under significant side winds and with the boat swaying due to waves, as shown in Figure 3.15. Finally, the sequence was also repeated con-

secutive times to show that the landing position is a viable position for the drone to take-off again and repeat the process. The position of the two platforms and the control effort of the drone during one of the field landings is shown in Figure 3.16. In this particular experiment the drone was able to land on the boat in $14sec$ while the ASV was moving with a velocity of around $2km/h$.

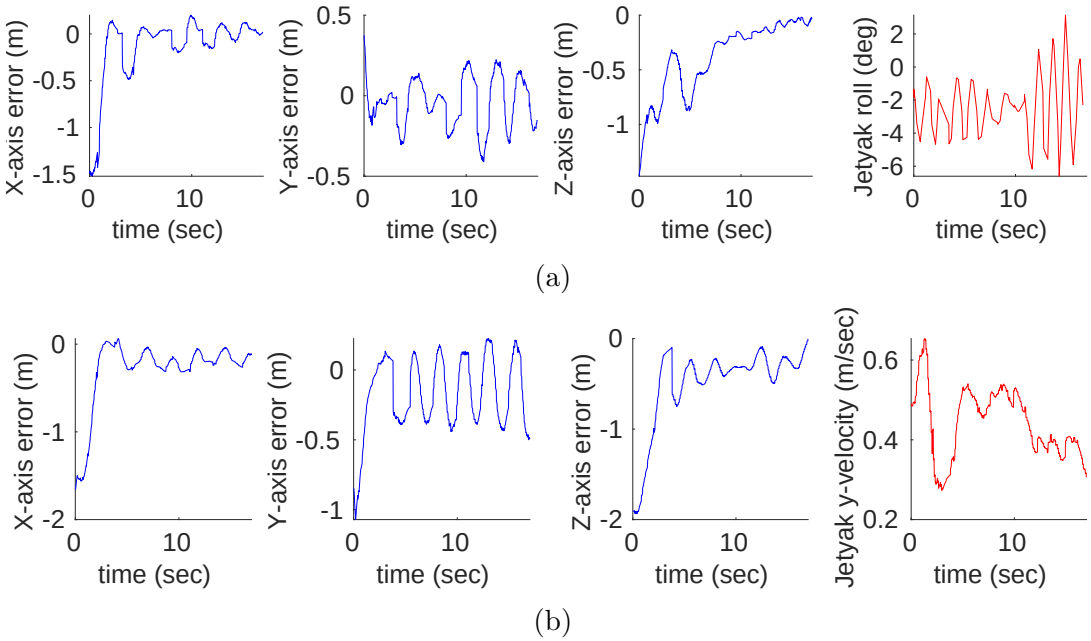


Figure 3.15: Landing in challenging conditions with the LQR controller. (a) Landing under sway. Waves in the water surface can significantly affect the Jetyak's attitude. The figure shows the errors in the x, y and z axes along with the roll angle of the Jetyak during landing; (b) Landing under side wind. The presence of significant side winds is affecting both platforms by introducing a velocity in the y-axis. The figure shows the errors in the x, y and z axes along with the y-axis velocity of the Jetyak in its body frame.

3.5.3 Surveying and Inspection Results

The system was tested on a number of successful field deployments in two different freshwater environments, Lake Murray and the Congaree River, both located in South Carolina, USA. During these deployments, the capabilities of the system were tested and its inspection effectiveness evaluated. A successful deployment is one where the drone starts on the Jetyak, flies to the region of interest, completes its mission, then

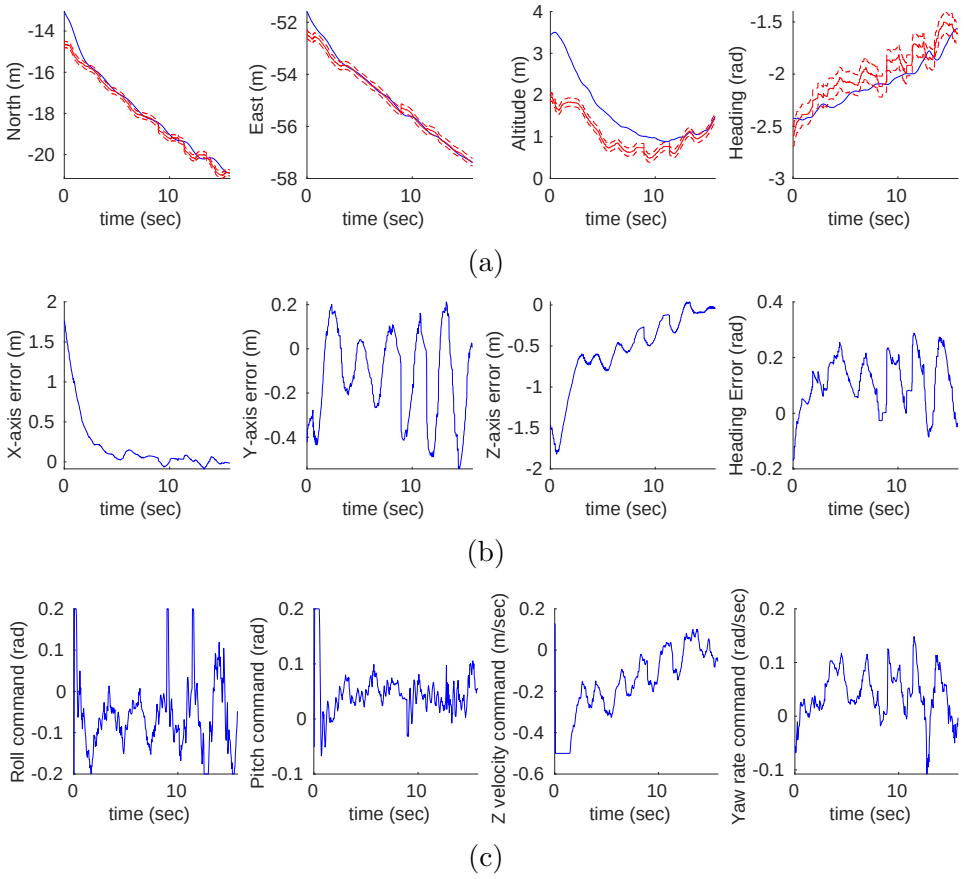


Figure 3.16: (a) The position of the Jetyak and the UAV during a field landing in a local ENU system. The blue line is the UAV position and the red solid line is the Jetyak position. The red dashed lines show the landing constraints; (b) The landing error on the UAV's body frame. Notice that the heading affects the X and Y axes errors; (c) The control effort commands of the drone during landing using the LQR controller.

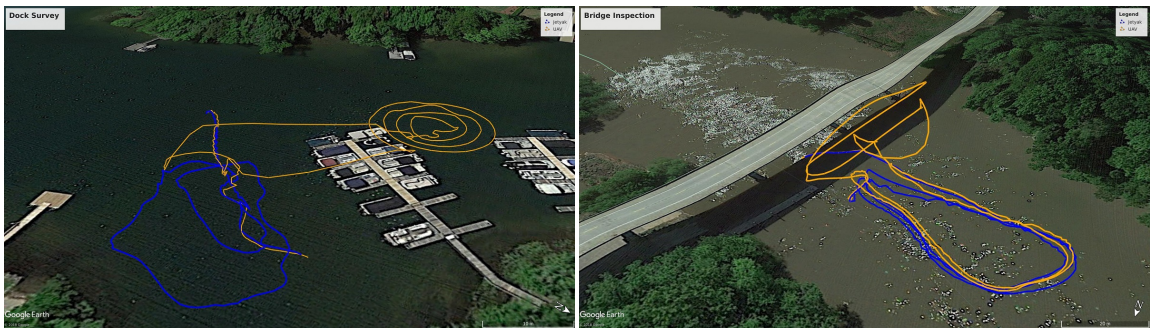
returns to the Jetyak, and performs a landing. While both controllers were used in the field deployments, only a small number of them were completed using the PID controller and the vast majority was done using the LQR controller.

To show the capability of the UAV to provide a bird's-eye-view perspective to the boat, a series of deployments were done with the drone following the ASV while the boat was following a trajectory autonomously using way-point navigation. In these tests, the Jetyak would start following a desired trajectory and at some point it would deploy the drone in order to get the augmented field of view. The drone would follow

behind the ASV at a safe distance while tracking the position of the Jetyak in a local world frame. When the ASV had finished its trajectory it would signal the UAV to start the landing procedure.

The **Leave** mode is designed to control the UAV when it operates remotely. As described in Section 3.4.1, the system can use predefined waypoints in order to cover an area of interest and gather data. In our deployments, three different types of missions were completed. First, the drone would move from the **Follow** position to a single waypoint where the mode would switch to **Hover** in order to fly in position and wait for further instructions. This type of mission is useful when information about a specific region needs to be gathered for a duration of time. As an example, in the case that the Jetyak needs to traverse a GPS degraded environment, the drone would fly in a position with good GPS coverage and stay there while tracking the Jetyak's position. In the second type, the drone would move to a waypoint and then a spiral pattern generator would produce a number of waypoints to create a spiral trajectory of the desired radius and number of rotations. This type of experiment is used in surveying missions to map an area of interest. Finally, using three different waypoints, a quadrilateral trajectory can be generated with the desired number of sweeps. This type of mission can either be used for surveying and mapping or for the inspection of a structure. The trajectories of a spiral trajectory mission and a quadrilateral trajectory mission, can be seen in Figure 3.17.

The primary application of the system is to automate surveys of near-shore environments and inspections of structures. In Lake Murray, a dock and its surrounding area were surveyed 6 months apart to show the change in the surrounding area over that period. Figure 3.17(a) shows the mission trajectory for the summer deployment while Figure 3.18 shows the high definition maps created from the survey data using the automatic stitching method described in [13]. One can see the change in the water level, the difference in the vegetation as well as the boards of the dock which



(a) (b)

Figure 3.17: The trajectories of the UAV (orange) and ASV (blue) during (a) a spiral survey of the docks and (b) a bridge inspection.

(a) (b)

Figure 3.18: Mosaic created from spiral survey of a dock and shore in (a) winter and (b) summer. The change in the water level is noticeable as well as the blooming of the near shore vegetation. The spiral trajectory that resulted in the mosaic in (b) can be seen in Figure 3.17 (a).

have been replaced in the structure.

To demonstrate the inspection capabilities of the system, a bridge at Congaree River was selected for inspection. There, the system was deployed multiple times to take videos of the structure and the surrounding area from different angles with Figure 3.17(b), showing one of such deployments. Using COLMAP [83, 84], a Structure-from-Motion software, a dense 3D reconstruction of the structure was generated as shown in Figure 3.19. The bridge under inspection, the surrounding environment

and the nearby transmission towers can all be seen in the reconstruction. This reconstruction can be used to assess the state of the bridge, measure the water level and monitor the surrounding environment.

Figure 3.19: Dense rendering of the inspected bridge and its surrounding environment at 33.753000N, 80.645223W. The trajectory of the mission that resulted in this rendering is shown in Figure 3.17 (b).

Chapter 4

Experimental Comparison of Fiducial Markers for Pose Estimation

4.1 Introduction

Autonomous systems need a reliable localization framework, whether it is for navigation [57] or object manipulation [37]. Autonomous navigation applications often rely on Global Navigation Satellite Systems (GNSS) like GPS or Galileo, however, recent developments in Visual Inertial Odometry (VIO) [22, 82, 29] and Simultaneous Localization and Mapping (SLAM) [30, 15] have made the deployment of autonomous platforms in GNSS degraded and denied environments possible as well. VIO and SLAM methods are able to provide accurate localization in structured environments with strong visual features but often fail in environments with poor visual features or challenging lighting conditions [18, 46]. In such cases, fiducial markers are often used to augment the localization capabilities of the platform.

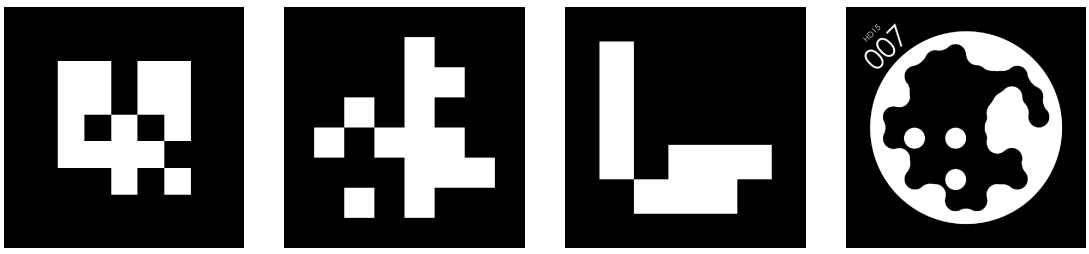
Fiducial markers offer a highly distinguishable pattern with strong visual characteristics and in most cases they also feature a specific encoding as a fail-safe against misdetections. As such, they are a commonly used framework in object manipulation application tasks where markers are used for the identification, detection and localization of different objects [100]. There are many different fiducial marker packages available, including some that are offered as open-source. Many of these open-source packages are also offered to the robotics community as ROS packages.

While there are previous works that provide comparisons between different pack-

ages, most of them focus on the detection rate under different conditions. For example, authors in [87] provide a comparison between three different packages (ARTag, AprilTag, and CALTag) in the presence of moderate degrees of rotation and with partial occlusions with a focus on the detection rate. Authors in [81] provide a comparison of three marker libraries (ARToolKit, AprilTag and ArUco) on their underwater detection performance and required processing time in different lighting conditions, water turbidity levels, marker sizes, distances, and rotations. Finally, in [6], the authors use two different cameras, a high-resolution webcam and a low-resolution High Dynamic Range (HDR) camera with the ArUco marker library in order to explore the effect of the camera sensor to the pose results.

In this work, the experimental comparison of the four fiducial marker packages (ARTag, AprilTag, ArUco, and STag) is presented with a focus on localization accuracy and robustness. Different marker configurations and their effect on the localization results are also considered including non-planar bundles and bundles of differently sized markers. Moreover, the effect of different lighting conditions and motion blurring is also investigated. Finally, the computational performance for each marker is evaluated in three different computing systems commonly used in autonomous platforms. The work presented here was published in [50, 51].

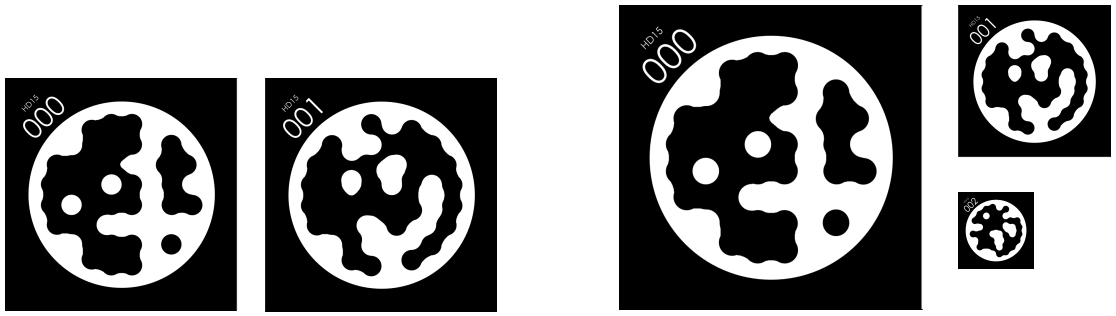
4.2 Experimental Setup



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

Figure 4.1: The markers used in this work: (a) ARTag [94], (b) AprilTag [71, 95], (c) ArUco [32], and (d) STag [10].

The most commonly used fiducials are the ones with a square shape. As such, the four fiducial markers chosen to be evaluated are ARTag [27, 28], AprilTag [71, 95], ArUco [32], and STag [10]. They are all square, monochromatic markers as shown in Figure 4.1. The selection of these particular markers is based on the fact that they are: (i) state-of-the-art markers, (ii) widely used in the UAS literature and (iii) open-source or offer a ROS implementation. ARTag, AprilTag, and ArUco are widely used in the robotics community; STag, however, is a new marker package which focuses on the stability of the generated results, an important factor in robust and precise localization.



(a) (b)

Figure 4.2: The STag multi-tag bundles (a) with two equal sized markers and (b) with three differently sized markers.

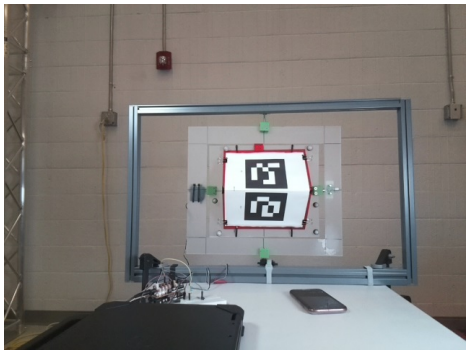


Figure 4.3: The two marker bundle in the non-planar configuration. The marker is rotated about the vertical axis.

This paper investigates multiple configurations of the previously noted packages including singular markers (Figure 4.1), planar multi-marker bundles (Figure 4.2),

and a non-planar two marker bundle (Figure 4.3). As shown in many of the previous works, it is common to group multiple markers in a single bundle in an effort to increase the accuracy or robustness of the localization framework. Bundles are considered to make the system more robust to failure than systems reliant on a single marker so in this paper we investigate this claim through experimentation.

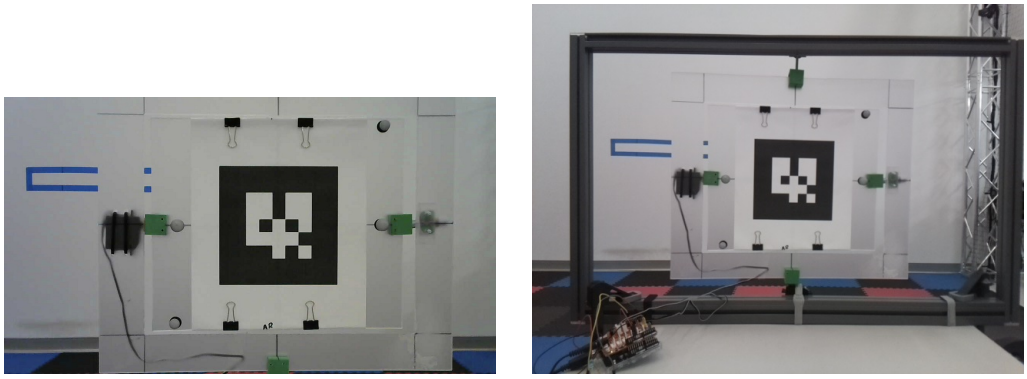
The size of the markers differs between the different configurations, specifically, the single marker configurations contain a 14.6cm square tag. The two-marker bundle features two 10.8cm square tags and the three-marker bundle includes a 10.8cm, a 5.4cm, and a 2.7cm square tags. The non-planar bundle is composed of the two-marker bundle bent at a 90° angle convexly and attached to a rigid platform.

Table 4.1: Computer system specifications.

	Pi 3B+	Jetson TX2	Intel NUC
RAM	1GB	8GB	16GB
RAM Type	LPDDR2 SDRAM	LPDDR4	DDR3L
CPU Cores	4	2 + 4	2
CPU Clock	1.4GHz	2GHz + 2GHz	3.1 GHz
GPU Cores	None	256	48
GPU Clock	N/A	1.3GHz	0.3GHz

This comparison includes an evaluation of the efficiency of the four packages on three different small form-factor computers that are commonly used onboard autonomous platforms. The computers used are the Raspberry Pi 3B+, the NVidia Jetson TX2, and the Intel NUC (NUC5i7RYH). The Raspberry Pi is a popular low-cost solution, while the Jetson TX2 and the Intel NUC are selected for their powerful GPU and CPU, respectively. The specifications for each computer system are shown in Table 4.1.

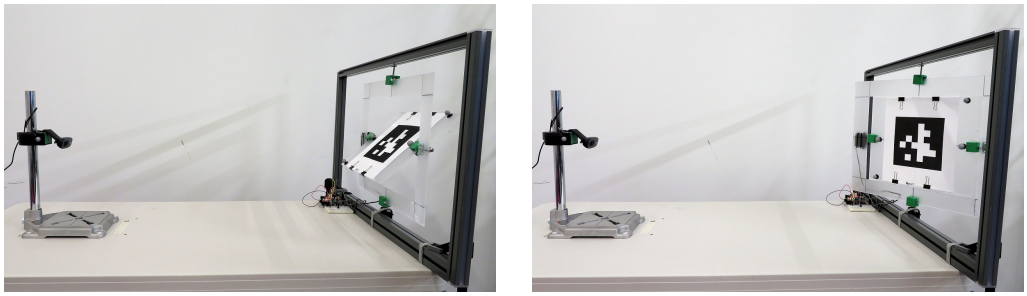
As for camera sensors, a Logitech C270 Webcam and a Raspberry Pi camera version 2 module (piCam) are used. The Logitech is used to capture a high resolution



(a) (b)

Figure 4.4: Images from (a) the Logitech camera and (b) the Raspberry Pi camera at a distance of 0.75m from the marker.

video of 1280×720 pixel^2 at $25fps$ while the piCam is used to collect data at a resolution of 640×480 pixel^2 at $20fps$. An image from both cameras is presented in Figure 4.4 at a distance of 0.75m from the marker. This showcases the larger field of view of the piCam and improved contrast and white balance over the Logitech camera.



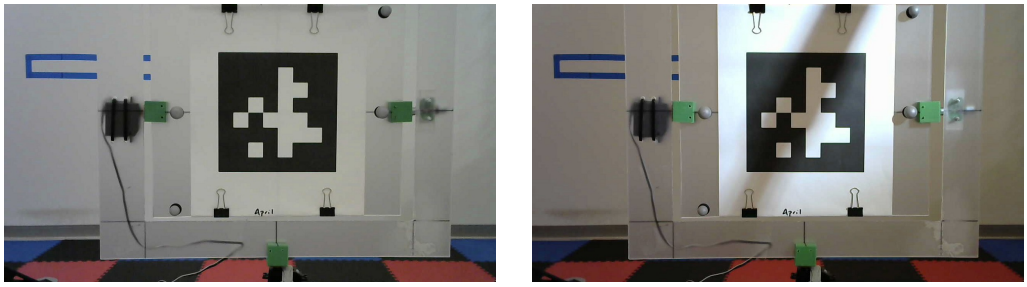
(a) (b)

Figure 4.5: Overview of the experimental setup with the two axes gimbal on the right side of each image, where the marker position is altered in the (a) longitudinal direction and (b) lateral direction.

For the purposes of this work an actuated, two axis gimbal is built to hold the markers and change their orientation as needed. Figure 4.5 shows an overview of the experimental setup. The marker base can be actuated in two perpendicular axes (vertical and horizontal). The cameras are mounted on a stand with adjustable height that can be moved to different distances from the marker base. To ensure that an

accurate ground truth is available even in cases where there are small changes in the setup between different experiments, both the camera stand and the marker base are tracked using an OptiTrack Motion Capture (MoCap) system. The image streamed from the cameras, along with the pose data from the MoCap are saved into ROS bag files so that the same data can be processed on all computers.

4.2.1 Single Marker Experiments



(a) (b)

Figure 4.6: The AprilTag marker used (a) under normal lighting conditions and (b) with a shadow being cast as seen from the Logitech camera.

To investigate the detection rate and accuracy of the package of each marker, data are collected at different distances and angles. The distances considered are $0.75m$, $1.0m$, $1.5m$, $1.75m$, and $2.0m$ at an angle perpendicular to the camera. Two different lighting conditions are considered (Figure 4.6) to determine the effect on the detection rate and the accuracy of each package. A 60 second video is captured for each package and each distance outputting a total of 1500 and 1200 images for the high and low resolution camera, respectively.

To test the performance at various angles, the gimbal rotates in 10° increments up to 80°, along the longitudinal axis, with the camera placed $1.0m$ away from the marker. The gimbal maintains the orientation of the marker for 15 seconds for each increment, resulting in 375 high resolution images and 300 low resolution images per angle. This test is only performed in the longitudinal direction because the lateral direction is expected to perform in the same manner. For these experiments two

different lighting conditions are used, as in the distance tests described earlier. This test is used to determine the impact of the angle of the camera on the orientation and position measurement accuracy as well as the detection rate.

(a) (b)

Figure 4.7: View of the ARTag marker in the translation motion blur experiments with (a) still camera and (b) camera moving at 300 degrees/sec.

The next parameter considered is resilience to motion blur. The two types of motion blur considered in this paper are rotation and translation blur.

Rotation blur is the effect created when the marker rotates quickly without translation movement in the camera frame. Rotation blur is induced by rotating the camera about its principal axis at a constant angular speed using a servo. The servo is rotated at the speeds: 60, 90, 135, 180, 240, and 300 $degrees/sec$, for 15 seconds each over a 180° range.

Translation blur is induced when an object quickly traverses the camera's frame. Translation blur is induced by rotating the camera perpendicular to its principal axis. The camera is rotated in a 50° window at the speeds: 30, 60, 90, 135, 180, 240, and 300 $degrees/sec$, for $15seconds$ each. The effect of motion blur on the image is visible in Figure 4.7.

The final parameter assessed is the computational efficiency. Using the apparatus shown in Figure 4.5, a series of continuous angle sweeps are completed along the two axes perpendicular to the camera. In the first motion, the longitudinal axis oscillates between ±45° while the lateral axis is stationary. Then the lateral axis

oscillates between $\pm 45°$. In the final motion, both axes are varied between $\pm 30°$ simultaneously to provide a variety of intermediate angles. The total duration of the sweep motion is $50 seconds$ and the experiment is repeated six times for each package.

4.2.2 Marker Bundle Experiments

To investigate any difference in the detection rates and computational cost results, the single marker experiments are repeated for the two-marker bundles. The only experiment not repeated is the motion blur experiment. The planar two-marker bundle represents the most common configuration so this will allow to make a direct comparison between single marker and multi-marker tracking.

To test whether multi-sized markers help at different distances, the distance experiment is repeated and data are collected at various different stationary distances. The distances considered are $0.5m$, $0.75m$, $1.0m$, $1.5m$, $1.75m$, and $2.0m$, at an angle perpendicular to the camera. The images for this test are collected using the camera with the best results with no shadowing included. This test aims to show whether multi-sized markers have a significant effect in the position accuracy and detection rates.

A common issue with fiducial marker pose estimation, especially in low resolution cameras, is the reduced accuracy in the orientation measurements when the marker is flat in the camera frame. The angle experiments are again conducted with the non-planar bundle to determine its impact on the orientation measurements. The angled bundles are rotated with the axis of rotation perpendicular to the axis about which the bundle is folded. The angle is increased in $10°$ steps, from $-60°$ to $60°$, with 15 second intervals to allow for an adequate number of samples to be taken.

4.2.3 Package Configurations

To process the experimental data, the ROS packages for ARTag[1], AprilTag[2] and ArUco[3] are used on all three computers with the same configuration for each camera. Our group has developed the ROS package for STag[4] and released it to the community.

Alvar [94] is a framework that builds on ARToolkit and ARTag but provides a large amount of flexibility in the style and content of its markers. Generally, a specific style that provides 32 bits of configurable space along with an additional four bits that provide heuristic information is used. This framework is originally part of a broader AR application, before being adapted for use with ROS as its initially de-facto fiducial marker framework. Finally, the original ArUco ROS package does not support marker bundles so we added bundle support using OpenCV's Perspective-n-Point algorithm (solvePnP). AprilTag and STag both use solvePnP for marker bundles while Alvar uses bundle adjustment.

4.3 Results

In this section we present and discuss the results from our experimental study. All of the experimental results are presented in Tables and selected results are presented in graphs to make specific comparisons clearer. In the Tables, the mean and standard deviation (std) columns are measured in centimeters for distances and in degrees for angles while the rate is measured in percentages. The package with the greatest detection rate is denoted in green, while the lowest error is noted in blue, and the lowest standard deviation is noted in magenta. The values marked with "ND", indicate that the tag was not detected in any of the frames and the values marked with "—" in-

[1] `https://github.com/ros-perception/ar_track_alvar`

[2] `https://github.com/AprilRobotics/apriltag_ros`

[3] `https://github.com/pal-robotics/aruco_ros`

[4] `https://github.com/usrl-uofsc/stag_ros`

dicate that the values were not calculated due to the presence of extreme outliers. Finally, when different lighting conditions are considered, normal light conditions are marked with an “N” and noise from shadows is marked with an “S”

4.3.1 Performance with varying marker distance

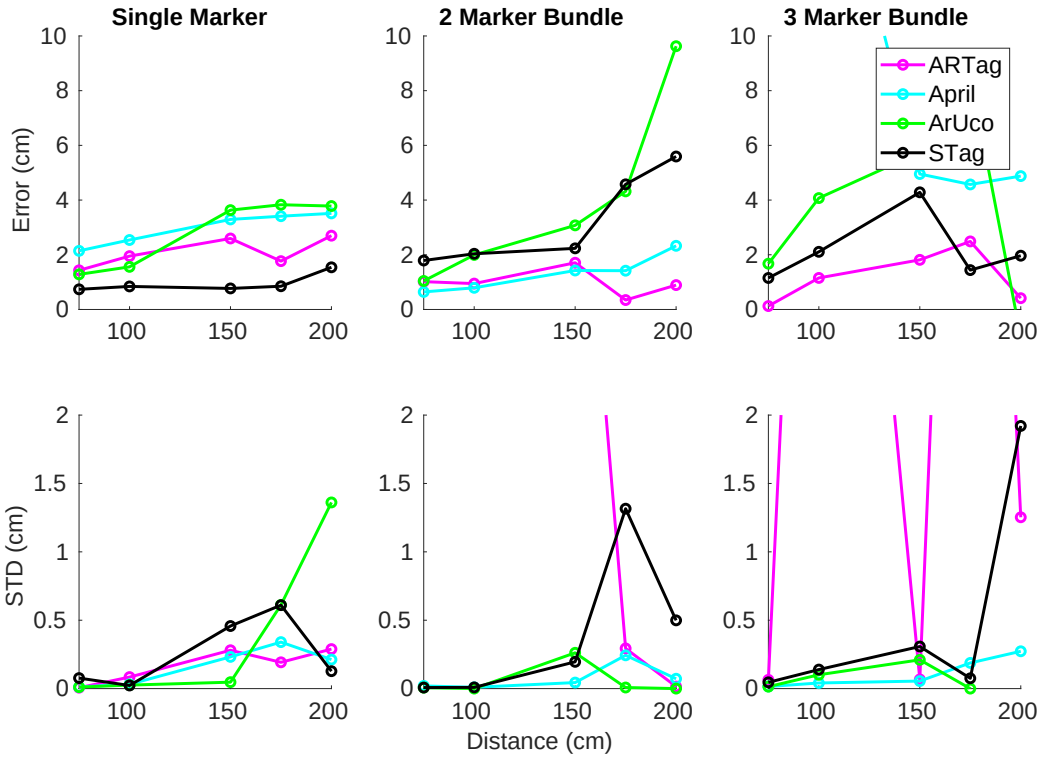


Figure 4.8: Distance results between the three different marker configurations.

The accuracy of the pose measurement and rate of detection for each package at varying distances are summarized in Table A.1 for individual markers, in Table A.2 for two marker bundles and in Table A.3 for the three marker bundle. For individual markers and two marker bundles the full range of experiments were executed using both cameras and lighting conditions. For the three marker bundle however, only the piCam under normal lighting conditions is used. This is done due to the superiority of the piCam in the majority of experiments and the data indicating that the lighting conditions have minimal effect on the marker systems as will be shown in the following discussion.

The metrics used for the accuracy of pose measurements are the average error and the standard deviation of the error for all successfully detected images in the experiment. The pose measurements provided by the fiducial markers are compared to the ground truth provided by the MoCap to yield the error and standard deviation presented in Figure 4.8.

While AprilTag and STag allow the user to specify a list of markers to detect, the other packages return a list with all the markers detected. Sometimes these lists contain false-positive detections. The detection rate reported in the tables shows the percentage of the images in which the marker is successfully detected. The false-positive rate is not measured for AprilTag and STag because they will not return any result for non-specified markers. For ARTag and ArUco the successful detections represent images that include the marker with the correct ID.

Individual markers

For the individual markers, STag is the package that provides the best accuracy in all cases with consistent performance at all distances. Moreover, STag is also the package that has the most precise measurements with ARTag and AprilTag often having comparable performance. In all the packages, the standard deviation increases with the distance indicating less precise results. Finally, AprilTag and ArUco show a consistent degrading in accuracy especially for the high resolution images.

The package with the best detection rate in most cases is AprilTag, while ARTag has a consistently low detection rate. ArUco and STag also have high detection rates, comparable to that of ArpilTag. AprilTag, ArUco, and STag have a better detection rate in the high resolution images and ARTag has a better performance in the low resolution images.

In general, the results from the low resolution images have a lower error in most settings. The piCam has a higher contrast and produces sharper images than the

Logitech camera which as a webcam is designed to produce smoother images. These more smooth transitions on edges yield lower gradients which makes the corner detection less accurate and robust. Moreover, in most cases there is no significant difference between the images under normal light conditions and the ones with the shadow. There are only two cases for the ARTag markers where the shadows cause the package to fail to detect any images.

Two marker bundle

For the two marker bundle, ARTag is the package that has the best accuracy in most settings with AprilTag and STag following closely behind. The package with the most precise results in this configuration is ArUco. While ARTag has the most accurate results, it is also the package with the worst precision in most cases. This is because ARTag has a higher rate of misdetections resulting in extreme outliers in some cases. Because of these extreme outliers, some entries have been omitted from Table A.2 since they were orders of magnitude larger than the rest. Such outliers create the need in practical applications to filter the data before use. Finally, AprilTag and ArUco again show a consistent degrading in accuracy as the distance increases.

In terms of detection rate for the two marker bundle (as seen in Table A.2), ARTag has a perfect detection rate in all cases. All of the packages have increased detection rates that in many cases match ARTag's results. AprilTag is the only package that fails to detect any marker at a distance of 2m while STag has a significantly reduced detection rate at the same distance.

Three marker bundle

For the three marker bundle the results are similar to the two marker bundle with ARTag having the most accurate results and higher detection rate but the worst precision. AprilTag results exhibit an unanticipated pattern for this configuration

with the accuracy being worse in closer distances than it is at further distances. STag again has a reduced detection rate at further distances while ArUco fails at 2m.

Figure 4.8 shows a comparison of the distance results for the three configurations used. The results used for the figures are produced using the piCam under normal lighting conditions. From the results it is clear that using multiple bundles does not have any significant impact on the accuracy of any package in close distances. On the other hand, STag and ArUco quickly deteriorate at further distances. At close distances, most of the packages have comparable accuracy and all of the packages except ARTag have increased precision. At further distances both precision and accuracy are worse. As explained in Section 4.2, the markers used in the bundles are smaller than the marker used in the individual package. This difference in size could explain the reduced performance in the bundles. It also explains why some packages have reduced detection rates at higher distances specifically in the low resolution images. One clear advantage of marker bundles is the increased detection rate. In the case of ARTag especially, using bundles doubled its detection rate.

4.3.2 Performance with varying marker orientation

Tables A.4 and A.5 show the estimated orientation results for varying marker orientation for a single marker and a two marker bundle respectively. Similarly, Tables A.6 and A.7 show the position results for varying marker orientation for the two configurations. Finally, the results for the non-planar marker are presented in Table A.8.

The orientation results for the individual markers show that AprilTag has an overall better performance. ArUco and STag have comparable performance in terms of the accuracy and precision of the results, as well as in terms of detection rates. This indicates a high aptitude to detect markers within images regardless of the orientation of the marker. ARTag consistently has a much lower detection rate hovering around 45% for the Logitech camera data and 49% for the piCam data. Despite the great

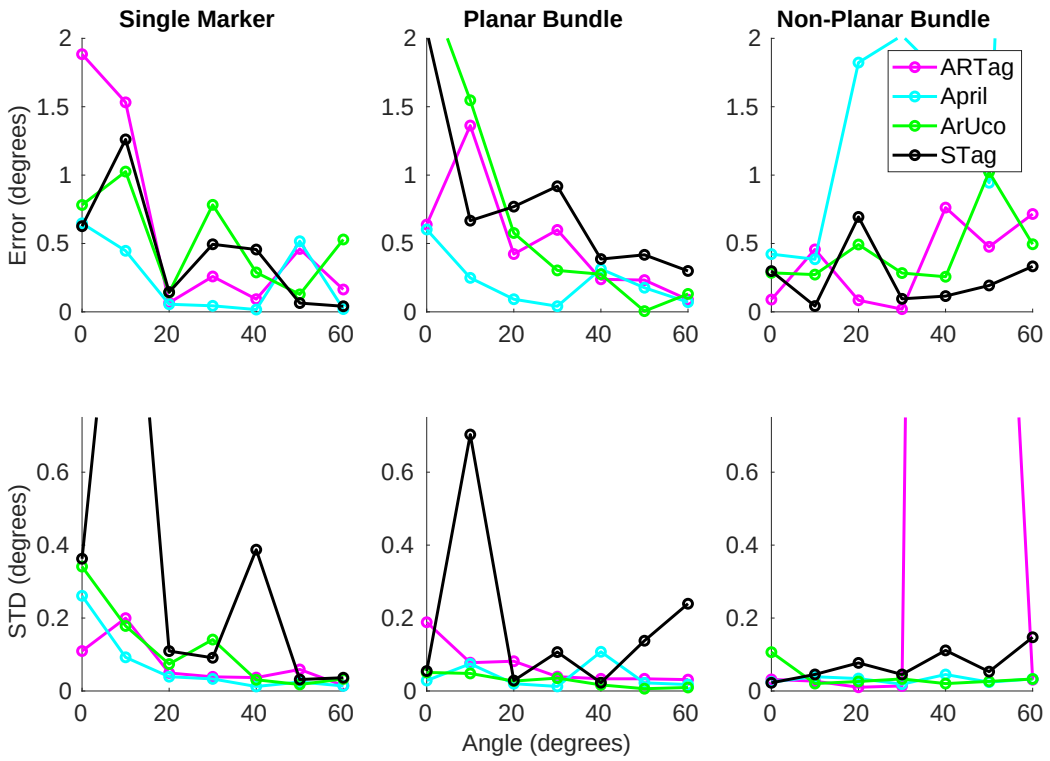


Figure 4.9: Angle results between the three different marker setups.

disparity in the ARTag detection rate, the mean error and standard deviation are generally competitive with the other three platforms. Overall, in the high resolution images, all packages tend to perform best at an angle between 40° and 70° where the estimated position is accurate within 1°. This is also the range that consistently experiences some of the lowest standard deviations for each marker.

Table A.6 indicates that the distance estimations are more accurate when the marker is perpendicular to the camera. All packages experience a degradation in their accuracy as the angle between the camera and the marker increases with slight improvements at 70° and 80°. For most angle offsets, STag presents the most accurate distance estimate, though in rare cases ArUco or ARTag outperform it. ARTag or AprilTag feature the lowest standard deviations with regards to the distance estimation for the varied angles.

For the two marker bundles, the results shown in Table A.5 signify a similar

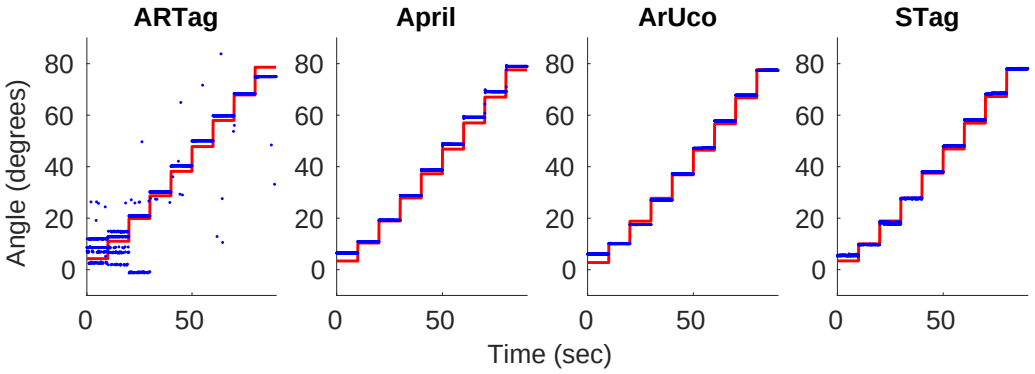


Figure 4.10: Angle results for the planar two marker bundle with the high resolution, low contrast camera.

performance as the single marker. AprilTag again has the overall better performance with ArUco and STag showing comparable results in many cases. The major difference in this configuration versus the single tag configuration is the significant difference in results between the two cameras used. ARTag has a significantly increased standard deviation showing the poor precision of the measurements for the Logitech camera when compared to the piCam. As shown in Figure 4.10, ARTag results experience a significant deviation in small angles while in larger angles the results include many outliers. While for all packages, the detection rate is around 20% lower for the Logitech camera, the orientation results are not significantly affected.

As for the position results, Table A.7 shows that ArUco has the overall better performance. ARTag and STag have comparable results in most cases. In this configuration, AprilTag is the only package that consistently degrades in accuracy as the angle increases. As before, the data from the piCam are much better than the results from the Logitech camera.

From the presented results, it is clear that the orientation measurements are better at angles between 40° and 70° while at small angles all packages produce their worst results. One way to improve these behaviors is the use of non-planar bundles. As shown in Figure 4.9, when the two marker bundle is used in a non-planar config-

uration, both the precision and accuracy of the results are improved. All packages except AprilTag have a better accuracy and all packages except ARTag in specific cases have better precision as well. Moreover, from Table A.8 it can be seen that the detection rate is not affected.

4.3.3 Performance with motion blur

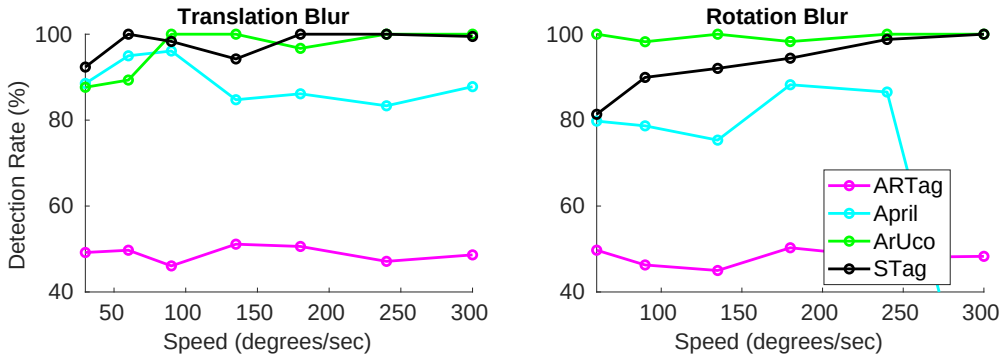


Figure 4.11: The effect of motion blur on the detection rate of each package.

The Tables A.9 and A.10 show the results of rotational and translational blur respectively. The data shown in these tables are the ones most affected in each case. For rotational blur, the results show the error and deviation of the measured marker orientation around the axis of rotation. For the translation blur, the results show the error and deviation of the x-axis position of the marker since this is the axis of motion of the marker.

In both cases and for all packages there is a significant amount of error and standard deviation. Moreover, the results deteriorate as the speed increases. As shown in Figure 4.7, the resulting images in high velocities are both blurry and distorted. The markers though, even in the worst cases are not distorted so much to justify the reported levels of error. Another reason for the reported levels of error that also explains the consistent deterioration of the results is the delay between the time that an image is captured until the results are produced. Moreover, the results

from the packages are compared to the measurements of the MoCap system that is capable of producing results at rates up to 360Hz. Because of the important difference in the latency between the two systems the reported accuracy results in Tables A.9 and A.10 cannot be used to evaluate the four packages.

One measurement that is not affected by the difference in latency however is that of the detection rate. The different rates are shown in Figure 4.11 for both motion blur types. AprilTag, ArUco and STag have an average detection rate of around 95% for static images of individual markers with ARTag having around 47%. In the case of motion blur however, AprilTag has a reduced detection rate averaging 89% for the translation blur and 81% for the rotation blur. Moreover, for the top speed of rotation blur, AprilTag fails to detect any bundle. The rest of the packages do not experience a significant change.

4.3.4 Computational Cost

Figures 4.12, 4.13, and 4.14 show box plots for the computational cost of each package for both a single marker and the two marker bundle on the NUC, TX2, and Pi platforms respectively. The CPU usage is reported as the percentage of a single CPU core of the platform while the memory usage is shown as the percentage of the total memory used. While the computational cost for processing the high resolution images is shown for the NUC and TX2 platforms, the processing cost for the low resolution images is shown in the case of the Raspberry Pi. The computational cost for processing the low resolution images in the NUC and TX2 is included in our previous work.

For a single marker, AprilTag is in all cases the most computationally expensive package. ArUco and ARTag are in general the least expensive packages in terms of CPU usage. STag used more resources, subsequent improvements have allowed for significant improvements, especially on the NUC. Furthermore, for the two marker

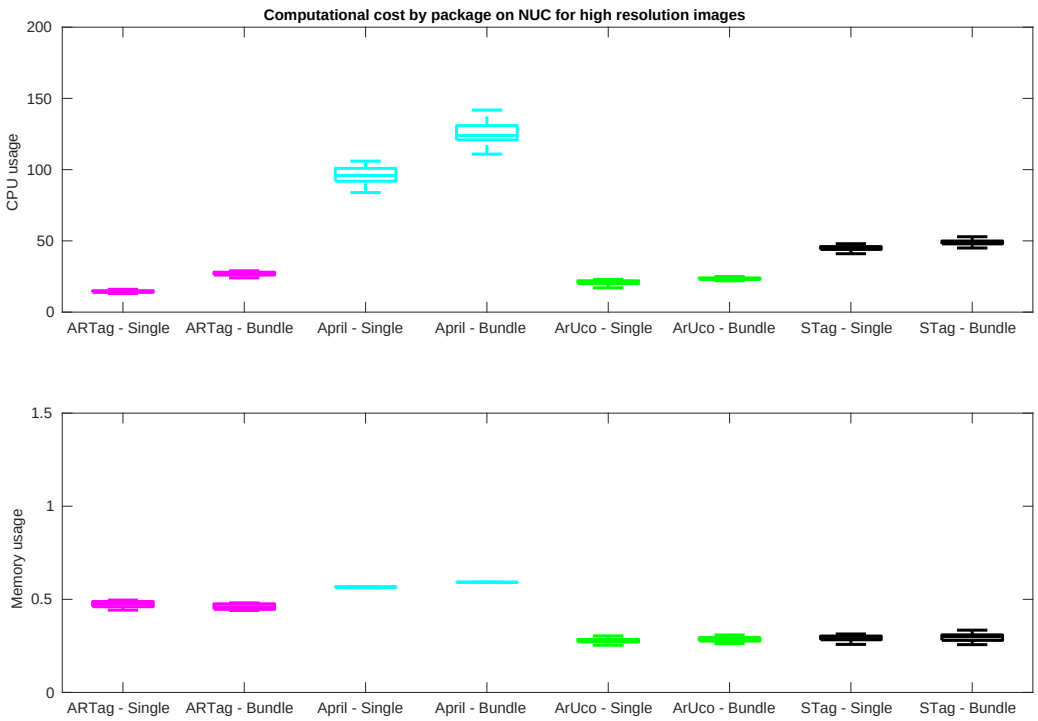


Figure 4.12: Processing cost on the Intel NUC computer for the high resolution images (1280×720 pixel^2). The CPU usage is reported as the percentage of a single CPU core of the platform and memory usage as a percentage of the total memory.

bundle on the Raspberry Pi, the performance of STag is comparable to that of ARTag and ArUco.

In terms of memory requirements, AprilTag again requires the most resources followed by ARTag. Both STag and ArUco have low memory needs. Contrary to the CPU usage though, the memory usage for all packages is insignificant, reaching a maximum of 11% on the Raspberry Pi and levels of 1% or lower on the more powerful platforms.

As for the bundle detection the only package that requires increased computational resources is ARTag. For bundle detection the CPU usage of ARTag are at the same levels of ArUco and at times STag. On the NUC and TX2 AprilTag also has an increase on CPU usage as does ArUco on the Raspberry Pi. As for the memory usage, there is no significant change in any of the packages when used for bundle detection.

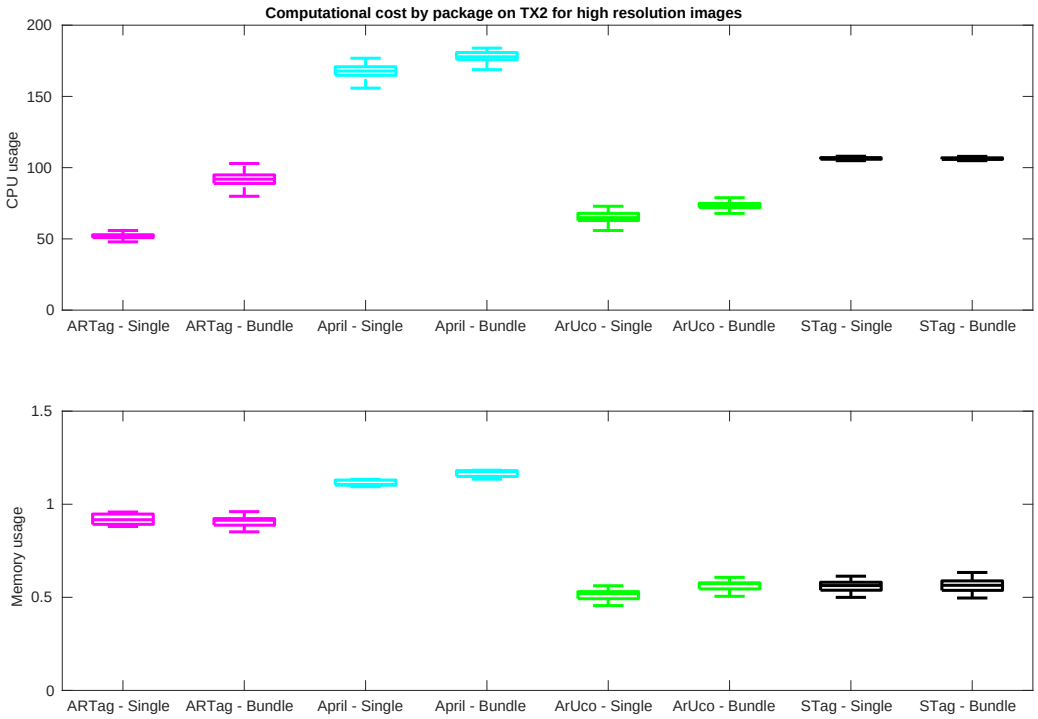


Figure 4.13: Processing cost on the NVidia TX2 computer for the high resolution images (1280×720 pixel^2). The CPU usage is reported as the percentage of a single CPU core of the platform and memory usage as a percentage of the total memory.

4.4 Remarks

Through our experiments, our first observation was in relation to the cameras used. The high resolution camera (1280×720 pixel^2) that has three times the resolution of the piCam (640×480 pixel^2) was initially expected to produce better results. Due to their field of view, the area that a single marker was covering was much larger for the Logitech camera (at $0.75m$ the marker was covering an area of around 310×310 pixel^2 for the Logitech and around 110×110 pixel^2 for the piCam). However, the images from the low resolution camera with the wider field of view turned out to produce better results in most cases. We attribute this difference to the quality of the images produced since the piCam has a higher contrast and better white balance. On the other hand, the Logitech camera produces images with smoother edges that lead to decreased accuracy and especially precision.

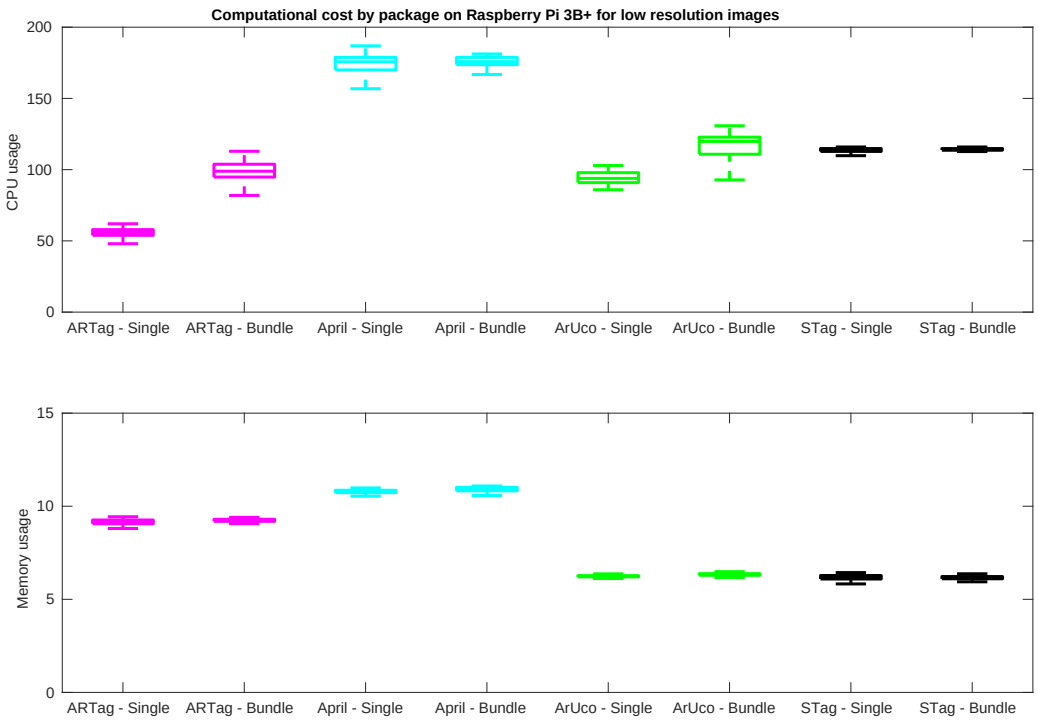


Figure 4.14: Processing cost on the Raspberry Pi 3B+ computer for the low resolution images (640×480 pixel^2). The CPU usage is reported as the percentage of a single CPU core of the platform and memory usage as a percentage of the total memory.

For individual markers, our results show that AprilTag, ArUco and STag all present high detection rates in almost all settings tested while ARTag has a significantly reduced detection rate. AprilTag is the package that has the best overall results in terms of orientation measurements while STag is the package with the best performance for position measurements.

When comparing the results from individual markers to the ones from bundles of multiple markers our results suggest that one of the main advantages is the increased detection rate. All packages have increased detection rates; especially ARTag that doubles its rate with multiple markers. ARTag also has an increased accuracy when used with bundles but the precision of the results is worse and it is prone to extreme outliers especially for orientation measurements. Finally, since the markers used in the bundles have smaller sizes than the single ones, the results and the detection rate of all packages deteriorate fast as the distance increases.

As for the multi-sized three marker bundle, no improvement was noticed in the results or detection rate when compared to the two marker bundle. The only advantage of such a configuration is that the small markers are still detectable even when the camera is in close proximity to the bundle. On the other hand, the non-planar bundle did improve the orientation results when compared to both the single marker as well as the planar bundle of the same size.

The results of our motion blur experiments are inconclusive since we cannot confidently say what part of the reported error levels can be attributed to the effect of the motion blur and what part can be attributed to the delay between the fiducial data and the MoCap data. However, the detection rate is independent of the latency of the process. AprilTag is the only package that has a reduced performance in both types of motion blur.

In terms of computational efficiency, AprilTag is the package that requires the most resources in both CPU and memory usage. STag has the second highest CPU usage but after our efficiency improvements it's cost is now closer to ArUco and ARTag and in some cases performs better. ARTag and ArUco are the most computationally efficient packages. While all packages require little memory, AprilTag and ARTag use more than ArUco and STag.

Comparing the computational cost between a single marker and a two marker bundle, AprilTag and ARTag are the two packages that show a noticeable increase in their CPU usage. The memory usage remains at the same levels for all packages.

Summarizing, in this work we identify the different design parameters for pose estimation frameworks using fiducial markers. Then we investigate how these parameters affect the accuracy and robustness of the results for four different packages. Table 4.2 summarizes the key advantages and disadvantages of each package. Our results suggest that AprilTag and STag are overall the packages that produce the best results with ArUco having a competitive performance. The use of multi-marker bundles can

increase the performance of all packages however a single larger marker will perform better than multiple smaller ones in greater distances. Furthermore, using non-planar bundles is an effective way to improve the orientation measurements. When used with bundles, ARTag had an increased deviation with some extreme outliers that might cause an erratic behavior when they are used directly in a controller. Another important parameter is the camera sensor used, where our experiments show that contrast is more important than resolution. The induced noise from casting shadows to the markers did not have an effect in most cases suggesting a high robustness to different lighting conditions and the only package that was affected from motion blurring was AprilTag. Finally, AprilTag is the most computationally expensive package making it an unsuitable choice for low-powered devices.

Table 4.2: Experimental comparison summary: Main advantages and disadvantages of the four evaluated packages.

Marker	Pros	Cons
ARTag	• Lowest computational cost overall	• Low detection rate for single markers • Extreme outliers & high Std Dev in marker bundles
April	• Great orientation results and detection rate • Good position results	• Most computationally expensive • Most sensitive to motion blur • Worst results in the non-planar setup
ArUco	• Good position and orientation results • Great detection rate • Low computational cost for single markers	• Sensitive to smaller marker sizes and larger distances • Computational cost scales with multiple markers
STag	• Great position results and detection rate • Good orientation results	• Sensitive to smaller marker sizes and larger distances

Chapter 5

Structural Health Monitoring using a StereoDIC enabled UAS

5.1 Introduction

In UAS inspections such as the ones shown in Chapter 3, the result is a set of images that can be processed to produce a map or a reconstruction of the inspected structure. The overall condition of the structure is visible in such results and then areas of interest can be selected to look into individual images and detect large scale changes or defects. However the results of such inspections are purely qualitative.

In this chapter, we investigate the feasibility of an autonomous UAS capable of performing structure inspections that produce quantitative results. Therefore, we equipped the UAS with the sensors required to perform StereoDIC, a non-contact method able to measure 2D and 3D shape and deformation fields. This chapter describes the system setup used and the experimental setup and results that validate the effectiveness of the proposed system.

The work presented here was published in [49, 48]. The remainder of the Chapter is organized as follows. In Section 5.2, the system used is presented with special emphasis on the StereoDIC system. Section 5.3 details the experimental setup used, and finally Section 5.4 presents the experimental results.

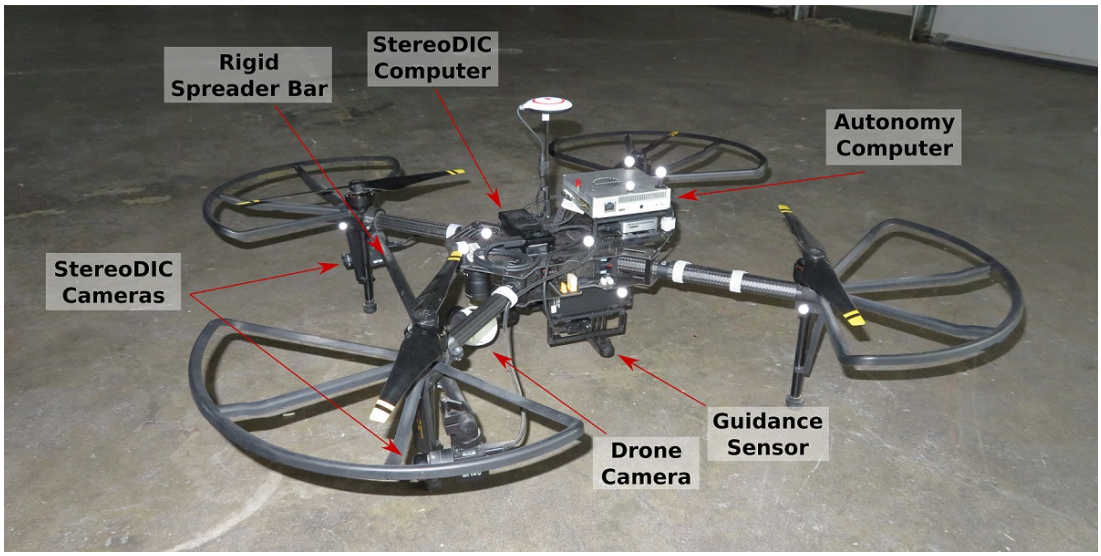


Figure 5.1: The DroneDIC system with all the peripheral devices and sensors noted.

5.2 System

The same UAS platform presented in Section 3.2 is used as the base of the overall system with the most significant change being the addition of the StereoDIC system. The platform, named DroneDIC, equipped with the StereoDIC system is shown in Figure 5.1. In this Section, the changes in the autonomy of the system will be discussed followed by a detailed presentation of the onboard StereoDIC system.

5.2.1 Autonomy

Large structures can degrade the GNSS signal by either completely blocking a region or by causing reflections [35]. Since the system is expected to operate in close proximity to large structures, it cannot rely on GNSS measurements for its positioning. As in Section 3, fiducial markers are again used here to provide a vision-based positioning framework. Apart from providing position measurements, the fiducial markers are used to identify the inspection area as well. In a large structure, there might be multiple points of interest. Using different markers in each area, the UAS can identify each point of interest and accurately position itself during the data acquisition

process.

To increase the accuracy of the positioning framework, the DJI Guidance[1] sensor kit was included. The DJI Guidance is a visual sensing system used for positioning and obstacle avoidance. One sensor from the Guidance system, featuring stereo cameras and an ultrasound proximity sensor was included on the UAS. This sensor, pointing downwards can provide measurements about the drone's altitude as well as its vertical velocity.

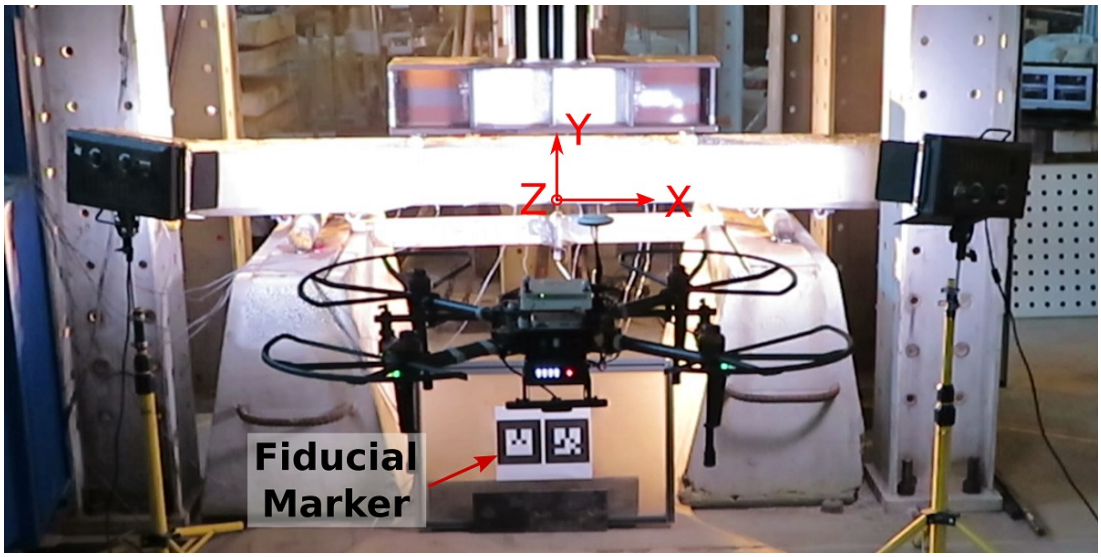


Figure 5.2: The UAS during the data acquisition process. The specimen fixed reference frame is shown as well as the fiducial marker used for the position measurements.

A fixed reference frame placed at the center of the specimen is used to define the UAS position and orientation as shown in Figure 5.2. The frame's Y-axis is perpendicular to the gravity vector and it is assumed that the center of the frame has a specific, known offset from the center of the fiducial marker. The positioning framework is based on an Extended Kalman Filter. The filter uses the IMU measurements to update the system's state and then the ARTag marker position measurements, and the altitude and velocity measurements from the Guidance sensor are used for the correction step.

[1] `https://www.dji.com/guidance`

As for the drone's flight controller, the same controller architecture as described in Section 3.4.3 is used. However, since the inclusion of the StereoDIC system affects the flight dynamics, the model used needs to be changed. Using flight data from the augmented system, the modified system dynamics was determined using a least-squares method. After the new model was derived, the LQR gains were tuned to favour smooth control inputs rather than more aggressive maneuvers. In contrast to the expected behavior of the UAS in Chapter 3, the UAS in this case needs to be able to hover in place as smoothly as possible. To allow aggressive maneuvers that enable the UAS to land on a moving platform, a set of different gains can be saved and used accordingly.

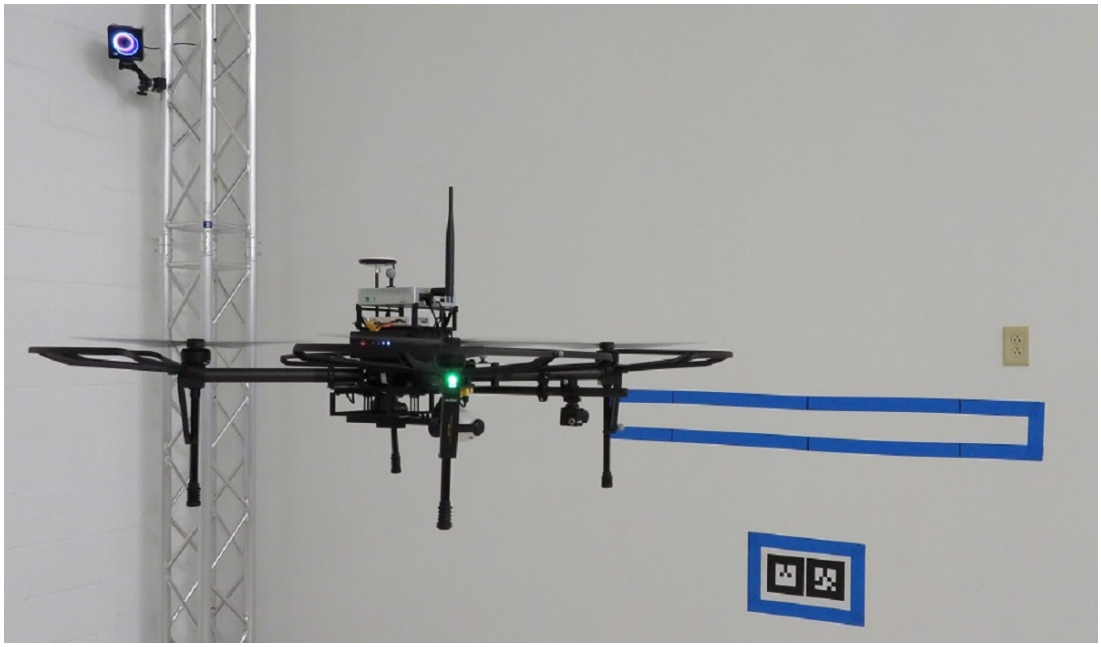


Figure 5.3: Lab experiments to validate the accuracy of the developed vision-based positioning and the control scheme.

The positioning framework and the developed controller were tested in the Mo-Cap room before deployment. After analyzing the camera hardware specifications Section 5.2.2 and the geometry of the experiment area Section 5.3.1, it was determined that the drone needs to hover in front of the specimen at a distance of $1.9m$. At this distance, the field of view of the DIC cameras captures the full specimen, while

allowing for a small motion of the drone. An area of the same size as the specimen was marked on the wall to verify that the specimen remains in the field of view for the duration of the experiment as shown in Figure 5.3. Figure 5.4 shows the UAS position as calculated by the image-based positioning framework. The dashed line signify an area that is $5cm$ in any direction around the desired position. This is the area that the drone can safely move and keep the specimen in the field of view of the DIC cameras. The standard deviation in position for the x and y axes is around $2cm$ while for the z axis it is around $1.5cm$. The data demonstrates that the drone is in the desired region more than 95% of the total flight duration.

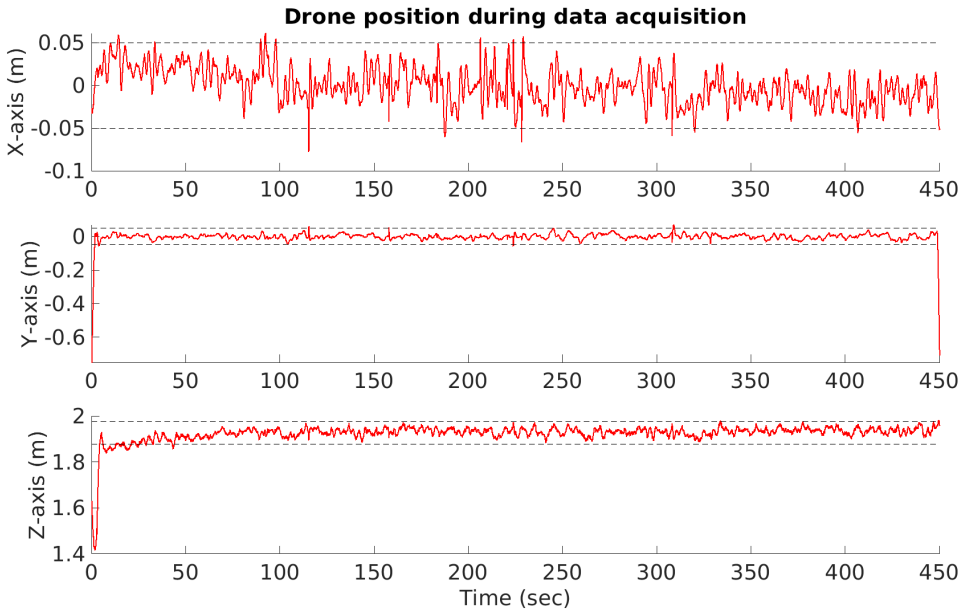


Figure 5.4: UAS position as measured from the vision-based system during the data acquisition process.

5.2.2 StereoDIC System

Due to the limited processing power of the autonomy computer, the StereoDIC system runs independently. An Intel Compute Stick is used to control the onboard imaging system, allowing the user to remotely monitor the system and to activate the frame capturing software VIC-Snap®. Since the stereo vision system is mounted to the

moving UAS platform, it is expected that there will be significant motion of the cameras during the data acquisition process. The effect of rigid body stereo vision system motion has been studied previously [96, 8] and compensation methods have been identified and are available in software, including commercial DIC software such as VIC-3D®. Another issue expected to arise from the motion of the UAS is motion blurring. The impact of motion blur on DIC measurements was studied in [7] and while it was found that there is no significant impact on shape measurements, the authors reported that there is an impact on strain measurements. Motion blur occurs when either a moving object is being captured or a steady object is imaged by a moving camera, which is our case. To minimize the motion blur effect in an image, one of the most effective ways is to minimize the exposure time. Consequently, the exposure time was set to $2ms$.

By reducing the exposure time, the amount of light incident on the sensor plane is reduced, resulting in darker images. To counter this effect, the aperture of both lenses is set at the minimum value of F#1.8. Moreover, rubber shock absorbers are employed to reduce the effect of motor and propeller vibrations on the cameras. Finally, the onboard mini-computer offers a lightweight solution for image acquisition and storage. However, this choice reduced available processing power, imposing limits on both camera resolution and image acquisition rate. Rates up to $5Hz$ were achievable with the 2Mpixel sensors. Summarizing, the onboard stereoDIC system is composed of:

- 2Mpixel Point Gray/FLIR cameras with CCD sensor and global shutter
- $12mm$ focal length, F#1.8 Edmund Scientific lenses
- Extra stiff, low-weight carbon fiber beam for mounting the cameras
- Rubber shock mounts between the carbon fiber beam and the UAS

- The Intel Compute Stick mini-computer for operation of the stereo-vision system and storing images

5.3 Experimental Investigation

This section presents the experimental validation of the proposed DroneDIC system through comparisons of measurements acquired with conventional instrumentation and a well-established, and proven, ground-based StereoDIC system. To this end, measurements are taken during the loading to failure of a prestressed concrete railroad tie subjected to four point bending. The test setup, instrumentation, and loading sequence are discussed in detail and the StereoDIC data acquisition process is detailed. Finally, the images are transferred to a more powerful computer for the measurement of full-field deformations and surface strains and the detection of damage.

5.3.1 Concrete Tie Specimen Description

The prestressed concrete tie used in this validation study is part of an extensive research program conducted at the University of South Carolina for high speed rail applications. The overall length of the tie is $2.591m$ and the detailed geometry is shown in Figure 5.5. Prestressing is applied through eight 7-wire, ASTM A886 grade 270 low relaxation strands, $9.53mm$ in diameter, placed in two rows. Details of the design parameters and materials are reported in [101]. As shown in Figure 5.6, the specimen is tested in four point bending by applying downward vertical forces at two roller contact points; the design capacity of the tie is $M = 18.31kN - m$.

5.3.2 Test Setup

As shown in Figure 5.6, the four point load testing is conducted in a reaction frame that consists of four $4.88m$ long W12 $\times$ 87 columns, two $2.135m$ long W18 $\times$ 119 beams and two $1.22m$ long W16 $\times$ 89 beams. In addition, a $1.525m$ long W18 $\times$

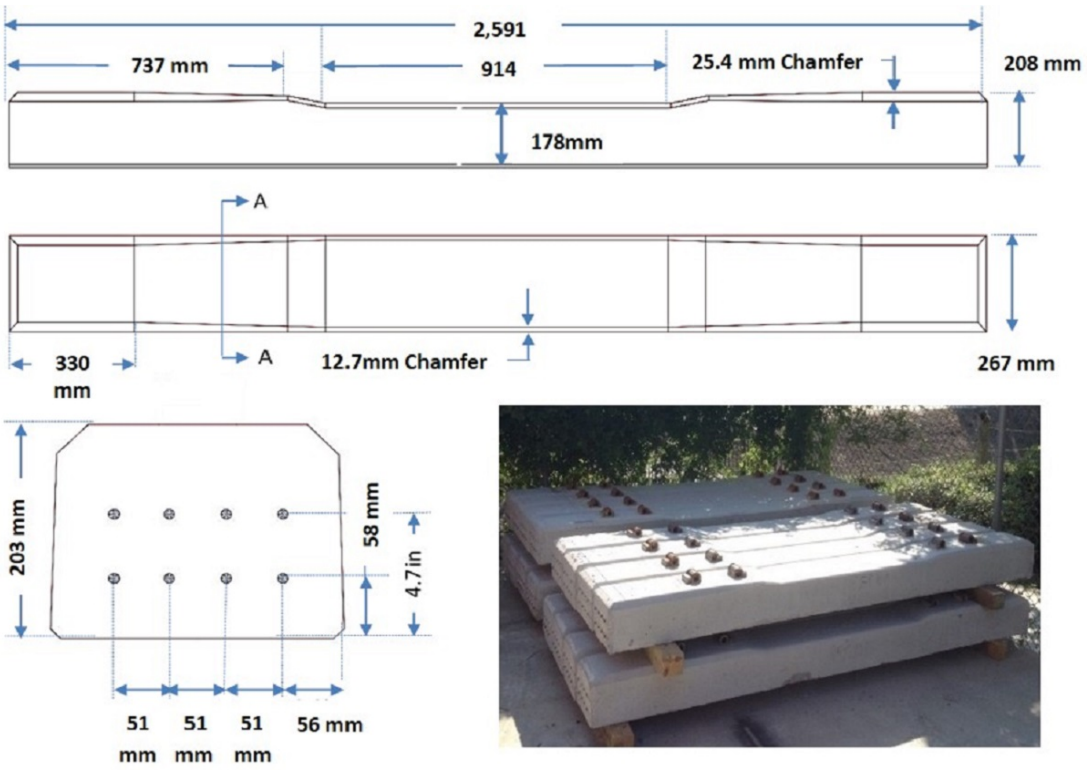


Figure 5.5: Geometry of the prototype with the finished product shown in the lower right corner of the image.

119 cross beam is attached to the long beams and carries the actuator. Two spreader beams, $0.2033m$ and $1.0167m$ length, respectively, are attached to the actuator head to transfer the load to the test specimen. The actuator is a $489.3kN$ MTS 5 with a $0.1525m$ stroke. The actuator is connected to a 151.4 lpm MTS hydraulic pump. The actuator is also equipped with a load cell and a Linear Voltage Differential Transformer (LVDT). All tests are carried out in load control mode. Two concrete blocks of trapezoidal cross sections provide the supports for the specimen. The blocks are $1.22m$ long, and $0.7625m$ tall, while the base is $0.559m$ wide and the top is $0.305m$ wide. The supports can be placed at variable spacing to accommodate the test requirements. The reaction frame is shown in Figure 5.6.

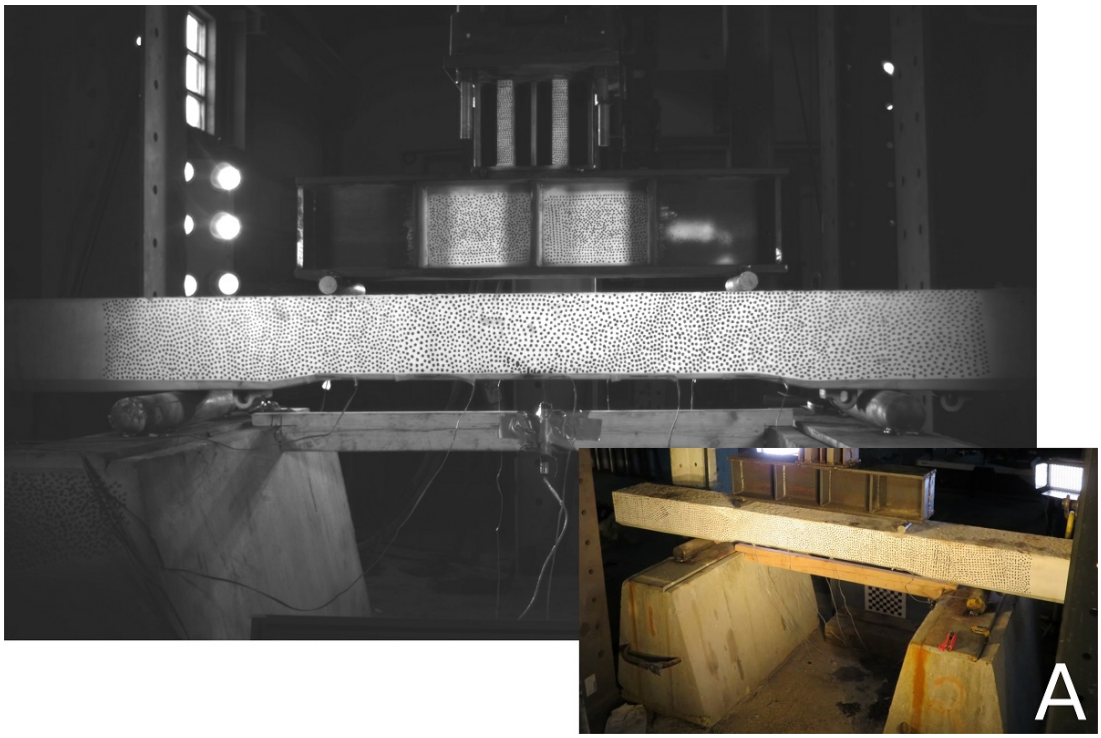


Figure 5.6: The reaction frame used for the load testing of the concrete tie. Image shows the four point testing for negative moment. Detail A shows the concrete blocks that provide the support to the specimen.

5.3.3 Instrumentation for Validation Measurements

For validation of the DroneDIC capability to acquire reliable measurements on a full-scale structure, additional instrumentation is added for comparison to the DroneDIC measurements. The additional instrumentation used during testing consists of a LVDT and a stationary (ground-based) StereoDIC system. The LVDT is used to measure the deflections at midspan of the specimen. The LVDT is secured to a single two by four board, with the tip contacting the surface of the tie at its midspan and at mid-width. The tip of the LVDT is adhered to the specimen by a small amount of epoxy. The stationary StereoDIC system images the "back side" of the specimen while the drone-based DIC measurements are acquired on the "front side" of the tie. Following the best practices guidelines developed by the International Digital Image Correlation Society [11]. Table 5.1 summarizes the stationary StereoDIC system pa-

Figure 5.7: Stationary StereoDIC system used for validation in this work and its components.

rameters and calibration grid data. Moreover, low heat emission lighting, a computer for image acquisition and cables were used, with the full list of components shown in Figure 5.7. The calibration target was used for the calibration of both the drone and ground-based StereoDIC systems. As noted previously, Correlated Solutions Inc. software VICSnap and VIC-3D are used to perform the image acquisition and DIC processing, respectively. Details on the data acquisition and image processing procedures as well as validation and application studies of the stationary StereoDIC system used in this work have been published in the literature [73, 101, 90].

Table 5.1: Stationary StereoDIC specifications

Parameter	Value
Cameras	Grasshopper3 GS3-U3-91S6M (8bits, $3376 \times 2704 pixel^2$)
Lenses	Schneider XENOPLAN 1.9/35-0511
Lens Filter	IR Filter with cross polarizer
Field of View	$1520 \times 1220 mm^2$
Stand-of Distance	2m
Average Digital Magnification Factor	2.2mm/pixel
Calibration Grid	9×12 Dot grid with 13mm dot diameter and 35mm spacing

5.3.4 Specimen Preparation

A speckle pattern is applied on both "front" and "back" sides of the concrete tie. First, the surface is painted with flat white paint to minimize reflections. Subsequently, the

speckles are applied manually over the entire surface within the field of view that includes a black marker. Speckles are applied in a way to create a well-balanced, high contrast, random pattern. The average diameter of the speckles is $6mm$. No other preparation of the specimen is required.

5.3.5 Loading Sequence

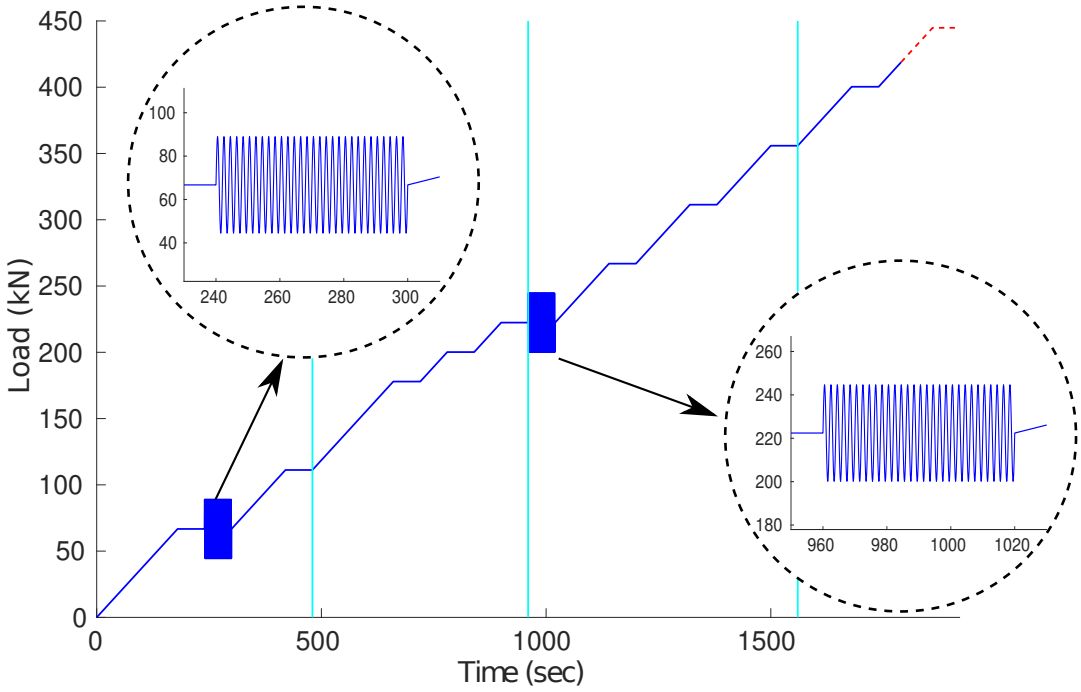


Figure 5.8: Loading sequence for experimental validation. Details show the harmonic loading of the specimen while the vertical lines show the 10 min load holds. Last ramp-and-hold portion of the loading line was not completed due to failure of the specimen.

The specimen loading process is performed to accommodate both the DroneDIC measurement process and the American Railway Engineering and Maintenance-of-Way Association (AREMA) guidelines for railroad tie experiments:

1. Loading rate of less than $370.7N/sec$ ($5Kip/sec$)

2. Multiple images acquired at constant load for (a) averaging of displacement and strain data from the stationary system to reduce variability in these measure-

ments and (b) to determine variability in the measurements obtained by the DroneDIC system.

a) DroneDIC system configured to continuous acquisition

b) Synchronization of multiple sensor systems performed manually

3. Stationary system is configured to acquire images manually and on demand

In view of these needs, the loading sequence is defined as shown in Figure 5.8. The loading sequence contains 1 min load holds at predetermined loads. At each hold time, the stationary system acquires multiple images at the specific load, while the DroneDIC continuously acquires images. Since the onboard battery can supply the required power for a maximum of $12min$, the load sequence also contains three $10min$ load holds to land the UAS, swap the batteries and immediately re-position the UAS to continue the measurements. The loading sequence also includes two, one minute, $0.5Hz$ harmonic loading segments with an amplitude of $\pm 22.24kN$. The first cyclic loading sequence is performed at $66.72kN$ before visible damage is observed. The second cyclic loading sequence is performed at $222.41kN$ after initial visible damage occurs. For both hold periods, the DroneDIC system continues to acquire images that are analyzed to demonstrate the dynamic acquisition capabilities of the system.

5.3.6 Calibration Data Acquisition and Processing

Both the DroneDIC and the stationary StereoDIC systems are calibrated before measurements are acquired. Calibration is performed based on either calibration target images, specimen speckle pattern images with a known scale factor or a hybrid mode. Though not required, in this work the DroneDIC system is calibrated prior to becoming airborne. The calibration process encompasses image acquisition at different angles between the camera and the target (or specimen) followed by image process-

ing to determine geometry parameters for the system, coordinate transformations between global and camera systems and the camera model parameters [85].

For the stationary system, before any load is applied, and then at each load hold, 50 images per camera are captured. The images at zero load are considered the baseline images for the correlation. During the specimen loading process, the stationary system acquires images only during the 1min load holds. The baseline images and the images acquired at each load hold are averaged six for each camera to reduce the noise due to the inherent variability in the pixel-level intensity values [85]. The averaged baseline pair of images and the averaged pair of images at each load hold are used in the correlation process to obtain displacement and strain data. A subset size of 45×45 pixel with an 11 pixel step size are selected for the image correlation. All components of the Lagrangian large strain tensor are calculated using a 5×5 filter size.

As noted previously, the DroneDIC platform acquires images continuously while in flight at a rate of five frames per second (fps). However, due to the relatively low-end cameras and computer hardware used in these studies, the achieved acquisition rate was not constant, varying between two and five fps. Because of the large relative motion between the cameras and the specimen, the image averaging technique for noise reduction implemented in the stationary system is not feasible, resulting in increased noise when selecting individual images for correlation. A subset size of 31×31 pixel with 7 pixel step size are selected for the image correlation. All components of the Lagrangian large strain tensor are calculated using a 15×15 filter size to provide additional local smoothing of the local displacement data for strain estimation.

5.4 Experimental Results

In the first part of this Section we present results that validate the accuracy of the measurements acquired by the DroneDIC. In the second part we present the

DroneDIC results that show how the system is able to detect the onset of damage and track its evolution during the experiment.

5.4.1 Experimental Validation for the DroneDIC Measurements

In this study, "validation" of the DroneDIC system pertains to the quality of the comparison between (a) the load-midspan vertical deflection data acquired by the DroneDIC system, the LVDT and the stationary, ground-based StereoDIC system and (b) the longitudinal strain field measured by the DroneDIC and stationary StereoDIC systems.

Mid-span vertical displacement

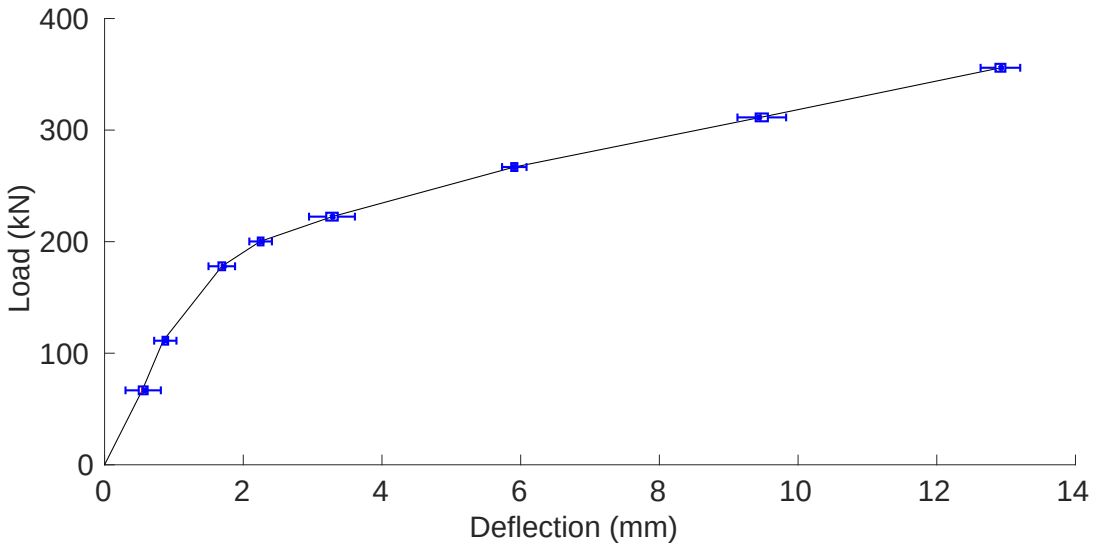


Figure 5.9: Box and whiskers plot for load vs midspan vertical displacement for the DroneDIC measurements.

First, we consider the DroneDIC data in more detail. As noted previously, the pixel intensity data from DroneDIC images are not averaged during load holds due to the continuous motion of the drone. Rather, during each load hold the DroneDIC system acquires multiple pairs of images of the specimen. The vertical displacement data is first computed by correlating each pair of images at a given load to the base-

line image pair. Subsequently, the average value is calculated at each load hold. Figure 8 shows a box-and-whiskers plot for load vs vertical midspan DroneDIC displacement measurement, where the mean, median and four quartile values for the displacement during each hold are shown. The standard deviation of the measurements varies from $0.08mm$ to $0.19mm$, with the largest values being present at the higher loads. It is worth noting here that the average pixel magnification for the DroneDIC is $0.98pixel/mm$. According to [85], a baseline standard deviation for in-plane deformations is on the order of 0.01 pixel, consequently, for the system's average magnification factor, a standard deviation of $0.01mm$ is given as an approximate noise floor. Experimental factors that were identified and are known to have contributed to the observed elevated variability are (1) known oscillations in the load by $\pm 100N$ due to electronic instabilities in the control system and (2) vibrations of the support structure due to the high pressure hydraulic system components. None of these factors are due to DroneDIC effects and hence should be observed with stationary StereoDIC systems

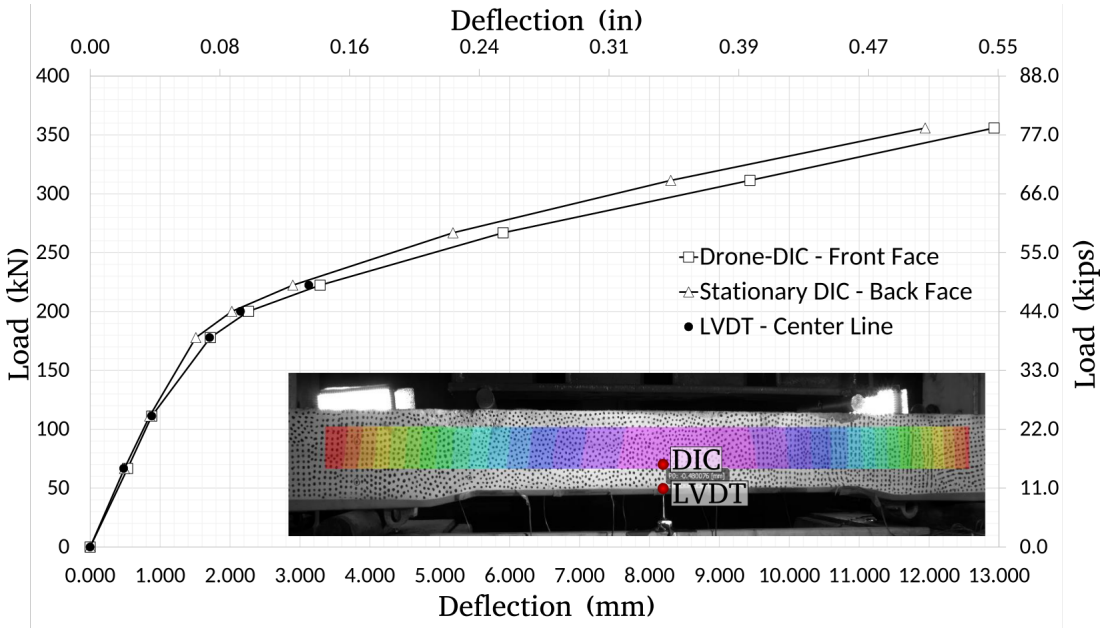


Figure 5.10: Load-vertical deflection data from DroneDIC, LVDT and stationary stereoDIC systems at mid-span of the specimen.

Using the average DroneDIC measurements at each load level from Figure 5.9, the comparison of the load-midspan vertical displacement measurements from the DroneDIC system, stationary StereoDIC system and the LVDT is presented in Figure 5.10 Also shown in Figure 5.10 is an inset with the full-field DroneDIC vertical displacement results as well as the location of the LVDT and the corresponding point in the DroneDIC measurements. It is noted that the LVDT is removed when the load reaches $250kN$ to prevent accidental damage to the instrument as specimen failure initiated. As noted previously, the ground-based StereoDIC measurements are acquired on the back face of the beam specimen. A corresponding point on the back face (not shown) is selected to extract the vertical deflection from the deformation map acquired with the ground-based StereoDIC system. Inspection of Figure 5.10 shows that all three curves are identical during the early stages (up to $111kN$), with separation in the measurements in direct correspondence to first cracking of the beam. Although the curves are in excellent agreement for all practical purposes, a slight deviation is observed after initial damage, with the LVDT data falling between the load-deflection data from the two DIC systems. Considering that: (1) the specimen supports do not provide an ideal contact throughout the width of the specimen, (2) the cracking that develops and propagates is not symmetric and thus the stiffness of the specimen deviates from a uniform value and (3) the actual physical location of the points where measurements are acquired by the DroneDIC, LVDT and stationary StereoDIC systems are at the front face, mid-plane, and back face of the specimen, respectively, the small deviation is attributed to a small twist rotation of the specimen with respect to the longitudinal centroidal axis of the beam.

Longitudinal strain fields and correlation uncertainty metrics

Figure 5.11 presents the measured longitudinal strain field for the DroneDIC and ground-based StereoDIC systems for a load of $177kN$. As shown in Figure 5.11,

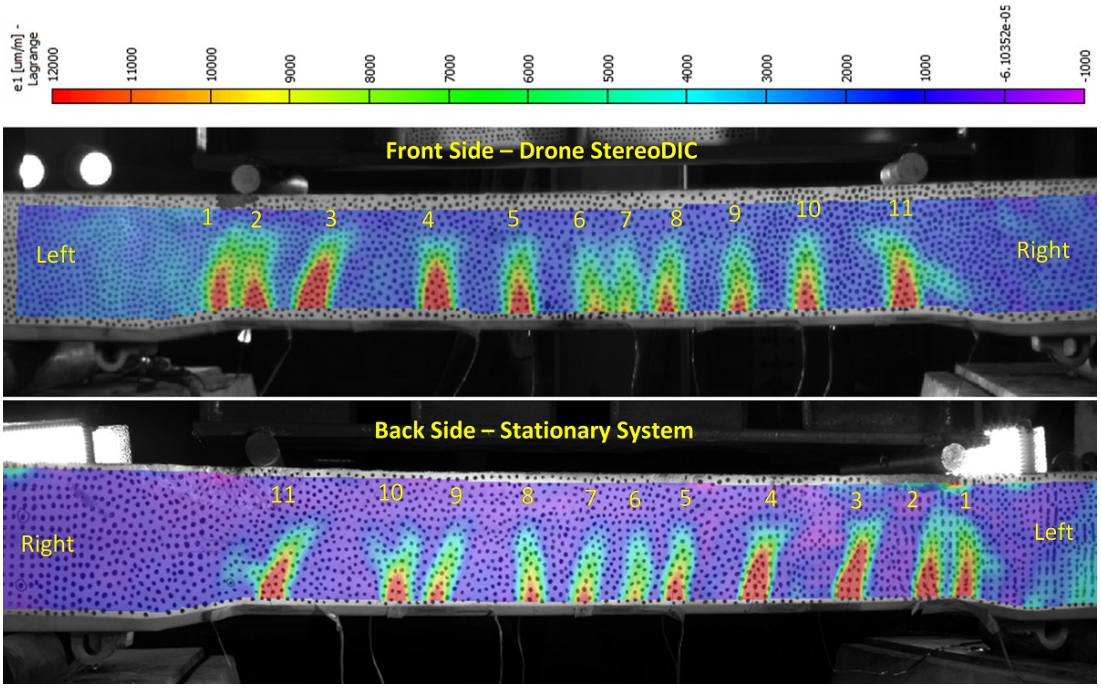


Figure 5.11: Measured longitudinal strain field, $\epsilon_{\chi\chi}$, for front (DroneDIC system) and back (ground-based stationary StereoDIC system) surfaces of specimen.

there is very good agreement in the strain data for both systems. At this load, the cracking of the beam has initiated, leading to the formation of several areas of apparent high strain at the bottom of the beam. Such strain measurements are referred to as "pseudo-strain" measurements in the literature [97, 92, 34]. The high strain measurements there are not real but are rather the damaged regions on the tension side of the specimen, that the subset-based correlation is converting to high strains. This is also evident in Figure 5.12, where the correlation confidence interval metric, sigma, is shown as color maps for both sides of the specimen. Due to formation of the cracks at the "pseudo-strain" areas, the speckle pattern has been damaged and thus at these areas the correlation is performed with reduced confidence. This suggests that the high strain measured there is not real but is rather due to damage. Eleven tension cracks are identified by both systems and were verified by visual inspection with respect to location and length.

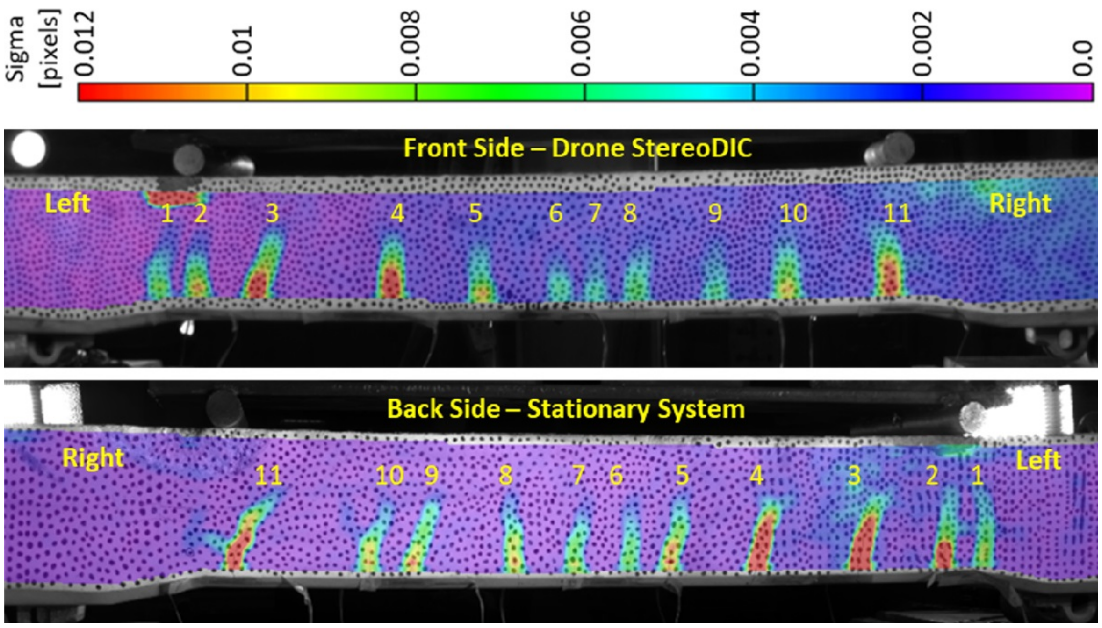


Figure 5.12: Map of measurement confidence intervals for front and back surfaces of specimen. Damage along the bottom of the beam is clearly visible as low-confidence intervals for measurements.

Time response of vertical displacement at midspan

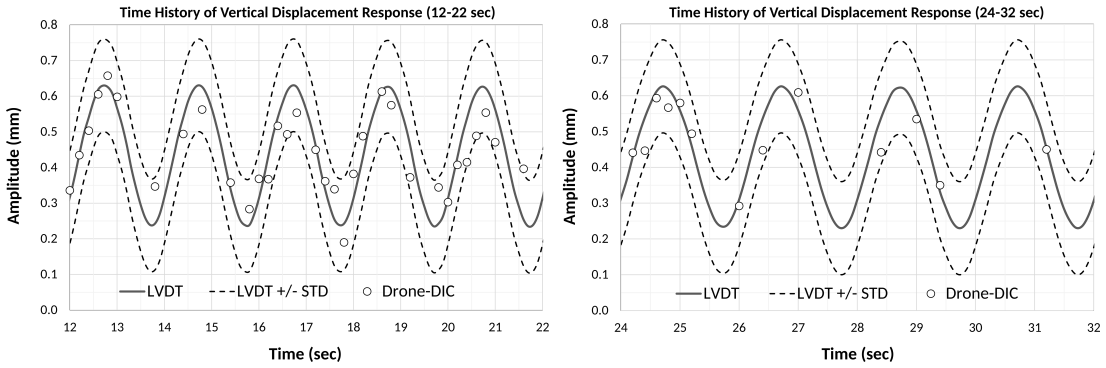


Figure 5.13: Time history of the vertical displacement response for the DroneDIC system and LVDT.

The time variation of the vertical deflection at midspan due the first harmonic loading (see Section 5.3.5) is shown in Figure 5.13 for the LVDT and DroneDIC measurement methods. The solid line shown in Fig. 12 corresponds to the deflection recorded by the LVDT every $0.1s$. The dashed line is the LVDT measurement ± 1 DroneDIC standard deviation ($0.13mm$) as it was found from the load hold at the

mean load of $66kN$. The DroneDIC data is shown as open circles. These results show that the LVDT and DroneDIC measurements are in a relatively good agreement, with the DroneDIC measurements being most of the times at most one standard deviation away from the LVDT measurements. The average error during this harmonic loading was found to be $0.06mm$ with a Root Mean Square Deviation of $0.08mm$. Although the camera acquisition rate is set to $5fps$, the actual rate was not steady, and it was occasionally recorded at rates lower than $1fps$. This is attributed to the low-end cameras and computer hardware used in this first prototype. The left graph in Figure 5.13 shows the response when the image acquisition rate is closer to the set rate. The right graph in Figure 5.13 is obtained when the acquisition rate has deteriorated. It is further observed that when the acquisition rate is low, the accuracy is somewhat better. In either case, the camera resolution (both temporal and spatial) is low for displacement measurements. It is apparent that higher-end hardware is needed to address the resolution issue. As shown in Figure 5.14, when the StereoDIC data is used in a Fourier transformation, the dominant frequency found matches the frequency of the load at $0.5Hz$ as shown in Figure 5.14. Thus, even with our hardware limitations, the temporal trends can be identified clearly.

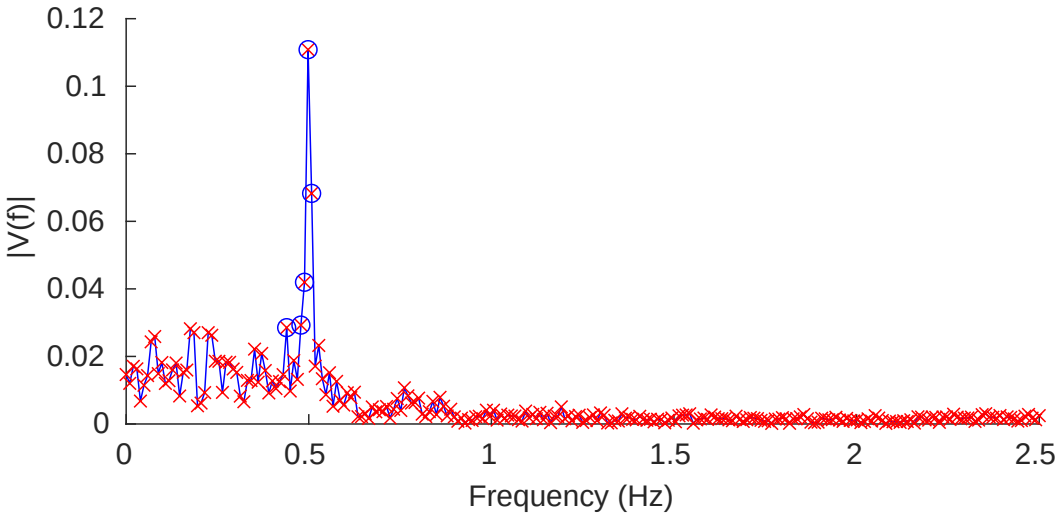


Figure 5.14: Single-sided amplitude spectrum for the midspan vertical deflection.

5.4.2 DroneDIC Measurements

To demonstrate the versatility and effectiveness of the DroneDIC measurement system, several different types of measurements that were captured in a single flight of the drone are shown in this section. The measurements include (a) vertical displacement of the bottom portion of the beam, (b) comparison of strain fields obtained by both DIC systems.

Beam Centerline Displacement

The vertical deflection of line AB, along the bottom portion of the beam, measured during flight by the DroneDIC system, is shown in Figure 5.15 (see Inset image for location of line AB). The DroneDIC system provides a dense, low-noise set of data showing the vertical motion of the full beam length. Historically, this deflection was experimentally measured through the use of multiple LVDT's placed at single points along the length of the beam, with curve-fitting used to estimate the entire displacement curve. The deflection curve is constructed afterwards by fitting a curve through the LVDT measurements.

As shown in Figure 5.15, data is obtained between the two supports. For loads less than $200kN$, where initial cracking occurs, the measured deflections are symmetric about the midpoint of the beam. As the load increases, and asymmetric cracking occurs, there is increased asymmetry in the beam response, with the maximum deflection shifting to the left of the center point. This observation is consistent with observed damage formation across the beam, where the first and largest cracks are on the left side as seen in Figure 5.11 and Figure 5.12.

Detection of Onset of Damage and Damage Propagation

There are several ways to use StereoDIC measurements to identify local damage. Two of the approaches, high local "pseudostrains" and local increases in correlation un-

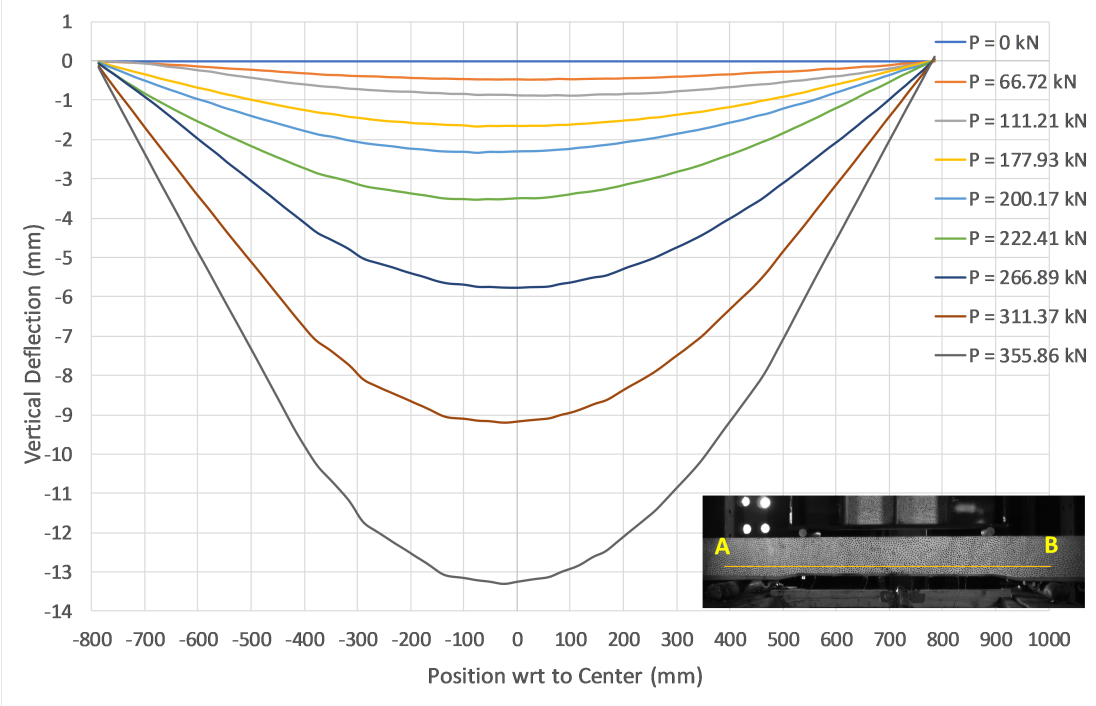


Figure 5.15: Vertical displacement of the beam centerline at different loads.

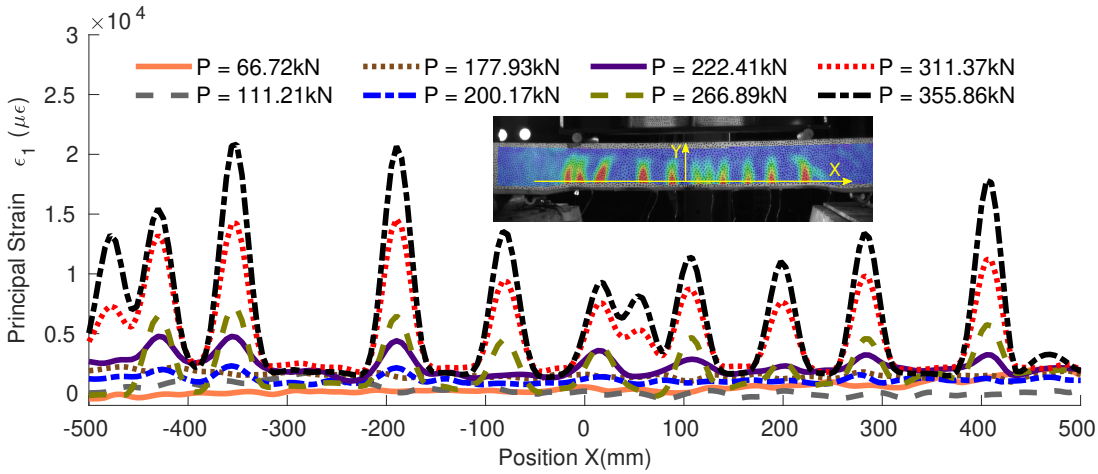


Figure 5.16: The principal strain along the longitudinal axis for each load hold. The inset shows the line where the measurements were extracted.

certainty, were highlighted in Section 5.4.1. Another approach is to monitor changes in principal strains during loading. A series of line plots are shown in Figure 5.16, with each line representing variation in the $\epsilon 1$ principal strain along line AB (shown in the inset of Figure 5.16) at each load hold. Tension cracking is first formed when

the measured tensile strain exceeds the cumulative cracking strain of concrete and the compressive strain induced by the prestressing of the beam. Average values of measurements at first cracking near the bottom surface for this specimen are $1,300\mu\epsilon$ [101].

Figure 5.16 highlights two of the curves, corresponding to $178kN$ and $200kN$ of load. The first one is relatively uniform and shows strain values close to the typical cracking strain for the specimen. It is also noted that all curves corresponding to lower loads are also relatively uniform with lower strain values. However, the second one exhibits strain peaks in the left segment of the specimen that are higher than the cracking strain and coincide with the location of cracks identified by visual inspection. The data for the higher load also exhibits peaks on the right side, though of much smaller amplitude. Visual inspection in this area showed that cracking has just reached the inspection line AB. It is observed that as the load increases the existing peaks increase, indicating that propagation of existing cracks is occurring as well as the formation of additional cracks along the bottom of the beam. The asymmetric distribution of cracks along the specimen length explains the slight asymmetry of the deflection curves after cracking shown in Figure 5.15.

Damage Detection During Dynamic Loading

DroneDIC measurements discussed in the preceding sections pertain to measurements acquired while the load is known and held constant, clearly demonstrating the versatility of the mobile system. In addition, the ability to capture the deformation fields while the specimen is under dynamic loading can be important in structural health assessment since it is possible to detect damage under "ambient" loading. In this study, the specimen was subjected to cyclic loading twice during the experiment, as discussed in Section 5.3.5 and shown in Figure 5.8. The first harmonic loading was designed to be in the region before damage occurred in order to compare the StereoDIC

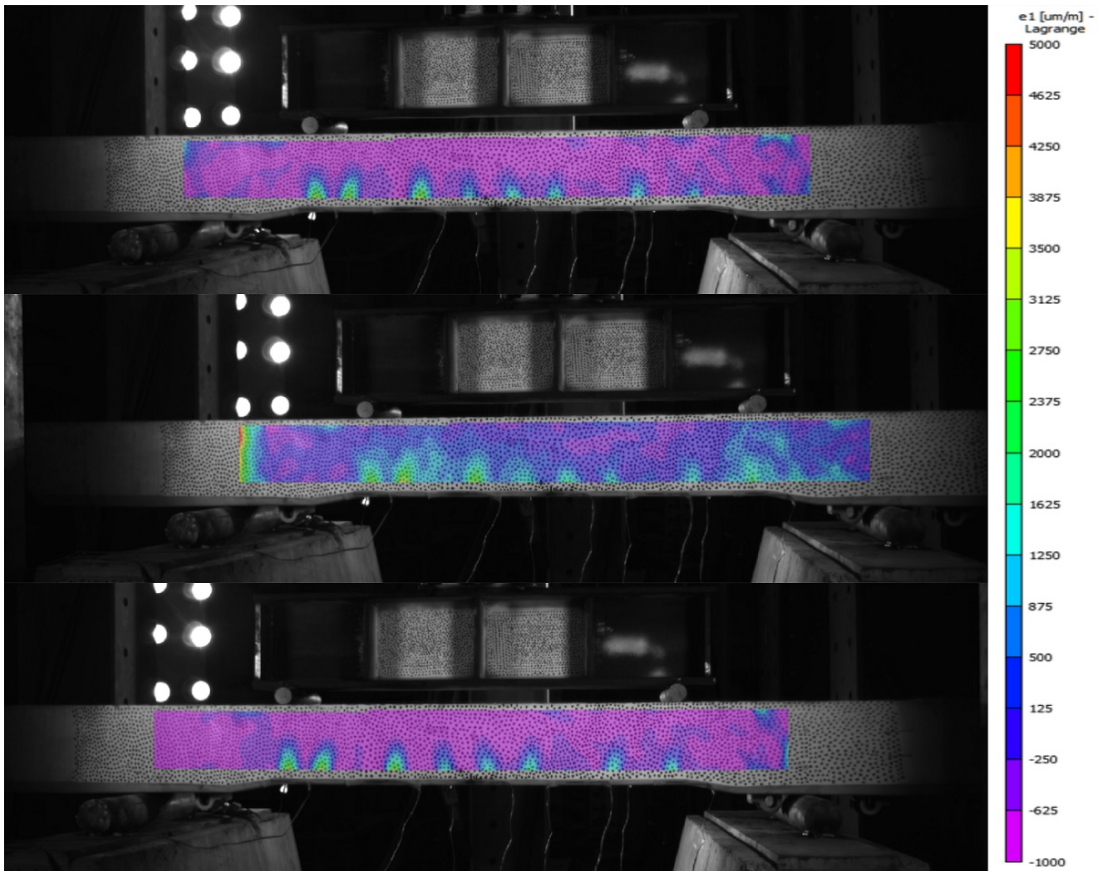


Figure 5.17: Principal strain field during harmonic loading at three random times.

data with the LVDT data while demonstrating the ability to measure the temporal characteristics of the loading. The second harmonic load was designed to be after the specimen is damaged to test whether the system is able to detect damage under dynamic loading. Figure 5.17 shows full-field data for the maximum principal strain data at three random times during the second harmonic load. While the strain amplitude varies among the three images, the pattern of the field in all three is the same and the locally high strains along the tensile side of the beam are visible indicators of damage. All three images exhibit eight cracks when the load oscillates $\pm$ 22.24kN about 222.4kN. This is consistent with the strain profile shown in Figure 5.17 for 222.4kN load. Thus, even with our prototype hardware limitations, damage is clearly detected during cyclic loading of the structure.

Chapter 6

Field Deployment of the StereoDIC enabled UAS

6.1 Introduction

The work presented in Chapter 5 aimed to show the feasibility of using a UAS equipped with a StereoDIC system as an inspection platform. Through lab experiments, it was shown that the system was able to detect the damage on the specimen and measure its deformation. The goal however is to deploy and test such a system on a outdoor infrastructure. Yet, the system used in the lab experiments is not suitable for field deployments since it has many limitations.

Several of the limiting factors are due to the UAS platform used. While it was appropriate for initial development since it is highly configurable and allows to easily mount different types of peripherals, it has a limited payload capacity. This led to using computers with restricted computational power, as well as using lower quality imaging sensors for the StereoDIC subsystem. Nevertheless, the payload capacity of the platform was reached, significantly limiting the flight time, range and maneuverability of the platform.

To safely and effectively operate in the field, the platform should have enough flight time so that it can collect all the desired data, and more importantly it should have enough thrust reserves to be able to operate in challenging weather conditions. Consequently, the prototype needs to be upgraded to a bigger platform. With a higher payload, all the peripherals and especially the StereoDIC subsystem can be

redesigned and upgraded.

Another challenge of field deployments is the changing and sometimes challenging weather conditions. To ensure that the area of interest will always be in the field of view of the StereoDIC cameras, a controller that can respond to challenging wind effects is needed. Such a controller design requires the system to measure the wind forces applied to the platform. Instead of adding more sensors that could directly measure the wind velocity and then calculate the applied forces, the state estimation framework was redesigned to estimate the effect of all external disturbances. The updated state estimation framework uses a nonlinear model of the UAS dynamics and all available position measurements to estimate any external forces applied to the system.

In this chapter, system updates are presented updates that address all the identified challenges and allow for the deployment of the updated platform in the field to take measurements from a structure under normal operation. The remainder of the Chapter is organized as follows. In Section 6.2, the updated system is presented with details about the most significant peripherals. In Section 6.3 the new state estimation and control framework is discussed while Section 6.4 contains details about the experimental setup and deployment types. Finally, Section 6.5 presents the results of this work.

6.2 System

The updated system, shown in Figure 6.1, is based on the Aurelia X6 Standard Long Endurance UAS with a flight time up to $45min$ and a payload capacity of $5kg$. The UAS is equipped with the CubePilot Blue FCU running the PX4 Autopilot firmware. ROS is used to establish communication between the FCU, onboard computer and other peripheral devices. A comparison of the most important parameters is shown in Table 6.1.

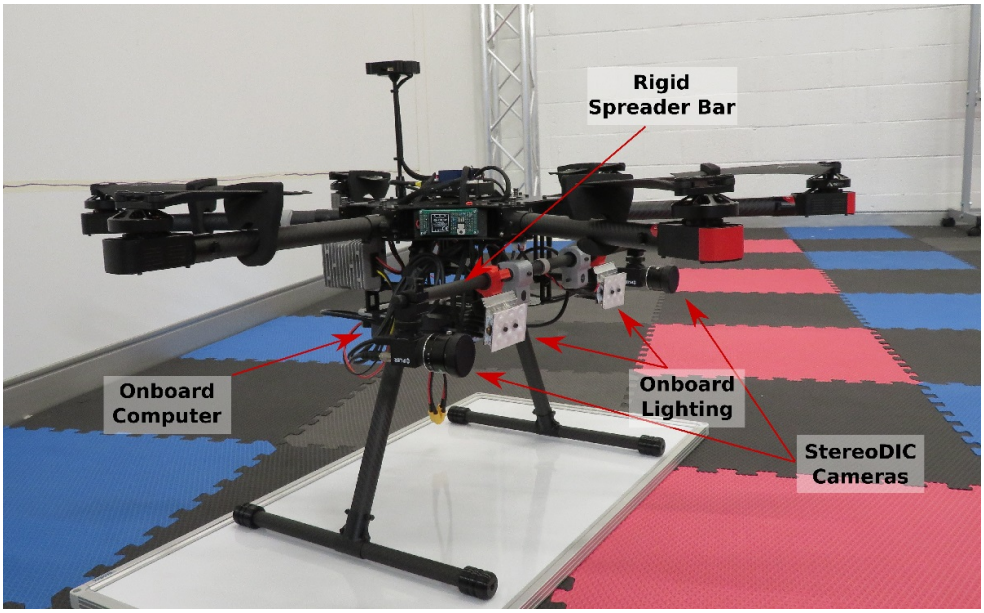


Figure 6.1: The updated UAS based on the Aurelia X6 Standard LE. The new cameras and lenses as well as the on board lighting system are visible.

While in the prototype, two, low-powered computers were used for the system autonomy and the StereoDIC, the increased payload capacity allows for a single computer with enough processing power. An Intel NUC i5 with $32GB$ of RAM and $1.5TB$ of storage space is used for all onboard processing needs. Using a single computer simplifies the overall system design and allows for the use of a single set of cameras for both the StereoDIC measurements and the marker tracking. The overall system architecture is shown in Figure 6.2.

6.2.1 StereoDIC System

The StereoDIC subsystem was also significantly upgraded. First, the cameras were upgraded to 8.9 Mpixel sensors with $42fps$ maximum frame rate and a global shutter. Moreover, $12mm$ lenses with adjustable focus and aperture were used. The aperture ranges from *f/1.4* to *f/16*. As mentioned in Section 5.2.2, it is critical to have a low exposure time to limit the effect of motion blurring in the images. However, in the field, the lighting conditions change with the weather and time of the day and year.

Table 6.1: Comparison of the prototype and updated systems.

Parameter	Prototype System	Updated System
	UAS platform	
Platform	DJI Matrice 100	Aurelia X6 Standard LE
Max Payload	1.17kg	5kg
Max Flight Time	22min	45min
FCU	DJI N1	CubePilot Blue
	Onboard Computing Power	
Computer	Intel Compute Stick DJI Manifold	Intel Nuc
Processor	Intel m3 (2 × 900MHz) NVidia Tegra K1	Intel i5 (4 × 2.3GHz)
RAM	4GB + 2GB	32GB
Storage	64GB + 16GB	1.5TB
	StereoDIC	
Resolution	1920 × 1200	4096 × 2160
Frame rate	5 fps	42 fps
Lens	Fixed, 12mm	Adjustable focus and aperture, 12mm
onboard Light	Non available	20W LED light
Synchronization	Software	Hardware

Having an adjustable aperture allows to control the amount of light captured by the sensor without having to change the exposure time. Moreover, $20W$ of onboard LED light was added to the platform to allow capturing data even in low light conditions. Finally, hardware triggering is used to ensure that the two cameras are synchronized. The FCU is used to generate the triggers for the cameras while it also generates a timestamp for each trigger instance. This means that the cameras are not only in sync with each other but with the avionics sensors as well.

While the system is able to record images up to $42fps$, the frame rate used in most deployments was $5fps$ due to the size of the required storage. Since the resolution of the cameras is 4096×2160 pixels, each raw image is around $8.85MB$. Saved as a lossless Tag Image File Format (TIFF), the file size is reduced to an average of $5.3MB$. At $5fps$ the required storage for a minute of data is $5.3MB \times 2cameras \times$

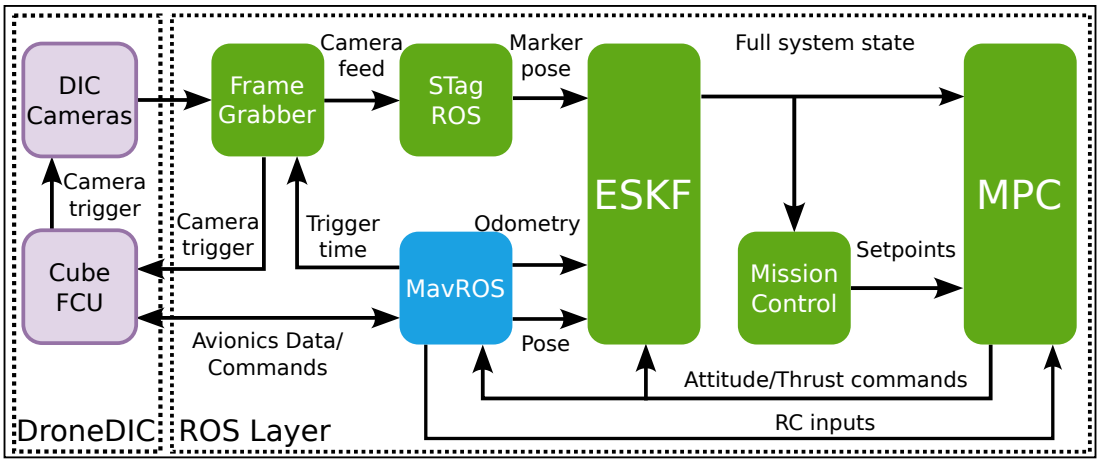


Figure 6.2: Block diagram of the system architecture.

$5fps \times 60sec = 3.18GB/min$. At full rate the required storage for a minute of data rises to $26.7GB/min$. The new platform can acquire data continuously for around $25min$. Therefore, at $5fps$, the total size of the data would be around $79GB$ while at $42fps$ the size would be around $667GB$. At maximum frame rate the full storage will have been used after just two full flights. Moreover, transferring these flights

Similar to the prototype, the StereoDIC cameras are mounted to an extra stiff carbon fiber rod that is attached to the UAS body using 3D printed mounts with rubber shock absorbers to limit the effect of motor and propeller vibrations on the cameras. Moreover, in the new platform the carbon rod is attached to the body of the drone close to the center instead of being attached at the arms. This moves the cameras out of the direct air flow generated by the propellers.

Since ROS is used as a middleware to establish communication between the different systems, the onboard computer was running on Ubuntu. As a result, the frame capturing software that was used in Chapter 5, could not be used for the updated version. A set of frame capturing ROS nodes were developed for this project. The first node is using software triggering and allows for quick configuration of the different camera parameters. The user should use this node to find the most appropriate camera setup for the experiment and lighting conditions. The second node does not

allow any changes to the initial configuration of the cameras but provides hardware triggering. This node should be used for data acquisition. Both nodes are made publicly available at `https://github.com/MikeK4y/spinnaker_driver_ros`.

6.3 State Estimation and Control

As discussed previously, there are a few challenges in deploying such a system. First, the GNSS localization is expected to be degraded close to large structures and then, weather conditions can make controlling the system challenging. While the state estimation framework presented in Chapter 5 addresses the first challenge, to effectively control the platform under challenging wind conditions, the state observer should also be able to estimate any external disturbances acting on the platform. Furthermore, an effective controller should consider the estimated disturbances and adapt its output accordingly. Both the state estimation and control frameworks were redesigned before the field deployments. The different portions of the updated control framework are developed as ROS nodes and the complete framework as well as scripts that implement the required system identification are made publicly available at `https://github.com/usrl-uofsc/px4_control`.

6.3.1 System modeling

A nonlinear model of the UAS is required by both the state estimator and the controller. To define the six Degree of Freedom (DoF) pose of the UAS, a fixed inertial frame (E) and a body frame (B), attached to the platform, need to be defined. Figure 6.3 shows a schematic of a quadrotor UAS with the body and inertial frames. While a quadrotor model is shown in the figure, the derived model for the dynamics can be used for multiple types of rigid body multirotors. Finally, the nonlinear model used

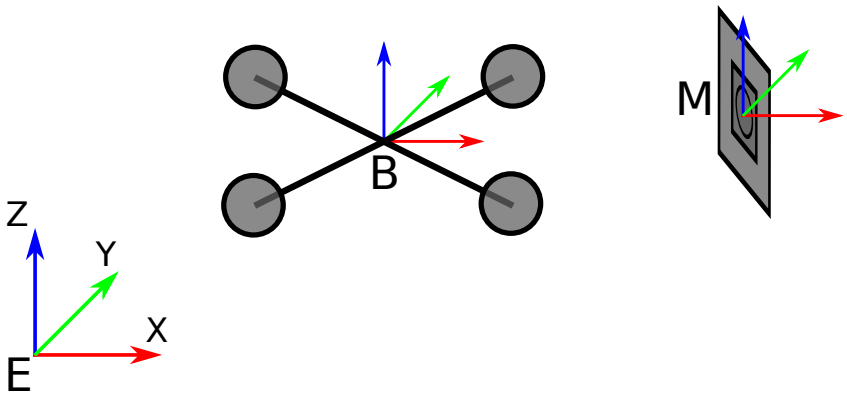


Figure 6.3: A model UAS with the attached body frame B, the fixed inertial frame E and a marker attached frame M.

for the UAS is:

$$\begin{aligned}
\dot{p} &= v \\
\dot{v} &= \begin{bmatrix} d_x & 0 & 0 \\ 0 & d_y & 0 \\ 0 & 0 & d_z \end{bmatrix} v + R(q) \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + F_{external} \\
\dot{q} &= \frac{1}{2} q \otimes \omega
\end{aligned} \tag{6.1}$$

where $p \in E$ is the vector of the position of the origin of B and $v \in E$ is the linear velocity of B. The rotation matrix of B with respect to E is denoted by R. d_x, d_y, d_z are the mass normalized drag coefficients, T is the mass normalized thrust, g is the gravitational acceleration and $F_{external}$ are the mass normalized external disturbances. The mass normalized thrust is modeled as $T = k_T T_{cmd}$ where k_T is the thrust coefficient and $T_{cmd} \in [0, 1]$ is the thrust command. Finally, q is the rotation quaternion equivalent of the rotation matrix R, $\otimes$ denotes the quaternion product and $\omega \in B$ is the angular velocity of B with respect to E written as a pure quaternion.

While quaternions are used for the state estimation, pitch (ϕ), roll (θ), and yaw (ψ) angles are used for the controller. Furthermore, it is assumed that a low level attitude controller can track the desired roll and pitch angles (ϕ_d and θ_d respectively). A first order model is used to model the response of the attitude controller for roll

and pitch. As for the yaw angle, it is assumed that the attitude controller can directly track the desired yaw rate $\dot{\psi}_d$. Finally:

$$\begin{aligned} \dot{\phi} &= \frac{1}{\tau_\phi}(k_\phi \cdot \phi_d - \phi) \\ \dot{\theta} &= \frac{1}{\tau_\theta}(k_\theta \cdot \theta_d - \theta) \\ \dot{\psi} &= \dot{\psi}_d \end{aligned} \tag{6.2}$$

with τ_ϕ, τ_θ the time constants and k_ϕ, k_θ the gain.

The inputs to the system are $T_{cmd}, \phi_d, \theta_d, \dot{\psi}_d$. All the platform specific parameters can be identified using flight data and classic system identification methods [59].

To improve the position estimation in GNSS degraded environments and to mark the area of interest, STag markers were used. As shown in Figure 6.3, the markers define a coordinate frame M. The frame M is assumed to be stationary with respect to the fixed frame E. As such its position and attitude are modeled as:

$$\begin{aligned} \dot{p}_M &= 0 \\ \dot{R}_M &= 0 \end{aligned} \tag{6.3}$$

where $p_M \in E$ is the vector of the position of the origin of M and the rotation matrix of M with respect to E is denoted by R_M.

6.3.2 State Estimation

The state and disturbance estimator developed for this work is based on an Error State Kalman Filter (ESKF). The structure and derivation of the filter is based on [89]. There are three state values considered in the ESKF formulation, the true state χ_{true}, the nominal state χ, and the error state $\delta\chi$. The true state is the composition of the nominal and error states $\chi_{true} = \chi \oplus \delta\chi$. The nominal state should be considered as the "large" part of the true state and it can be calculated by integrating the nonlinear model of the system. The error state however, should be considered as the "small" part of the true state that all modeling errors and process noise accumulates into.

The ESKF estimates the error state instead of the true state and uses it to correct the nominal state.

The ESKF process is as follows. Using the inputs to the system and the model derived in 6.3.1, the state of the system is tracked by integrating the nonlinear system. However, the modeling errors and unmodelled dynamics will gradually accumulate in the error state, so the filter is propagated to predict a Gaussian estimate of the error state. When a measurement becomes available, the error state becomes observable and a filter correction is performed. The error state mean can then be calculated and injected to the nominal state.

There are a few advantages in using an ESKF compared to other Kalman Filters [61, 89]. Compared to the EKF, the error state system operates closer to the origin and thus the linearization assumption is more likely to hold. Also, the error state is small and thus all higher order products are indeed negligible, simplifying the computation of Jacobians. Since the generalized composition is used to correct the nominal state and find the true state, it simplifies working with constrained quantities such as rotations in 3D. Finally, it has been shown that the ESKF formulation performs better for mobile robot localization than the EKF formulation[78].

The three state values considered in the filter are shown in Table 6.2. The angles vector contains the error state of the UAS attitude quaternion. As in Section 3.3, it is assumed that the GNSS position measurement contains both White and Brownian noise. To track the random bias of the Brownian noise an extra state b_{true} is added to the filter.

To derive the dynamics of the true state it is assumed that the modeling errors

Table 6.2: The ESKF variables.

	True	Nominal	Error	Composition
UAS position	p_{Dtrue}	p_D	δp_D	$p_{Dtrue} = p_D + \delta p_D$
UAS velocity	v_{Dtrue}	v_D	δv_D	$v_{Dtrue} = v_D + \delta v_D$
UAS attitude	q_{Dtrue}	q_D	δq_D	$q_{Dtrue} = q_D \otimes \delta q_D$
Angles vector			$\delta\theta$	$\delta q_D = e^{\delta\theta/2}$
Disturbances	F_{true}	F	δF	$F_{true} = F + \delta F$
Marker position	p_{Mtrue}	p_M	δp_M	$p_{Mtrue} = p_M + \delta p_M$
Marker attitude	q_{Mtrue}	q_M	δq_M	$q_{Mtrue} = q_M \otimes \delta q_M$
GNSS bias	b_{true}	b	δb	$b_{true} = b + \delta b$

are additive and affect directly the velocity and attitude of the UAS.

$$\begin{aligned}
\dot{p}_{Dtrue} &= v_{Dtrue} \\
\dot{v}_{Dtrue} &= Dv_{Dtrue} + RT - g + F_{true} + v_{me} \\
\dot{q}_{Dtrue} &= \frac{1}{2} q_{Dtrue} \otimes (\omega + \omega_{me}) \\
\dot{F}_{true} &= 0 \\
\dot{b}_{true} &= \upsilon \\
\dot{p}_{Mtrue} &= 0 \\
\dot{q}_{Mtrue} &= 0
\end{aligned} \tag{6.4}$$

where v_{me} and ω_{me} contain all modeling errors and υ is the zero mean Gaussian noise that drives the random bias. The nominal state is the same as the true without these three parameters. Consequently the error state dynamics are:

$$\begin{aligned}
\dot{\delta p}_D &= \delta v_D \\
\dot{\delta v}_D &= D\delta v_D + R[T]_x \delta\theta + \delta F + v_{me} \\
\dot{\delta\theta} &= -[\omega]_x \delta\theta \\
\dot{\delta F} &= 0 \\
\dot{\delta b} &= \upsilon \\
\dot{\delta p}_M &= 0 \\
\dot{\delta\theta}_M &= 0
\end{aligned} \tag{6.5}$$

where $[]_x$ denotes the skew symmetric matrix.

Three types of measurements are considered. A UAS pose measurement, a UAS odometry measurement and finally a relative pose measurement from the marker tracking. The first two measurements come from the avionics sensor fusion that runs on the FCU. The FCU is using data from the GNSS sensor, the IMU and magnetometer as well as any other available sensors to get an estimate of the drone's pose and velocities. These are then used as measurements for the ESKF.

The pose measurement contains two parts, the position and the attitude of the drone. For the position measurement, since it is based on GNSS measurements we assume that it contains the GNSS bias while the attitude is measured directly. As for the velocity of the UAS, the FCU reports the velocity in the body frame instead of the fixed frame. The first and second measurements are:

$$\begin{aligned} y_{1a} &= p_D + b \\ y_{1b} &= q_D \\ y_2 &= R(q_D)^T v_D \end{aligned} \tag{6.6}$$

When the marker is detected, the position and orientation of the marker relative to the camera are measured. Assuming that the camera has a known fixed position relative to the body frame the marker's pose relative to the UAS can instead be used. So the two part of the marker measurement are:

$$\begin{aligned} y_{3a} &= R(q_D)^T (p_M - p_D) \\ y_{3b} &= q_D^{-1} q_M \end{aligned} \tag{6.7}$$

As in Section 3.3, the χ^2 measurement gating is used for outlier rejection and covariance matching is used to adapt the measurement covariance matrices.

6.3.3 System Control

As shown in Figure 6.2, there are two levels of control. At the top, a mission controller is used that checks the overall system state and then provides suitable setpoints to

the position controller that is implemented as a Model Predictive Controller (MPC). The position controller then calculates the optimal attitude and thrust commands to reach the desired position and sends these commands to the FCU and the State Observer.

For the field deployments, it is assumed that the position of the inspection area is known. The position controller can then be used to reach the area, or a human pilot can fly the UAS there. However, due to the GNSS positioning drift, the actual position and the estimated position might be a few meters off. Consequently, instead of defining the desired inspection position relative to the fixed frame, it is defined relative to the marker's frame. The mission controller monitors the state and once the marker that flags the inspection area is detected, it calculates the desired inspection position in the fixed frame and pushes it to the position controller. While the filter assumes that the marker's position is stationary, the estimated position might change due to incoming measurements until it converges. The mission controller continuously monitors the marker's estimated position and if it changes significantly from the one used to generate the setpoint, a new setpoint is generated and sent to the position controller.

Position Controller

To increase the quality of the DIC results, the cameras need to be accurately positioned relative to the inspection area. While the desired position specified allows for some small errors in the position of the cameras, if they move outside of these bounds or the orientation changes too much, the whole or portions of the specimen will go out of the field of view. Moreover, in the field, the changing weather conditions are expected to affect the system's flight. An effective position controller should be able to keep the cameras in the desired position while being robust to any external disturbances. Finally, since the motion of the cameras can affect the quality of the images,

the control inputs need to result in smooth and slow changes in the UAS position and attitude.

A nonlinear model predictive controller (NMPC) was developed for this work. The controller design is based on [52] and the model used is the one defined in 6.3.1. Since the model used contains an estimation for the external disturbances, the controller can adapt to the changing weather conditions.

The acados toolkit [43, 44] was used to generate the solver for the nonlinear Optimal Control Problem (OCP). The acados toolkit can generate C code to solve OCPs efficiently and fast allowing running such a controller in an onboard computer. The optimization problem is given by:

$$\begin{aligned}
\min_{\chi,u} \quad & \int_0^T (\chi(t)-\chi_r(t))^T Q(\chi(t)-\chi_r(t)) + (u(t)-u_r(t))^T R(u(t)-u_r(t))dt \\
& + (\chi(T)-\chi_r(T))^T P(\chi(T)-\chi_r(T)) \\
\text{subject to} \quad & \dot{\chi} = f(\chi, u) \\
& u(t) \in \mathbb{U} \\
& \chi(0) = \chi(t_0)
\end{aligned} \tag{6.8}$$

where χ, u denote the UAS state and input respectively, χ_r, u_r the UAS state and input reference respectively. Q, R are the cost weights for the state and the input respectively, while P is the weight for the terminal state. $\dot{\chi} = f(\chi, u)$ denotes the state dynamics as they were defined in 6.3.1, $\mathbb{U}$ is the set of valid inputs and $\chi(t_0)$ is the initial state of the system. Finally, T is the controller's horizon.

The horizon of the controller is set at $3sec$ and the control rate at $20Hz$. At the beginning of each control loop, the current initial conditions and external disturbances estimates are updated. If a single setpoint is provided, then all the reference points are set to this single setpoint. However, the controller can also be used to track a specific trajectory and in that case, the appropriate reference points are selected. As for the input reference, the attitude inputs are all set to zero while the thrust input

is set to the expected thrust required for hovering $T_{hover} = g/k_T$.

It is worth noting here that while the MPC framework can be used to calculate the input for the yaw rate, there is an important drawback due to the discontinuities of the yaw angle. It is common to define the yaw angle as $\psi \in [-\pi, \pi)$. As such, when the UAS operates close to $-\pi$ or π and the yaw value changes enough to jump to the other side, the MPC instead of applying a small correction, will try to turn the UAS 2π to reach back to the desired value. To work around this issue, quaternions were initially used also for the MPC model. However, this complicated the OCP and thus more time was required to reach a solution, leading to a decreased control rate or a smaller horizon. The most common approach here, is to use another controller for the yaw, and use the input of this controller as a parameter for the MPC.

Another important issue is that there are cases, where the MPC cannot find a solution to the OCP resulting in not having a control input at the end of the control loop. While it is not common, provided that a reachable reference is given, it is advised to have a backup controller. A set of PID controllers are used as a backup in the case of MPC failure. Moreover, a warning message is displayed to inform that the OCP could not be solved.

6.4 Field deployments and experimental setup

6.4.1 Inspection Site Description

Figure 6.4: The Park Street Bridge used for the field deployments.

A railroad bridge was selected for assessing the system in field deployments. The bridge is positioned at Park St. Columbia SC, a short distance from the University

of South Carolina (Figure 6.4). Another advantage of the selected bridge is that it spans a large section of open land on one side, which makes the UAS deployment easier. No traffic control or other special measures are needed to fly in this area.

The selected bridge is a girder type bridge, built in 1991 with a total length of $123.7m$. There are four abutments supporting the bridge, splitting the total length in five separate sections. The first section on the northeast side spans $20.9m$ over an area with dense vegetation. The following section is the longest at $26.9m$ and it spans over Park St. The last three sections are around $25.3m$ each and span over the open area. The clearance for the first two sections is around $4.9m$ while over the open land it is $5.8m$. The superstructure of all five units is deck welded steel.

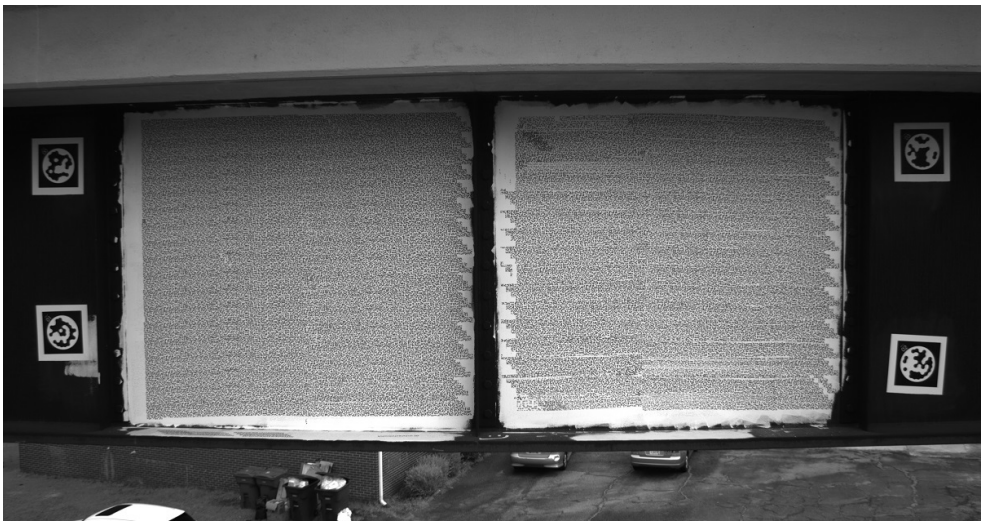


Figure 6.5: The inspection area on the southwest side of the bridge as captured by the StereoDIC cameras.

The middle span was selected for the field experiments. The full length of the girder is split into 15 sections of unequal length. On both sides of the bridge, the panel at the midspan with one more on its west side were selected as the area of interest. The height of the panels is $1.2m$ while each panel is around $1.8m$ in length totaling an area of $3.6m \times 1.2m$. Four STag markers were placed at the four corners outside of the area of interest. Each marker is $0.2m \times 0.2m$ with an offset of $0.2m$ between the marker and the speckled area, resulting in an area of $4.4m \times 1.2m$. To

ensure that both cameras can view the full area, using the imaging sensor size and lens focal length, the cameras need to be $4.2m$ away. At this distance, the field of view is $4.48m \times 2.36m$. An image of the area of interest acquired by the DIC cameras while the system is in operation can be seen in Figure 6.5.

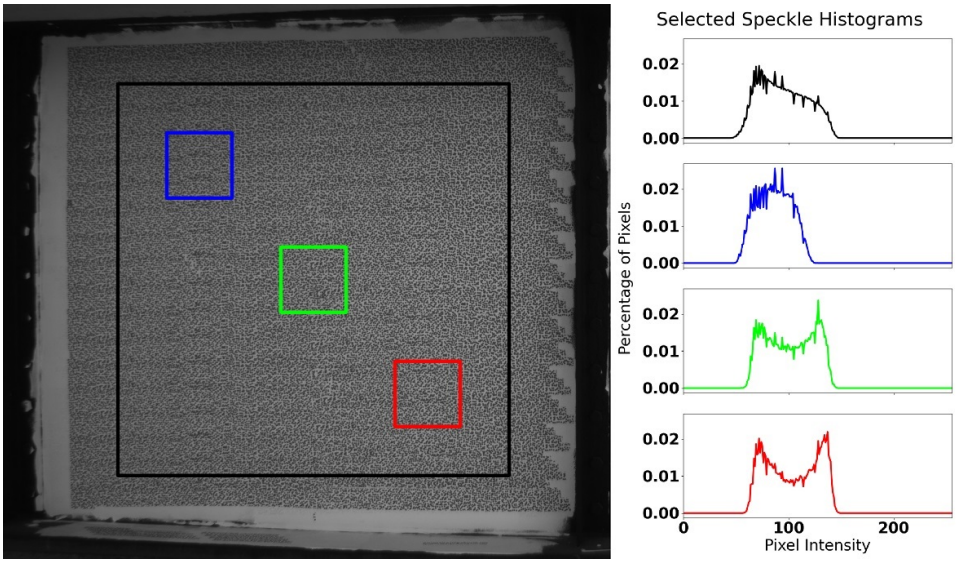


Figure 6.6: Detail of the inspection area with intensity histograms on various points that show the acquired contrast from inkjet printing.

The average pixel magnification at this distance is expected to be $1.1mm/pixel$. With the expected pixel magnification, to ensure that each speckle will cover a 3×3 pixel area in the image [11], the speckle size should be at least $3.3mm$. To allow some room for the expected motion, the speckle size was selected to be $5mm$. A number of different methods were explored for the most effective speckle applications in such structures and environments. It was finally concluded that using a hand held inkjet printer to print black speckles on a white background resulted in a high contrast pattern with a quick application time. Figure 6.6 shows half of the inspection area along with histograms of the overall region as well as over multiple smaller regions. Close to the center of the image (region in red), where the cameras have the best focus, the histogram has a bimodal distribution which is ideal for DIC.

6.4.2 Deployments

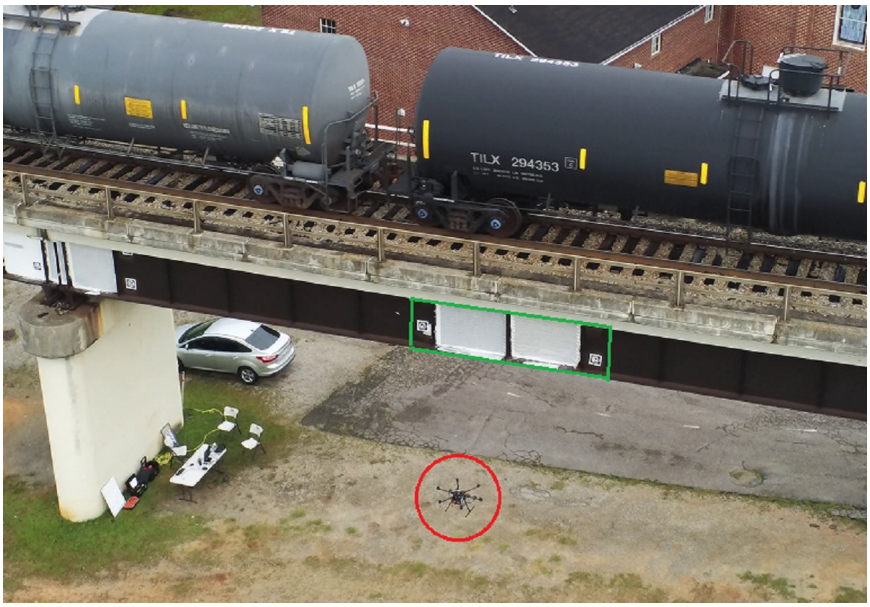


Figure 6.7: The system during a deployment at Park Street Bridge. The UAS is highlighted with a red circle and the inspection area is highlighted with a green rectangle.

The system presented was deployed multiple times at the inspection site during a span of eight months. The system can be seen in Figure 6.7 in one of the field deployments while a train is passing and loading the bridge.

During the speckle pattern application investigation, the system was deployed to acquire images from the different patterns tested. These images were later processed to assess the quality of the pattern and guide the investigation. In these deployments, the platform was remotely operated by a human pilot.

When the speckle pattern was finalized, and the STag markers that flag the area were placed, then the system was properly deployed to acquire data. During these deployments, a safety pilot would control the platform close to the area of interest, so that a visual tether could be established. Once any of the markers is detected, the platform can autonomously fly to the desired position and then hover in place until the required data have been collected. Once the data acquisition process is

completed, the safety pilot can take control of the UAS to return and land.

Two types of deployments were performed. First, with the bridge without any load, and second while a train was passing over the bridge, loading it. The first type of experiments was performed to gather images that could be used as a reference for the default state of the bridge as well as to assess the quality of the acquired measurements. For the second type of experiments, the system would be deployed a few minutes before the train reached the bridge and would start acquiring images. The system would continue to acquire images as long as the train was over the bridge provided that there was enough power to return and land safely. If the train passed and there was enough power to continue acquiring images, the system would do so for a few more minutes. Doing so would allow to have a reference for the unloaded bridge, the full loading sequence by the passing train and finally any transient response of the bridge after the passing of the train before returning to its default state. Finally, before and after each deployment, a DIC calibration was performed. The calibration data were used to ensure that the imaging system was not affected during the deployment by checking that there was no significant difference in the relative position between the two cameras.

6.5 Results

6.5.1 State Estimation and Controller Lab Results

The proposed state observer and position controller were tested first in lab experiments. Due to the size of the UAS platform used for this work, it was not considered safe to use it for the indoor experiments. However, the developed frameworks can be used with multiple rigid body multirotors that have a PX4 FCU and a calibrated camera. The only changes required is to modify the nonlinear model parameters, alter the set of valid inputs and update the sensor covariances.

A small hexrotor UAS was used for the in lab experiments. The platform was

equipped with an Odroid X4U onboard computer and a Logitech C270 Webcam. It is worth noticing here, that the both the onboard computer and camera are inferior to the ones on the main platform. The cameras used in the main platform have a higher resolution, a global shutter and are in sync with the avionics sensors. All these parameters can significantly affect the results of the state estimation and the response of the controller. Finally, the UAS used designed for deploying sensors to the underside of structures, so the camera is pointing towards the ceiling.

To emulate outdoor conditions inside the lab, first simulated GNSS position data were generated and then a drum fan was used to generate wind disturbances. To generate virtual GNSS position data, White and Brownian noise were injected to the MoCap position data. With the additive White noise the accuracy of the MoCap measurements is significantly reduced and with the Brownian noise a random walk bias is introduced to the measurements. At time step k:

$$\begin{aligned} p^k_{GNSS} &= p^k_{MoCap} + b^k + \omega \\ b^k &= b^{k-1} + \upsilon \end{aligned} \tag{6.9}$$

where p^k_{GNSS} is the virtual GNSS position used in the observer, p^k_{MoCap} is the UAS position from the MoCap, $\omega \sim \mathcal{N}(0, 10^{-3})$ is the additive White noise and b^k is the Brownian noise that is driven by $\upsilon \sim \mathcal{N}(0, 10^{-3})$. Figure 6.8 shows the generated GNSS position, the estimated position from the observer, the real position of the system provided by the MoCap and the desired hovering position. As it can be seen, while significant noise is introduced to the system, as long as the marker is visible, the state estimation is able to track the real position with high accuracy. Moreover, the controller tracks the desired position with small errors.

For the experiment shown, initially there were no wind disturbances generated. However, while the UAS was hovering in place, the fan was switched on to its maximum speed. The fan was pointing towards the positive x-axis and it was turned on for $30sec$ before being turned off again. Figure 6.9, shows the estimated disturbances

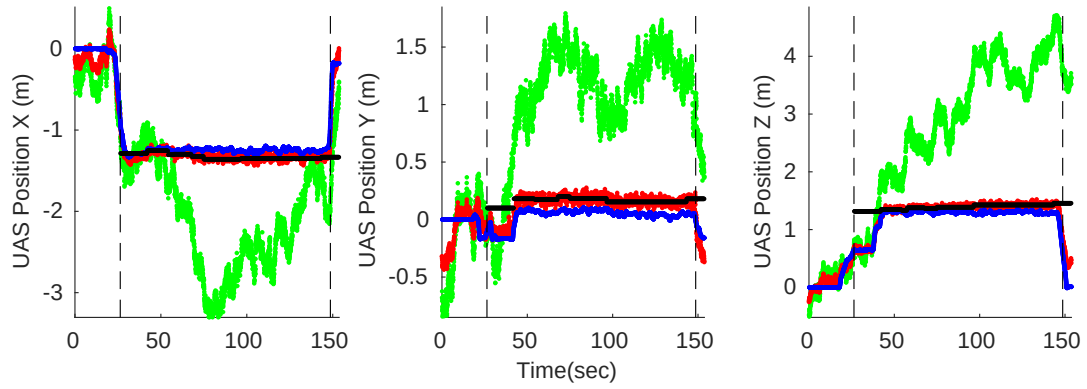


Figure 6.8: The UAS position during lab experiments. The green curve shows the generated GNSS data, the red shows the estimated position from the developed state observer and the blue is the UAS position according to the MoCap data. The black curve shows the desired position and the dashed vertical lines show the portion of the flight when the marker is visible.

for during the autonomous portion of the experiment. The portion between the dashed vertical lines is when the fun was on and a positive disturbance is registered as expected.

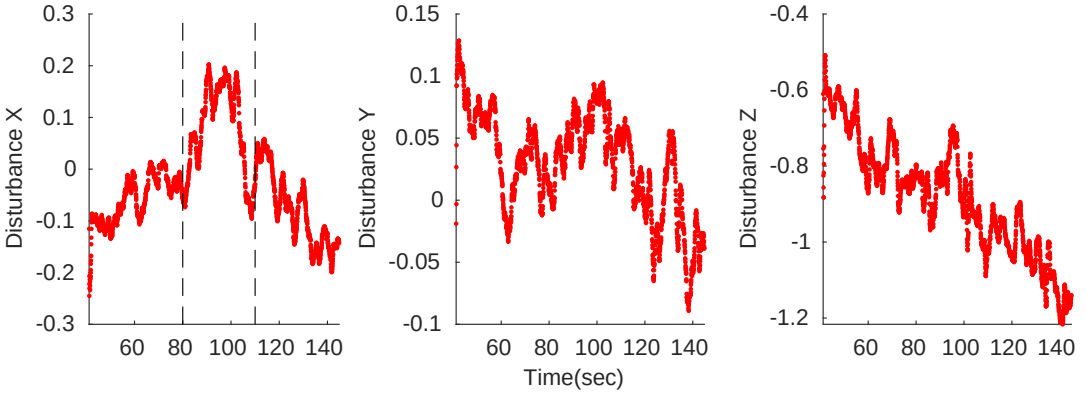


Figure 6.9: The estimated disturbances during the lab deployment. The dashed vertical lines at the x-axis disturbances show the portion of the time while the fan was on.

6.5.2 State Estimation and Controller Field Results

Figure 6.10 shows the position of the platform during one field deployment. The blue curve shows the position of the platform according to the GNSS and other avionics

sensors and the red curve shows the estimated position of the platform according to the developed state estimation framework. The black curve is the desired position and the vertical dashed lines show the portion of the flight that was autonomous. The drift of the GNSS position which in a few cases was over a meter is visible in the figure. However, the developed state estimation framework uses the model projections and the extra information from the marker to get a better estimation of the real position of the platform while tracking the GNSS random walk bias.

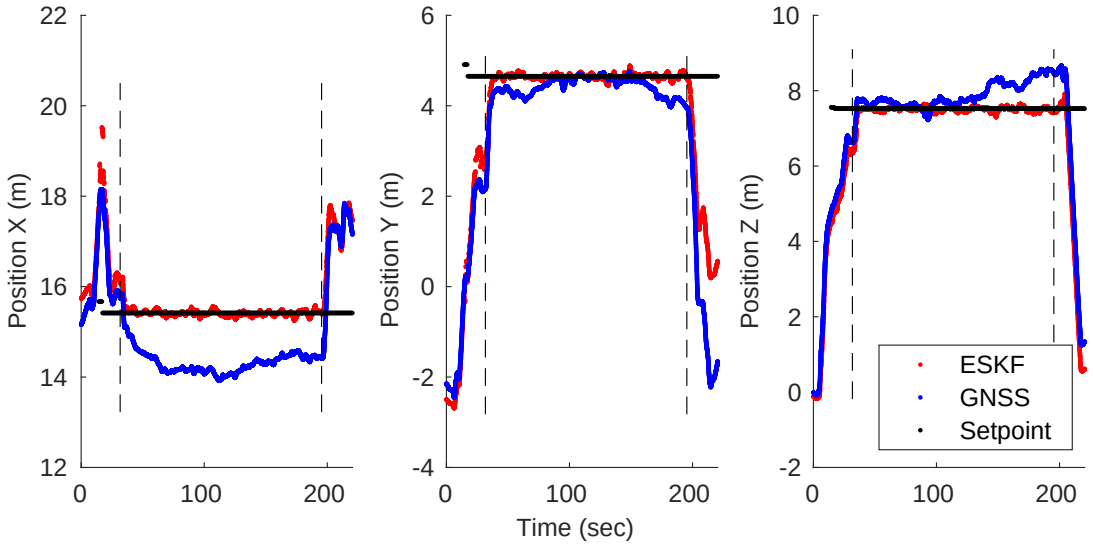
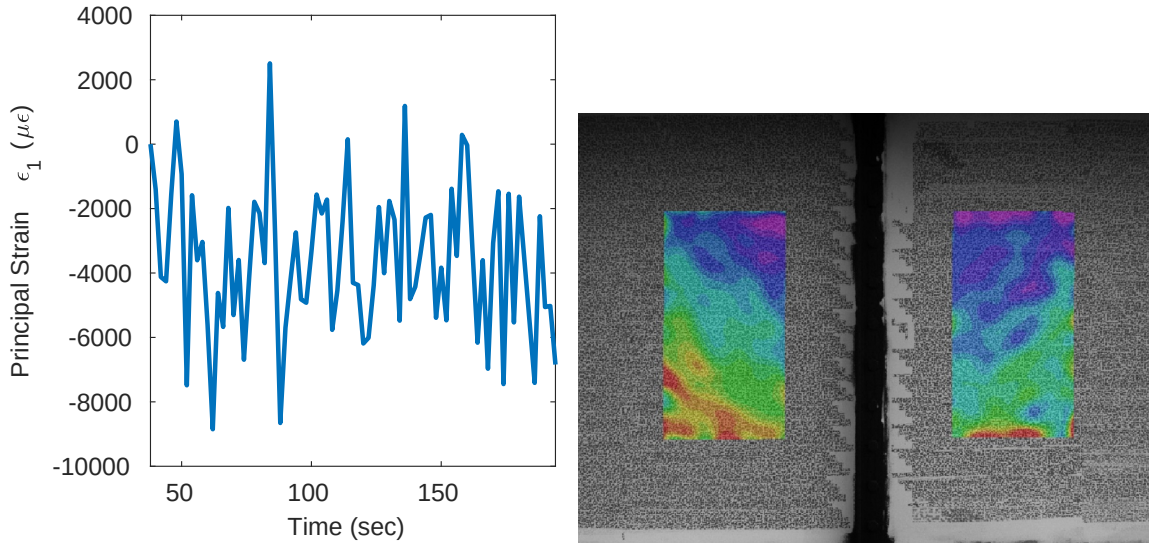


Figure 6.10: The position of the platform during one of the field deployments. The red curve shows the estimated position from the developed state observer while the blue is the reported one from the avionics sensors. The black curve shows the desired position and the dashed vertical lines show the portion of the autonomous flight.

As for the developed controller, Figure 6.10 shows that it is able to keep the system close to the desired state. During the deployment shown in Figure 6.10, there were significant side winds, nevertheless, the controller was able to keep the platform in the desired position with an RMS error of $12cm$ on the x-axis, $23cm$ on the y-axis and $16cm$ on the z-axis. On the same day, the RMS error over five deployments with different wind conditions was $13cm$ on the x-axis, $15cm$ on the y-axis and $22cm$ on the z-axis.

6.5.3 Inspection Results

The DIC images were processed using VIC-3D®. A subset of 49×49 pixel was used with a 7 pixel step size. All the components of the Lagrangian large strain tensor are calculated using a 15×15 filter. Again, since image averaging is impossible due to the motion of the cameras relative to the specimen, each image is processed individually.



(a) (b)

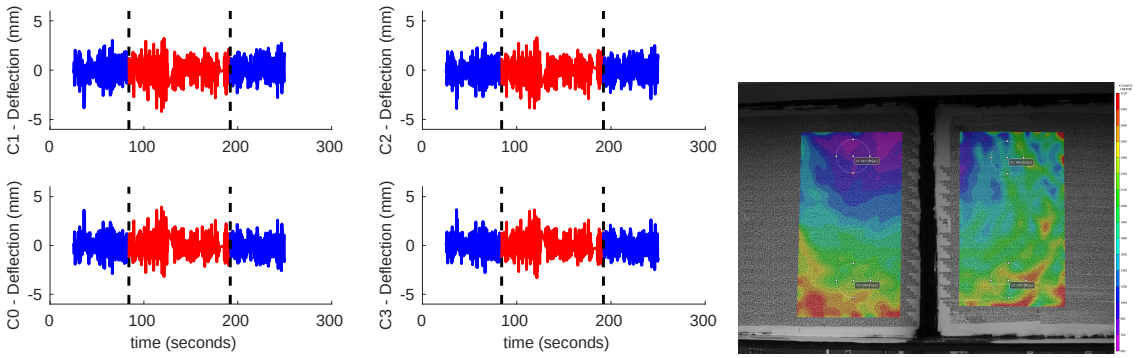
Figure 6.11: The principal strain average shown in (a) over the two highlighted areas at the center of the inspection region shown in (b). The results are sub-sampled to $0.5Hz$ for clarity.

As described in 6.4.2, the system was initially deployed to acquire data from the inspection site while the bridge was not being used. This type of experiments was useful to test the expected quality of the results. Figure 6.11 shows principal strain data averaged over the highlighted areas from one such experiment. While the images were captured at $5fps$ the data shown are sub-sampled to $0.5fps$ for better clarity. The results show strain levels up to a few thousand microstrains. This is much higher than the expected accuracy of the DIC system. Moreover, the inspected structure is expected to deform at most around $1200\mu\epsilon$ before critical failure.

Similar noise levels were also recorded for deployments where a train was passing.

Since the noise levels are much higher than the maximum expected deformation, no difference could be noticed between the state of the bridge with a train loading it and while it was unloaded. As it is known, there is a significant impact on strain measurements due to motion blur [8], the images where individually checked and only the sharpest ones were used. However, even in that case, the strain measurements remained at the same levels.

As for deflection results, Figure 6.12 shows vertical deflection measurements from a deployment with the bridge under loading. The red region of the plot shows the portion of the deployment with the train over the bridge, while the blue regions are from before and after. Again, no noticeable difference can be seen between the two regions. In both cases, the recorded response appears random while the amplitude of the measured deflection is at the same levels.



(a) (b)

Figure 6.12: Deflection results while the bridge is being loaded by a passing train. The region in red is while the train is on the bridge and the regions in blue are before and after the train. The four points (C0, C1, C2, C3) where deflection is measured can be seen in (b).

Noise Investigation

Due to the high noise levels recorded in the field measurements, an extensive investigation to identify the source of the noise was undertaken. First, as mentioned in 6.4.2, the calibration data captured before and after each experiment were used to

ensure that there was no change in the relative position of the imaging sensors. In all cases, no significant difference was noted.

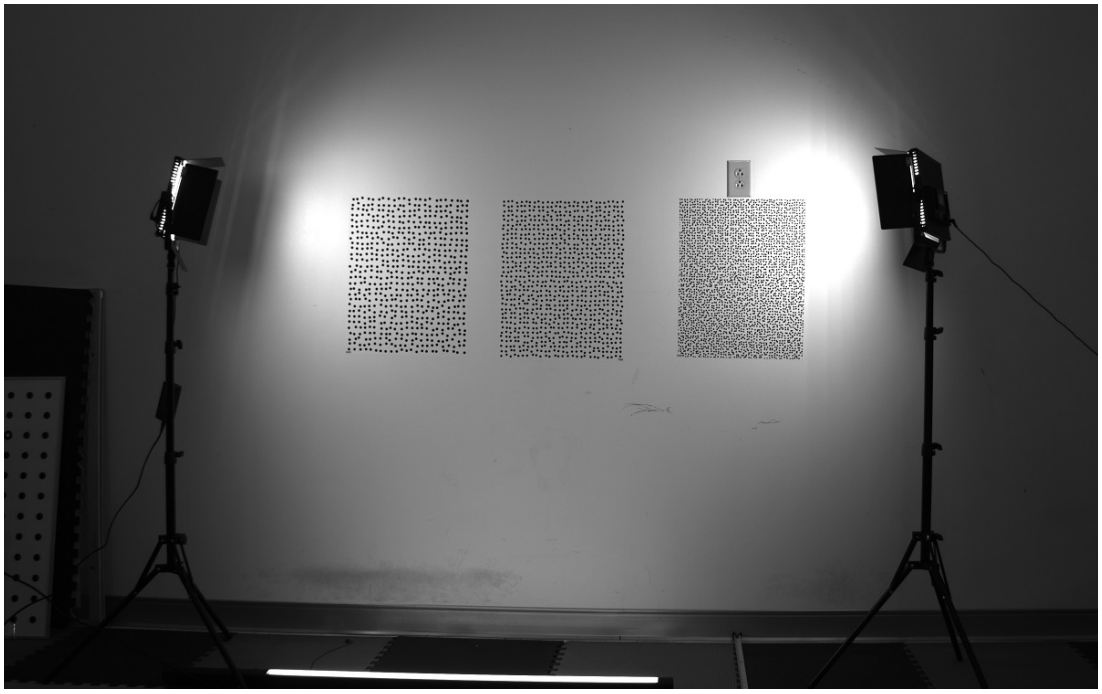


Figure 6.13: The experimental setup during the noise investigations.

A series of experiments was then performed in the lab. As shown in Figure 6.13 three separate speckle patterns with different speckle sizes were used. Then, the system was initially put on a platform at the same distance that it was used in the field experiments. While stationary, it recorded a series of 500 image pairs. Then, the system was deployed in an autonomous flight at the same position and another series of 500 image pairs were captured. Both captures were at the same rate and exposure time used in the field. Finally, additional lighting was used to reach similar illumination levels as the ones on the field.

The three speckle patterns were then processed separately, using the same subset and step size. The principal strain results are shown in Figure 6.14. The results show that the measured strain levels for the stationary system are at the expected levels for the hardware and setup used. However, the in-flight results are an order of magnitude higher. Another important finding is that while the data from the three

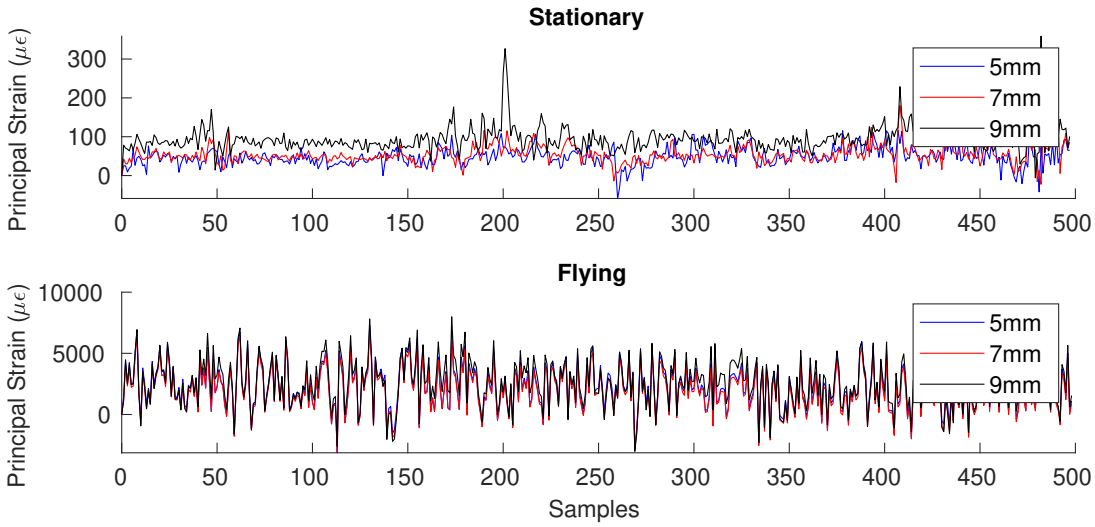


Figure 6.14: Principal strain results from the system during lab tests while stationary and in flight.

speckles are uncorrelated for the stationary system, for the in-flight data, the three series are highly correlated. This strongly indicates that there is some relative motion between the cameras while the system is in flight.

Figure 6.15: The platform with the vibration isolation mounts. A close up of the new mounts is shown in the bottom right corner.

Trying to pinpoint the source for this relative motion, a set of experiments were performed changing one critical parameter each time. The parameters altered were the mounting position, the length of the carbon rod, the size of the lenses and the

mounting type. Redesigning the part that was used to attach the carbon rod to the body of the UAS was the only change that produced a significant difference. The new part uses vibration isolation connectors to further reduce the effect of any vibrations produced from the UAS. The updated design is shown in Figure 6.15. The results from the system in flight with the new mounts are show in Figure 6.16. The principal strain levels were reduced to around one third when compared to the previous mount. However, the noise levels remain too high to allow proper inspections.

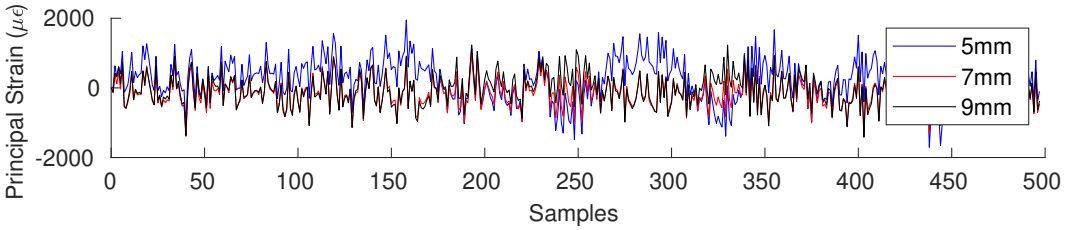


Figure 6.16: Principal strain results from the system in flight with the vibration isolation mounts.

The results obtained from these experiments indicate that the source of noise is the relative motion between the cameras during the data acquisition process. To measure the actual motion between the cameras, the carbon rod and the cameras were speckled so that the DIC method can be used. A set of high speed high resolution cameras was used to capture images of the platform at a rate of $3000Hz$. The platform and the experimental setup can be seen in Figure 6.17.

The objective of this experiment was to measure the relative motion between the cameras while the system is in flight. However, due to safety issues, the UAS could not be airborne. Trying to emulate the in-flight conditions, the system was placed on a table and the thrust was incrementally increased until it reached the take-off levels. In this case, while the system is still in contact with the table, there is significant down-wash and vibrations from the propellers.

The results of this test showed that during flight, the relative orientation between the cameras changes up to 0.3 degrees as shown in Figure 6.18. This motion continu-

(a) (b)

Figure 6.17: The platform with the cameras and camera mounting system speckled (a). The experimental setup used to record the expected relative motion between the cameras (b).

ously changes the stereo calibration between the cameras. Consequently, even though there is no deformation of the speckled area, it appears as there is. This is captured as false deflections and deformations explaining the high noise levels present in the data.

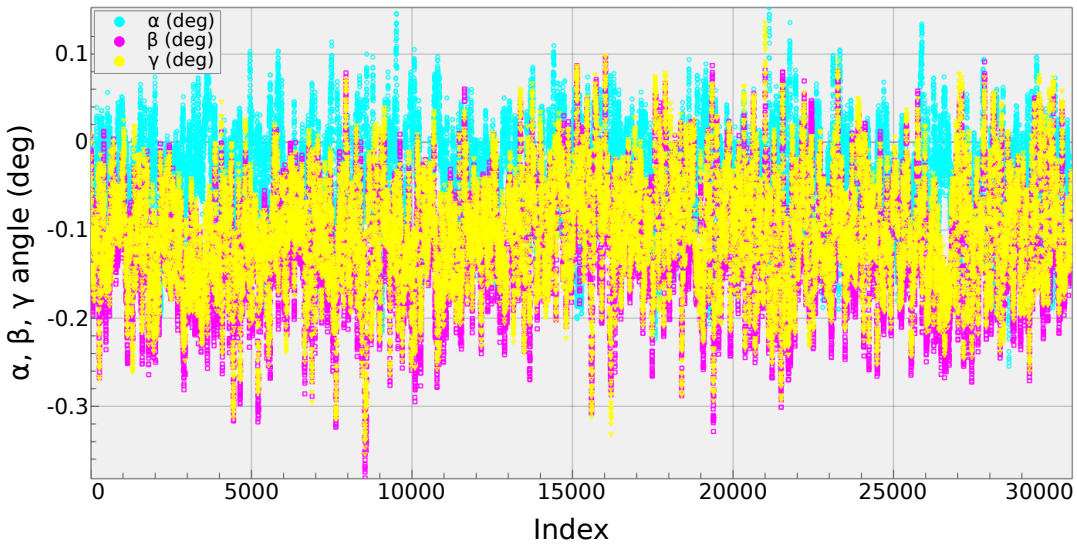


Figure 6.18: Change in the relative angle between the two cameras of the StereoDIC system.

Chapter 7

Conclusions

In this dissertation, the problem of autonomous, quantitative UAS inspections has been investigated. The main objective of this work was to improve the reach and safety of drone inspections while pushing towards quantitative results. Solutions to the main existing challenges were proposed, and experimentally tested.

In Chapter 3 we presented a heterogeneous marsupial robot as a solution to the limited range and flight time of UASs. Pairing an autonomous surface vehicle with a UAS, the range and operation time of the overall system was significantly increased. A cooperative localization framework and an effective flight controller allowed the reliable separation and docking of the two platforms. The proposed system was deployed in multiple inspections of structures close to water bodies. The mapping of a dock and a 3D reconstruction of a bridge are presented as sample inspection results. While the system was successfully deployed over a six-month period, there are still some issues limiting the fully autonomous deployments. Safety features were implemented to allow the system to recover from emergency situations. However, in case the safety features also failed, a safety pilot on site had to take over and take control of the two platforms. Also, no obstacle avoidance was implemented, increasing the probability of a collision.

Chapter 4, presents an extensive comparison of fiducial marker tracking for pose estimation. Fiducial markers were used throughout this dissertation to enable the autonomous platforms to operate safely even on GNSS denied or degraded environments. The state of the art and the most commonly used fiducial markers were tested

in multiple scenarios. Their performance was assessed under different distances, orientations, lighting conditions, imaging sensors and blurring effects. Moreover, their computational efficiency was recorded. The results of this study can be used to make an informed decision when selecting a fiducial marker for pose estimation.

In Chapters 5 and 6 a StereoDIC enabled UAS was developed to allow strain and deformation measurements of structures. In Chapter 5 a prototype of the system was initially developed and tested in lab experiments to validate its effectiveness. An experimental setup was designed that used a pre-stressed concrete beam subject to four-point bending. A stationary StereoDIC and LVDT sensor data were used to validate the results of the UAS deployed system. During the loading sequence, the UAS was autonomously hovering in the desired location using pose data from a fiducial marker and optical flow sensors. The results of this experiment showed that the prototype was able to accurately measure displacements as well as to identify the onset and progression of damage.

Chapter 6, builds on the results and identified limitations of the prototype system and a new platform suitable for field deployments is developed. First, the upgraded system is based on a bigger platform, allowing more flight time and the use of better hardware for the StereoDIC subsystem. More importantly though, the state estimation and control of the platform are redesigned to ensure the safe field deployments of the system. The updated state observer can estimate the effect of external disturbances on the platform. This estimate is then used by the position controller, making the system able to adapt to the weather conditions present on the site. A railroad bridge was selected as the deployment site and the system was successfully deployed there over an eight-month period. The new control framework allowed the autonomous deployment of the system even under challenging wind conditions. However, the inspection results showed increased noise levels, making the measurement of bridge's response impossible. Extensive experimental studies were then performed to

find the source of the noise. The studies showed a change in the relative orientation of the cameras up to 0.3 degrees that explains the noise levels. Building a camera mount that provides enough rigidity should lower the noise levels in the measurements allowing the system to measure the response of the bridge under load.

Bibliography

[1] *A Comprehensive Assessment of America's Infrastructure*. American Society of Civil Engineers, 2021. URL: www.infrastructurereportcard.org.

[2] Aysha Alharam et al. "Real Time AI-Based Pipeline Inspection using Drone for Oil and Gas Industries in Bahrain". In: IEEE, Dec. 2020. ISBN: 978-1-7281-9673-2. DOI: 10.1109/3ICT51146.2020.9312021.

[3] M.O. Anderson, M.D. McKay, and B.S. Richardson. "Multirobot automated indoor floor characterization team". In: *Proceedings of IEEE International Conference on Robotics and Automation*. Vol. 2. Minneapolis, MN, USA, 1996, pp. 1750–1753. ISBN: 0-7803-2988-0. DOI: 10.1109/ROBOT.1996.506965.

[4] Oualid Araar, Nabil Aouf, and Ivan Vitanov. "Vision Based Autonomous Landing of Multirotor UAV on Moving Platform". In: 85.2 (Aug. 2016), pp. 369–384. DOI: 10.1007/s10846-016-0399-z.

[5] *Autonomous Drones for Inspection*. URL: https://www.skydio.com/inspection (visited on 11/06/2021).

[6] Andrej Babinec et al. "Visual Localization of Mobile Robot Using Artificial Markers". In: *Procedia Engineering* 96 (2014), pp. 1–9. DOI: 10.1016/j.proeng.2014.12.091.

[7] R. Balcaen et al. "Impact of motion blur on stereo-digital image correlation with the focus on a drone-carried stereo rig". In: 55.1 (Oct. 2018), e12300. DOI: 10.1111/str.12300.

[8] R. Balcaen et al. "Influence of Camera Rotation on Stereo-DIC and Compensation Methods". In: *Experimental Mechanics* 58 (7 Sept. 2018), pp. 1101–1114. ISSN: 17412765. DOI: 10.1007/s11340-017-0368-x.

[9] Milan Banić et al. "Intelligent Machine Vision Based Railway Infrastructure Inspection and Monitoring using UAV". In: *Facta Universitatis, Series: Mechanical Engineering* 17 (3 Nov. 2019). ISSN: 2335-0164. DOI: 10.22190/FUME190507041B.

[10] Burak Benligiray, Cihan Topal, and Cuneyt Akinlar. "STag: A stable fiducial marker system". In: *Image and Vision Computing* 89 (Sept. 2019), pp. 158–169. DOI: 10.1016/j.imavis.2019.06.007.

[11] Rory Bigger et al. *A Good Practices Guide for Digital Image Correlation*. International Digital Image Correlation Society, Oct. 2018. DOI: 10.32720/idics/gpg.ed1/print.format.

[12] Christian Brommer et al. "Long-Duration Autonomy for Small Rotorcraft UAS Including Recharging". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Madrid, Spain, Oct. 2018, pp. 7252–7258. ISBN: 978-1-5386-8094-0. DOI: 10.1109/IROS.2018.8594111.

[13] Matthew Brown and David G Lowe. "Automatic Panoramic Image Stitching using Invariant Features". In: *International Journal of Computer Vision* 74.1 (Apr. 2007), pp. 59–73. ISSN: 0920-5691. DOI: 10.1007/s11263-006-0002-3.

[14] F. Caballero et al. "A visual odometer without 3D reconstruction for aerial vehicles. Applications to building inspection". In: IEEE, 2005. ISBN: 0-7803-8914-X. DOI: 10.1109/ROBOT.2005.1570841.

[15] Cesar Cadena et al. "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age". In: *IEEE Transactions on Robotics* 32.6 (Dec. 2016), pp. 1309–1332. DOI: 10.1109/tro.2016.2624754.

[16] Sabrina Carroll, Michail Kalaitzakis, and Nikolaos Vitzilaios. "UAS Sensor Deployment and Retrieval to the Underside of Structures". In: *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*. Athens, Greece: IEEE, June 2021. DOI: 10.1109/icuas51884.2021.9476737.

[17] Sean Catt et al. "Development of a Semi-autonomous Drone for Structural Health Monitoring of Structures Using Digital Image Correlation (DIC)". In: Springer International Publishing, May 2018, pp. 49–57. DOI: 10.1007/978-3-319-74476-6_7.

[18] Georges Chahine and Cedric Pradalier. "Survey of Monocular SLAM Algorithms in Natural Environments". In: *2018 15th Conference on Computer and Robot Vision (CRV)*. Toronto, ON, Canada, May 2018, pp. 345–352. ISBN: 978-1-5386-6481-0. DOI: 10.1109/CRV.2018.00055.

[19] Andrew Chambers et al. "Robust multi-sensor fusion for micro aerial vehicle navigation in GPS-degraded/denied environments". In: *2014 American Control Conference*. Portland, OR, USA, June 2014, pp. 1892–1899. ISBN: 978-1-4799-3274-0. DOI: 10.1109/ACC.2014.6859341.

[20] Gongfa Chen et al. “Homography-based measurement of bridge vibration using UAV and DIC method”. In: *Measurement* 170 (Jan. 2021), p. 108683. ISSN: 02632241. DOI: 10.1016/j.measurement.2020.108683.

[21] Antonio C. B. Chiella, Bruno O. S. Teixeira, and Guilherme A. S. Pereira. “State Estimation for Aerial Vehicles in Forest Environments”. In: *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*. Atlanta, GA, USA, June 2019, pp. 890–898. ISBN: 978-1-7281-0333-4. DOI: 10.1109/ICUAS.2019.8797822.

[22] Jeffrey Delmerico and Davide Scaramuzza. “A Benchmark Comparison of Monocular Visual-Inertial Odometry Algorithms for Flying Robots”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2018. DOI: 10.1109/icra.2018.8460664.

[23] J.R. Martinez de Dios, A. Ollero, and J. Ferruz. “Infrared Inspection of Buildings using Autonomous Helicopters”. In: *IFAC Proceedings Volumes* 39 (16 2006). ISSN: 14746670. DOI: 10.3182/20060912-3-DE-2911.00105.

[24] A. Ellenberg et al. “Use of Unmanned Aerial Vehicle for Quantitative Infrastructure Evaluation”. In: 21.3 (Sept. 2015), p. 04014054. DOI: 10.1061/(asce)is.1943-555x.0000246.

[25] Christian Eschmann, Chen-Ming Kuo, and Christian Boller. “Unmanned Aircraft Systems for Remote Building Inspection and Monitoring”. In: *Proceedings of the 6th European Workshop on Structural Health Monitoring, July 3-6, 2012, Dresden, Germany* 2 (2012), pp. 1–8.

[26] Davide Falanga et al. “Vision-based autonomous quadrotor landing on a moving platform”. In: *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. IEEE, Oct. 2017. DOI: 10.1109/ssrr.2017.8088164.

[27] M. Fiala. “ARTag, a Fiducial Marker System Using Digital Techniques”. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 2. San Diego, CA, USA, 2005, pp. 590–596. ISBN: 0-7695-2372-2. DOI: 10.1109/CVPR.2005.74.

[28] Mark Fiala. “Designing Highly Reliable Fiducial Markers”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.7 (July 2010), pp. 1317–1324. DOI: 10.1109/tpami.2009.146.

[29] Friedrich Fraundorfer and Davide Scaramuzza. “Visual Odometry : Part II: Matching, Robustness, Optimization, and Applications”. In: *IEEE Robotics &*

Automation Magazine 19.2 (June 2012), pp. 78–90. DOI: 10.1109/mra.2012.2182810.

[30] Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. "Visual simultaneous localization and mapping: a survey". In: *Artificial Intelligence Review* 43.1 (Nov. 2012), pp. 55–81. DOI: 10.1007/s10462-012-9365-8.

[31] H. Garnier, M. Mensler, and A. Richard. "Continuous-time model identification from sampled data: Implementation issues and performance evaluation". In: *International Journal of Control* 76.13 (Jan. 2003), pp. 1337–1357. ISSN: 0020-7179. DOI: 10.1080/0020717031000149636.

[32] S. Garrido-Jurado et al. "Automatic generation and detection of highly reliable fiducial markers under occlusion". In: *Pattern Recognition* 47.6 (June 2014), pp. 2280–2292. DOI: 10.1016/j.patcog.2014.01.005.

[33] Alvika Gautam, P.B. Sujit, and Srikanth Saripalli. "A survey of autonomous landing techniques for UAVs". In: *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. Orlando, FL, USA, May 2014, pp. 1210–1218. ISBN: 978-1-4799-2376-2. DOI: 10.1109/ICUAS.2014.6842377.

[34] R. Ghorbani, F. Matta, and M.A. Sutton. "Full-Field Deformation Measurement and Crack Mapping on Confined Masonry Walls Using Digital Image Correlation". In: 55.1 (June 2014), pp. 227–243. DOI: 10.1007/s11340-014-9906-y.

[35] *GPS accuracy*. URL: https://www.gps.gov/systems/gps/performance/accuracy/ (visited on 11/06/2021).

[36] Benjamin A. Graybeal et al. "Visual Inspection of Highway Bridges". In: *Journal of Nondestructive Evaluation* 21.3 (Sept. 2002), pp. 67–83. DOI: 10.1023/a:1022508121821.

[37] Roderic A. Grupen, Thomas C. Henderson, and Ian D. McCammon. "A Survey of General- Purpose Manipulation". In: *The International Journal of Robotics Research* 8.1 (Feb. 1989), pp. 38–62. DOI: 10.1177/027836498900800103.

[38] Nenad Gucunski et al. "Nondestructive evaluation inspection of the Arlington Memorial Bridge using a robotic assisted bridge inspection tool (RABIT)". In: ed. by H. Felix Wu et al. SPIE, Apr. 2014. DOI: 10.1117/12.2063963.

[39] Salua Hamaza et al. "Sensor Installation and Retrieval Operations Using an Unmanned Aerial Manipulator". In: *IEEE Robotics and Automation Letters* 4.3 (July 2019), pp. 2793–2800. DOI: 10.1109/lra.2019.2918448.

[40] Karol Hausman et al. “Self-calibrating multi-sensor fusion with probabilistic measurement validation for seamless sensor switching on a UAV”. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. Stockholm, Sweden, May 2016, pp. 4289–4296. ISBN: 978-1-4673-8026-3. DOI: 10.1109/ICRA.2016.7487626.

[41] Shuhei Hiasa et al. “Bridge Inspection and Condition Assessment Using Image-Based Technologies with UAVs”. In: American Society of Civil Engineers, Apr. 2018. ISBN: 9780784481332. DOI: 10.1061/9780784481332.020.

[42] Hamido Hourani et al. “A Marsupial Relationship in Robotics: A Survey”. In: *Intelligent Robotics and Applications. ICIRA 2011. Lecture Notes in Computer Science*. Ed. by Sabina Jeschke, Honghai Liu, and Daniel Schilberg. Vol. 7101. Berlin, Heidelberg: Springer, 2011, pp. 335–345. DOI: 10.1007/978-3-642-25486-4_34.

[43] B. Houska, H.J. Ferreau, and M. Diehl. “ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization”. In: *Optimal Control Applications and Methods* 32.3 (2011), pp. 298–312.

[44] B. Houska, H.J. Ferreau, and M. Diehl. “An Auto-Generated Real-Time Iteration Algorithm for Nonlinear MPC in the Microsecond Range”. In: *Automatica* 47.10 (2011), pp. 2279–2285. DOI: 10.1016/j.automatica.2011.08.020.

[45] Dongna Hu et al. “Wall crack detection based on image processing”. In: IEEE, July 2012. DOI: 10.1109/icicip.2012.6391474.

[46] Bharat Joshi et al. “Experimental Comparison of Open Source Visual-Inertial-Based State Estimation Algorithms in the Underwater Domain”. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Nov. 2019. DOI: 10.1109/iros40897.2019.8968049.

[47] Michail Kalaitzakis et al. “A marsupial robotic system for surveying and inspection of freshwater ecosystems”. In: *Journal of Field Robotics* 38.1 (Jan. 2021), pp. 121–138. DOI: 10.1002/rob.21957.

[48] Michail Kalaitzakis et al. “Drone-Based StereoDIC: System Development, Experimental Validation and Infrastructure Application”. In: (Apr. 2021). DOI: 10.1007/s11340-021-00710-z.

[49] Michail Kalaitzakis et al. “Dynamic Structural Health Monitoring using a DIC-enabled drone”. In: *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*. Atlanta, GA, USA, June 2019, pp. 321–327. ISBN: 978-1-7281-0333-4. DOI: 10.1109/ICUAS.2019.8798270.

[50] Michail Kalaitzakis et al. “Experimental Comparison of Fiducial Markers for Pose Estimation”. In: *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*. Athens, Greece, Sept. 2020, pp. 781–789. DOI: 10.1109/icuas48674.2020.9213977.

[51] Michail Kalaitzakis et al. “Fiducial Markers for Pose Estimation”. In: *Journal of Intelligent & Robotic Systems* 101.4 (Mar. 2021). DOI: 10.1007/s10846-020-01307-9.

[52] Mina Kamel et al. “Model Predictive Control for Trajectory Tracking of Unmanned Aerial Vehicles Using Robot Operating System”. In: *Studies in Computational Intelligence*. Springer International Publishing, 2017, pp. 3–39. DOI: 10.1007/978-3-319-54927-9_1.

[53] Seokyong Kim et al. “Design, Implementation, and Experiment of an Underwater Robot for Effective Inspection of Underwater Structures”. In: Springer International Publishing, 2014, pp. 821–829. DOI: 10.1007/978-3-319-05582-4_72.

[54] Xiangxiong Kong and Jian Li. “Vision-Based Fatigue Crack Detection of Steel Structures Using Video Feature Tracking”. In: *Computer-Aided Civil and Infrastructure Engineering* 33.9 (Feb. 2018), pp. 783–799. DOI: 10.1111/mice.12353.

[55] Sarantis Kyristsis et al. “Towards Autonomous Modular UAV Missions: The Detection, Geo-Location and Landing Paradigm”. In: *Sensors* 16.11 (Nov. 2016), p. 1844. DOI: 10.3390/s16111844.

[56] David Lattanzi and Gregory Miller. “Review of Robotic Infrastructure Inspection Systems”. In: *Journal of Infrastructure Systems* 23 (3 Sept. 2017), p. 04017004. ISSN: 1076-0342. DOI: 10.1061/(asce)is.1943-555x.0000353.

[57] J. J. Leonard and H. F. Durrant-Whyte. “Mobile robot localization by tracking geometric beacons”. In: *IEEE Transactions on Robotics and Automation* 7.3 (1991), pp. 376–382.

[58] Yao Liu et al. “Two-Layer Routing for High-Voltage Powerline Inspection by Cooperated Ground Vehicle and Drone”. In: *Energies* 12 (7 Apr. 2019). ISSN: 1996-1073. DOI: 10.3390/en12071385.

[59] Lennart Ljung. “System Identification”. In: *Signal Analysis and Prediction*. Boston, MA: Birkhäuser Boston, 1998, pp. 163–173. ISBN: 978-1-4612-1768-8. DOI: 10.1007/978-1-4612-1768-8_11.

[60] Barritt Lovelace and Jennifer Wells. *Improving the Quality of Bridge Inspections Using Unmanned Aircraft Systems (UAS)*. MN/RC 2018-26. Minnesota Department of Transportation, 2018. URL: http://mndot.gov/research/reports/2018/201826.pdf.

[61] Venkatesh Madyastha et al. "Extended Kalman Filter vs. Error State Kalman Filter for Aircraft Attitude Estimation". In: *AIAA Guidance, Navigation, and Control Conference*. American Institute of Aeronautics and Astronautics, June 2011. DOI: 10.2514/6.2011-6615.

[62] Danylo Malyuta et al. "Long-duration fully autonomous operation of rotorcraft unmanned aerial systems for remote-sensing data acquisition". In: *Journal of Field Robotics* 37.1 (2019), pp. 137–157. ISSN: 1556-4959. DOI: 10.1002/rob.21898.

[63] Nick McCormick and Jerry Lord. "Digital Image Correlation". In: *Materials Today* 13 (12 Dec. 2010), pp. 52–54. ISSN: 13697021. DOI: 10.1016/S1369-7021(10)70235-2.

[64] R. Mehra. "Approaches to adaptive filtering". In: *IEEE Transactions on Automatic Control* 17.5 (Oct. 1972), pp. 693–698. ISSN: 0018-9286. DOI: 10.1109/TAC.1972.1100100.

[65] Najib Metni and Tarek Hamel. "A UAV for bridge inspection: Visual servoing control law with orientation limits". In: 17.1 (Nov. 2007), pp. 3–10. DOI: 10.1016/j.autcon.2006.12.010.

[66] Mark Moore et al. *Reliability of Visual Inspection for Highway Bridges*. FHWA-RD-01-020. Department of Transportation, Federal Highway Administration, 2001. URL: https://www.fhwa.dot.gov/publications/research/nde/pdfs/01020a.pdf.

[67] Jason Moulton et al. "An Autonomous Surface Vehicle for Long Term Operations". In: *OCEANS 2018 MTS/IEEE Charleston*. IEEE, Oct. 2018, pp. 1–10. DOI: 10.1109/oceans.2018.8604718. URL: https://doi.org/10.1109/oceans.2018.8604718.

[68] Jason Moulton et al. "Dynamic Autonomous Surface Vehicle Controls, Aiding Disaster Response and Enabling Responsible Port Growth". In: *12th Conference on Field and Service Robotics (FSR)*. Tokyo, Japan, Aug. 2019.

[69] Robin R. Murphy et al. "Marsupial-like mobile robot societies". In: *Proceedings of the third annual conference on Autonomous Agents - AGENTS '99*. New York, NY, USA, 1999, pp. 364–365. ISBN: 158113066X. DOI: 10.1145/301136.301236.

[70] Anibal Ollero et al. “Past, Present, and Future of Aerial Robotic Manipulators”. In: (2021), pp. 1–20. DOI: 10.1109/tro.2021.3084395.

[71] Edwin Olson. “AprilTag: A robust and flexible visual fiducial system”. In: *2011 IEEE International Conference on Robotics and Automation*. Shanghai, China, May 2011, pp. 3400–3407. ISBN: 978-1-61284-386-5. DOI: 10.1109/ICRA.2011.5979561.

[72] Brent M. Phares et al. “Routine Highway Bridge Inspection Condition Documentation Accuracy and Reliability”. In: *Journal of Bridge Engineering* 9.4 (2004), pp. 403–413. DOI: 10.1061/(ASCE)1084-0702(2004)9:4(403).

[73] S. Rajan et al. “A Stereovision Deformation Measurement System for Transfer Length Estimates in Prestressed Concrete”. In: *Experimental Mechanics* 58 (7 Sept. 2018), pp. 1035–1048. ISSN: 17412765. DOI: 10.1007/s11340-017-0357-0.

[74] Tarek Rakha and Alice Gorodetsky. “Review of Unmanned Aerial System (UAS) applications in the built environment: Towards automated building inspection procedures using drones”. In: *Automation in Construction* 93 (Sept. 2018). ISSN: 09265805. DOI: 10.1016/j.autcon.2018.05.002.

[75] Daniel Reagan, Alessandro Sabato, and Christopher Niezrecki. “Unmanned aerial vehicle acquisition of three-dimensional digital image correlation measurements for structural health monitoring of bridges”. In: ed. by H. Felix Wu et al. Vol. 10169. Apr. 2017, p. 1016909. ISBN: 9781510608238. DOI: 10.1117/12.2259985.

[76] Ioannis Rekleitis, Gregory Dudek, and Evangelos Milios. “Multi-Robot Exploration of an Unknown Environment, Efficiently Reducing the Odometry Error”. In: *International Joint Conference in Artificial Intelligence (IJCAI)*. Vol. 2. Nagoya, Japan, 1997, pp. 1340–1345.

[77] Ioannis Rekleitis, Gregory Dudek, and Evangelos Milios. “On Multiagent Exploration”. In: *Vision Interface*. Vancouver, BC, Canada, 1998, pp. 455–461.

[78] S.I. Roumeliotis, G.S. Sukhatme, and G.A. Bekey. “Circumventing dynamic modeling: evaluation of the error-state Kalman filter applied to mobile robot localization”. In: *Proceedings 1999 IEEE International Conference on Robotics and Automation*. IEEE. DOI: 10.1109/robot.1999.772597.

[79] Fabio Ruggiero, Vincenzo Lippiello, and Anibal Ollero. “Aerial Manipulation: A Literature Review”. In: *IEEE Robotics and Automation Letters* 3.3 (July 2018), pp. 1957–1964. DOI: 10.1109/lra.2018.2808541.

[80] Jose Luis Sanchez-Lopez et al. “Toward visual autonomous ship board landing of a VTOL UAV”. In: *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*. Atlanta, GA, USA, May 2013, pp. 779–788. ISBN: 978-1-4799-0817-2. DOI: 10.1109/ICUAS.2013.6564760.

[81] Diego Brito dos Santos Cesar et al. “An evaluation of artificial fiducial markers in underwater environments”. In: *OCEANS 2015 - Genova*. IEEE, May 2015. DOI: 10.1109/oceans-genova.2015.7271491.

[82] Davide Scaramuzza and Friedrich Fraundorfer. “Visual Odometry [Tutorial]”. In: *IEEE Robotics & Automation Magazine* 18.4 (Dec. 2011), pp. 80–92. DOI: 10.1109/mra.2011.943233.

[83] Johannes L. Schönberger and Jan-Michael Frahm. “Structure-from-Motion Revisited”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA, June 2016, pp. 4104–4113. ISBN: 978-1-4673-8851-1. DOI: 10.1109/CVPR.2016.445.

[84] Johannes L. Schönberger et al. “Pixelwise View Selection for Unstructured Multi-View Stereo”. In: *Computer Vision – ECCV 2016. Lecture Notes in Computer Science*. Vol. 9907. Springer, Cham, 2016, pp. 501–518. DOI: 10.1007/978-3-319-46487-9_31.

[85] Hubert Schreier, Jean-José Orteu, and Michael A. Sutton. *Image Correlation for Shape, Motion and Deformation Measurements*. Springer US, 2009. ISBN: 978-0-387-78746-6. DOI: 10.1007/978-0-387-78747-3.

[86] Junwon Seo, Luis Duque, and Jim Wacker. “Drone-enabled bridge inspection methodology and application”. In: *Automation in Construction* 94 (Oct. 2018). ISSN: 09265805. DOI: 10.1016/j.autcon.2018.06.006.

[87] Ksenia Shabalina et al. “ARTag, AprilTag and CALTag Fiducial Systems Comparison in a Presence of Partial Rotation: Manual and Automated Approaches”. In: *Informatics in Control, Automation and Robotics*. Springer International Publishing, Apr. 2019, pp. 536–558. DOI: 10.1007/978-3-030-11292-9_27.

[88] Hazim Shakhatreh et al. “Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges”. In: *IEEE Access* 7 (2019). ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2909530.

[89] Joan Solà. *Quaternion kinematics for the error-state Kalman filter*. 2017. DOI: 10.48550/ARXIV.1711.02508.

[90] M. A. Sutton et al. “Recent Progress in Digital Image Correlation: Background and Developments since the 2013 W M Murray Lecture”. In: *Experimental Mechanics* 57 (1 Jan. 2017), pp. 1–30. ISSN: 17412765. DOI: 10.1007/s11340-016-0233-3.

[91] Kazuya Takeuchi et al. “A close inspection and vibration sensing aerial robot for steel structures with an EPM-based landing device”. In: *Nondestructive Characterization and Monitoring of Advanced Materials, Aerospace, and Civil Infrastructure 2017*. Ed. by H. Felix Wu et al. SPIE, Apr. 2017. DOI: 10.1117/12.2260386.

[92] V. Valle et al. “Digital Image Correlation Development for the Study of Materials Including Multiple Crossing Cracks”. In: 55.2 (Sept. 2014), pp. 379–391. DOI: 10.1007/s11340-014-9948-1.

[93] Panagiotis Vlantis et al. “Quadrotor landing on an inclined platform of a moving ground vehicle”. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. Seattle, WA, USA, May 2015, pp. 2202–2207. ISBN: 978-1-4799-6923-4. DOI: 10.1109/ICRA.2015.7139490.

[94] VTT Technical Research Centre of Finland Ltd. *A Library for Virtual and Augmented Reality*. http://virtual.vtt.fi/virtual/proj2/multimedia/alvar/. 2019.

[95] John Wang and Edwin Olson. “AprilTag 2: Efficient and robust fiducial detection”. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Daejeon, South Korea, Oct. 2016, pp. 4193–4198. ISBN: 978-1-5090-3762-9. DOI: 10.1109/IROS.2016.7759617.

[96] Y. Q. Wang et al. “Quantitative error assessment in pattern matching: Effects of intensity pattern noise, interpolation, strain and image contrast on motion measurements”. In: *Strain* 45 (2 Apr. 2009), pp. 160–178. ISSN: 00392103. DOI: 10.1111/j.1475-1305.2008.00592.x.

[97] Zongzheng Wang, Shibin Wang, and Zhiyong Wang. “The elimination of pseudo strains in 2D-DIC caused by out-of-plane translation using light strip method”. In: ed. by Chenggen Quan, Kemao Qian, and Anand Asundi. SPIE, June 2013. DOI: 10.1117/12.2018827.

[98] Chul Min Yeum and Shirley J. Dyke. “Vision-Based Automated Crack Detection for Bridge Inspection”. In: *Computer-Aided Civil and Infrastructure Engineering* 30 (10 2015), pp. 759–770. ISSN: 14678667. DOI: 10.1111/mice.12141.

[99] Hyungchul Yoon et al. “Cross-Correlation-Based Structural System Identification Using Unmanned Aerial Vehicles”. In: *Sensors* 17.9 (2017), p. 2075. ISSN: 1424-8220. DOI: 10.3390/s17092075.

[100] Gang Yu et al. “Objects Grasping of Robotic Arm with Compliant Grasper Based on Vision”. In: *Proceedings of the 2019 4th International Conference on Automation, Control and Robotics Engineering*. ACM, July 2019. DOI: 10.1145/3351917.3351958.

[101] Adam I. Zeitouni, Dimitris C. Rizos, and Yu Qian. “Benefits of high strength reduced modulus (HSRM) concrete railroad ties under center binding support conditions”. In: *Construction and Building Materials* 192 (Dec. 2018), pp. 210–223. DOI: 10.1016/j.conbuildmat.2018.10.124.

[102] Xuetao Zhang et al. “Enhanced Fiducial Marker Based Precise Landing for Quadrotors”. In: *2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, July 2019. DOI: 10.1109/aim.2019.8868532.

[103] Qin Zou et al. “CrackTree: Automatic crack detection from pavement images”. In: 33.3 (Feb. 2012), pp. 227–238. DOI: 10.1016/j.patrec.2011.11.004.

[104] Yongxia Zuo, Guoqiang Wang, and Chuncheng Zuo. “A Novel Image Segmentation Method of Pavement Surface Cracks Based on Fractal Theory”. In: IEEE, Dec. 2008. DOI: 10.1109/cis.2008.206.

Appendix A

Experimental Comparison of Fiducial Markers for Pose Estimation Results

Table A.1: Single Marker - Distance Results

Camera	Distance (cm)	Light	ARTag			AprilTag			ArUco			STag		
			mean	std	rate	mean	std	rate	mean	std	rate	mean	std	rate
Logitech	75	N	2.262	0.046	42.899	3.850	**0.017**	96.914	2.067	0.021	**97.072**	**1.863**	0.024	93.577
		S	2.204	0.024	43.130	3.900	0.021	**96.997**	2.034	0.027	95.976	**1.859**	**0.017**	93.090
	100	N	2.183	**0.017**	43.699	4.721	0.021	**96.402**	2.204	0.028	95.926	**1.671**	0.067	89.325
		S	2.224	0.051	44.260	4.723	0.021	**96.163**	2.184	0.028	96.030	**1.716**	**0.007**	91.958
	150	N	2.484	0.442	43.375	5.981	**0.073**	**96.644**	2.242	0.112	96.165	**1.450**	0.249	95.562
		S	1.909	0.212	43.634	6.052	**0.106**	**97.068**	2.443	0.315	96.071	**1.425**	0.108	95.007
	175	N	2.668	0.344	43.546	6.559	0.146	96.074	2.315	0.229	96.103	**1.012**	**0.107**	**96.112**
		S	ND	ND	0.0	6.639	0.121	**96.554**	2.620	0.197	95.994	**1.131**	**0.059**	96.036
	200	N	3.155	0.220	43.611	7.236	**0.191**	96.012	3.050	0.315	96.083	**0.648**	0.257	**96.511**
		S	ND	ND	0.0	7.273	0.251	**96.473**	3.615	0.200	**96.473**	**0.680**	**0.192**	96.054
Pi camera	75	N	1.431	**0.007**	47.544	2.142	0.012	**94.704**	1.284	0.011	94.693	**0.737**	0.077	90.606
		S	1.549	**0.015**	47.118	2.306	0.031	94.444	1.353	0.020	**94.829**	**0.764**	0.057	94.636
	100	N	1.954	0.084	49.766	2.541	0.031	94.429	1.562	0.026	94.595	**0.842**	**0.023**	**98.448**
		S	2.214	0.053	47.349	2.785	0.042	**95.333**	1.813	0.100	93.417	**0.993**	**0.024**	**95.333**
	150	N	2.617	0.280	47.313	3.292	0.232	**95.333**	3.632	**0.047**	94.051	**0.475**	0.757	75.805
		S	3.184	0.226	47.447	3.596	0.082	**94.865**	3.748	**0.078**	94.123	**0.495**	0.588	94.624
	175	N	1.768	**0.192**	47.292	3.410	0.340	94.787	3.827	0.612	**94.917**	**0.093**	1.041	93.279
		S	3.824	**0.071**	46.891	4.081	0.235	**94.366**	4.838	0.696	93.001	**0.398**	0.540	93.361
	200	N	2.697	0.289	47.627	3.510	0.212	94.429	3.780	1.361	95.067	**1.542**	**0.127**	**95.410**
		S	4.403	0.441	47.024	4.877	0.290	94.648	6.151	0.875	**95.087**	**1.194**	**0.240**	91.589

Table A.2: Two Marker Bundle - Distance Results

Camera	Distance (cm)	Light	ARTag			AprilTag			ArUco			STag		
			mean	std	rate	mean	std	rate	mean	std	rate	mean	std	rate
Logitech	75	N	3.205	4.223	100.00	1.946	0.006	100.00	1.584	0.007	99.670	0.637	0.008	100.00
		S	1.759	1.132	100.00	1.605	0.019	100.00	1.788	0.009	99.750	0.887	0.016	100.00
	100	N	2.179	2.299	100.00	2.731	0.009	100.00	2.315	0.010	99.579	0.932	0.019	100.00
		S	1.314	5.906	100.00	2.384	0.040	100.00	1.914	0.037	99.665	1.034	0.062	100.00
	150	N	0.625	0.032	100.00	3.943	0.034	100.00	3.633	0.049	99.753	1.777	0.167	100.00
		S	—	—	100.00	3.571	0.078	100.00	2.619	0.115	99.658	1.655	0.172	100.00
	175	N	0.486	0.086	100.00	5.128	0.179	100.00	4.623	0.165	99.756	1.980	0.340	100.00
		S	1.117	7.541	100.00	4.825	0.080	99.916	2.344	0.097	99.830	1.395	0.218	100.00
	200	N	0.403	6.797	100.00	5.779	0.088	100.00	5.614	0.203	99.677	1.640	0.233	100.00
		S	0.175	0.089	100.00	5.315	0.126	99.915	2.496	0.053	99.661	1.425	0.217	100.00
Pi camera	75	N	1.017	3.639	100.00	0.642	0.020	99.633	1.043	0.007	99.637	1.790	0.008	100.00
		S	1.911	5.934	100.00	0.358	0.015	100.00	0.460	0.009	99.727	2.069	0.048	99.816
	100	N	0.946	2.963	100.00	0.792	0.009	99.363	1.994	0.000	99.727	2.036	0.009	99.813
		S	—	—	100.00	0.445	0.032	99.320	1.003	0.027	99.819	2.611	0.615	100.00
	150	N	1.708	4.687	100.00	1.425	0.044	99.415	3.075	0.262	99.633	2.235	0.196	99.908
		S	1.148	2.251	100.00	1.300	0.087	98.948	2.816	0.716	99.728	2.459	0.068	100.00
	175	N	0.345	0.294	100.00	1.417	0.243	99.813	4.318	0.008	99.638	4.575	1.316	100.00
		S	0.217	0.131	100.00	1.523	0.267	99.546	4.393	1.507	99.632	3.444	0.329	100.00
	200	N	0.892	0.015	100.00	2.327	0.072	99.909	ND	ND	0.000	5.593	0.499	99.909
		S	2.086	8.158	100.00	2.125	0.744	99.818	9.808	0.329	99.819	2.429	2.226	67.854

Table A.3: Three Marker Bundle - Distance Results

Distance	ARTag			AprilTag			ArUco			STag		
	mean	std	rate	mean	std	rate	mean	std	rate	mean	std	rate
50cm	0.128	0.005	100.00	8.254	0.005	99.815	0.925	0.005	99.728	0.641	0.033	99.909
75cm	0.124	0.064	100.00	13.852	0.015	99.427	1.674	0.014	99.729	1.151	0.046	100.00
100cm	1.153	7.788	100.00	19.394	0.041	99.548	4.073	0.101	99.725	2.108	0.139	99.816
150cm	1.814	0.067	100.00	4.952	0.055	99.728	5.746	0.210	99.727	4.283	0.308	100.00
175cm	2.488	8.858	100.00	4.571	0.188	99.083	8.573	0.000	99.728	1.439	0.075	12.375
200cm	0.413	1.253	100.00	4.877	0.273	99.546	ND	ND	0.000	1.966	1.921	44.918

Table A.4: Single Marker Angle Step - Angle Results

Camera	Angle	Light	ARTag			AprilTag			ArUco			STag		
			mean	std	rate	mean	std	rate	mean	std	rate	mean	std	rate
Logitech	0°	N	2.831	0.158	45.833	2.450	**0.068**	99.539	**2.386**	0.176	**99.552**	3.317	0.347	92.825
		S	3.132	0.102	45.045	3.417	**0.039**	**100.00**	**2.012**	0.053	99.545	3.380	0.092	94.444
	10°	N	**0.625**	**0.047**	46.083	0.884	0.168	**99.552**	1.773	0.115	99.548	1.118	0.135	91.855
		S	2.039	**0.081**	45.455	**1.386**	0.136	**99.541**	1.401	0.131	99.539	1.421	0.329	93.953
	20°	N	2.326	**0.055**	46.296	**1.517**	0.080	99.545	2.219	0.178	**100.00**	2.319	0.115	92.793
		S	3.129	0.089	45.045	**2.299**	**0.068**	**99.545**	2.744	0.074	99.543	2.971	0.089	93.088
	30°	N	**1.162**	0.068	46.330	1.325	**0.066**	99.541	1.379	0.122	**100.00**	1.565	0.077	91.818
		S	1.527	0.125	45.662	**1.310**	**0.057**	99.537	1.455	0.105	**99.539**	1.728	0.302	94.907
	40°	N	0.214	0.050	46.083	0.641	**0.024**	99.091	**0.203**	0.025	**99.550**	0.590	0.070	92.991
		S	0.564	0.063	45.662	**0.456**	**0.031**	99.087	0.910	0.050	**99.548**	0.605	0.186	96.789
	50°	N	**0.141**	0.022	44.595	0.365	**0.015**	**99.543**	0.459	0.046	99.087	0.244	0.033	94.064
		S	0.156	**0.021**	45.662	**0.022**	0.024	99.528	0.162	0.046	**99.537**	0.047	0.204	95.909
	60°	N	0.346	0.030	44.395	**0.126**	**0.009**	99.543	0.239	0.011	**100.00**	0.422	0.019	97.727
		S	0.473	0.020	45.455	0.264	**0.016**	**100.00**	**0.188**	0.018	**100.00**	0.194	0.558	95.045
	70°	N	0.738	0.039	45.662	**0.476**	0.044	99.083	0.779	**0.015**	**99.545**	0.699	0.064	99.541
		S	0.716	**0.009**	45.872	0.399	0.012	**99.103**	0.607	0.013	99.087	**0.097**	0.282	96.330
	80°	N	1.270	**0.039**	41.176	**1.052**	0.089	**100.00**	1.238	0.046	99.087	1.258	0.042	98.190
		S	1.143	0.042	45.045	0.874	**0.009**	99.552	0.976	0.010	**100.00**	**0.602**	0.235	97.273
Pi camera	0°	N	1.883	**0.110**	50.249	0.649	0.267	99.010	**0.602**	0.604	99.010	0.627	0.363	**99.502**
		S	**0.010**	**0.147**	49.751	0.288	0.611	**99.502**	4.863	0.435	**99.502**	0.411	0.419	**99.502**
	10°	N	1.532	0.200	49.751	**0.476**	**0.092**	97.887	1.024	0.178	**99.502**	1.261	1.531	**99.502**
		S	1.521	0.100	49.451	**0.323**	0.222	**99.502**	4.201	**0.041**	**99.502**	1.801	0.874	**99.502**
	20°	N	0.062	**0.052**	49.254	**0.028**	0.054	**99.502**	0.138	0.080	**99.502**	0.144	0.109	**99.502**
		S	0.487	**0.063**	49.751	0.561	0.152	**99.502**	**0.278**	0.111	**99.502**	0.600	0.184	98.701
	30°	N	0.256	**0.039**	49.367	**0.035**	0.055	99.502	0.765	0.141	99.502	0.494	0.091	**100.00**
		S	0.294	**0.036**	49.751	0.288	**0.036**	**99.502**	0.255	0.067	**99.502**	**0.134**	0.097	**99.502**
	40°	N	0.092	0.038	49.254	**0.034**	**0.016**	98.658	0.296	0.031	99.502	0.457	0.387	**100.00**
		S	0.148	0.034	49.751	**0.026**	0.075	99.005	0.502	**0.016**	**99.502**	0.497	0.025	**99.502**
	50°	N	0.440	0.059	48.795	0.519	0.025	**99.502**	0.134	**0.018**	**99.502**	**0.064**	0.032	**99.502**
		S	0.210	0.018	49.751	**0.025**	0.020	99.502	0.207	**0.016**	99.502	0.434	0.020	**100.00**
	60°	N	0.163	**0.015**	49.254	**0.022**	0.016	97.887	0.528	0.033	**99.502**	0.064	0.032	**99.502**
		S	0.477	0.015	49.412	**0.066**	0.019	**99.502**	0.193	**0.010**	99.005	0.172	0.034	97.279
	70°	N	**0.131**	0.015	49.751	0.137	0.011	97.902	0.299	**0.007**	**100.00**	0.324	0.048	98.026
		S	**0.071**	0.031	49.751	0.175	0.031	**99.502**	0.342	**0.009**	99.005	0.184	0.152	98.990
	80°	N	0.335	0.019	49.020	**0.235**	0.012	**99.502**	0.518	**0.004**	**99.502**	0.276	0.169	36.634
		S	0.169	0.097	49.505	**0.010**	0.025	**100.00**	0.153	**0.016**	99.502	0.257	0.073	94.527

Table A.5: Planar Two Marker Bundle Angle Step - Angle Results

Camera	Angle	Light	ARTag			AprilTag			ArUco			STag		
			mean	std	rate	mean	std	rate	mean	std	rate	mean	std	rate
Logitech	0°	N	5.468	6.256	80.198	3.031	0.061	78.947	3.292	**0.043**	**81.068**	**2.102**	0.118	79.512
		S	3.814	1.544	79.426	2.116	**0.061**	79.327	2.314	0.103	**80.193**	**1.637**	0.112	79.904
	10°	N	3.777	2.716	49.268	1.549	1.237	51.196	1.390	1.556	49.029	**1.023**	**0.913**	**52.261**
		S	3.753	2.558	**51.232**	1.331	0.647	49.528	1.548	**0.640**	50.495	**0.812**	0.746	50.244
	20°	N	4.096	4.297	69.903	0.540	**0.024**	**70.388**	**0.166**	**0.024**	69.565	0.270	0.059	66.505
		S	3.760	3.076	69.268	0.769	**0.057**	**71.905**	1.026	0.072	67.788	**0.220**	0.160	69.712
	30°	N	6.856	9.106	79.710	**0.279**	**0.022**	80.193	1.289	0.024	**80.882**	0.842	0.351	80.392
		S	7.321	8.107	78.469	**1.241**	0.041	79.147	1.526	**0.024**	**80.676**	1.401	0.052	80.097
	40°	N	3.559	8.774	**50.242**	0.659	0.335	48.571	0.921	**0.315**	47.619	**0.492**	0.493	49.048
		S	6.438	12.883	50.244	**0.954**	**0.187**	50.476	1.122	0.325	49.519	1.179	0.211	**50.485**
	50°	N	3.115	11.316	70.443	0.940	**0.013**	70.874	0.628	0.032	**71.359**	**0.102**	0.089	68.421
		S	2.246	4.627	**72.059**	**0.799**	0.017	71.014	0.859	**0.012**	71.220	1.021	0.069	69.903
	60°	N	4.502	12.057	82.178	1.419	0.064	80.861	**0.405**	0.005	81.643	0.459	**0.020**	**82.353**
		S	1.853	**0.008**	79.710	0.343	0.054	**80.000**	**0.230**	0.020	79.327	0.307	0.016	79.426
	70°	N	4.071	13.815	49.038	1.708	0.221	50.244	**0.589**	**0.150**	50.000	0.907	0.379	**51.220**
		S	2.163	0.270	49.029	**0.197**	0.109	50.000	0.249	**0.066**	**51.442**	0.262	0.095	48.804
	80°	N	3.615	10.127	67.773	1.872	**0.016**	70.048	**0.934**	0.122	68.750	1.262	0.037	**70.874**
		S	2.671	3.486	68.571	**0.212**	**0.009**	68.396	0.454	0.027	67.619	0.371	0.104	**69.565**
Pi camera	0°	N	0.638	0.189	**100.00**	**0.602**	**0.029**	**100.00**	2.408	0.051	**100.00**	2.055	0.055	**100.00**
		S	**0.492**	0.213	99.502	1.628	0.165	**100.00**	1.733	0.801	**100.00**	1.969	**0.086**	99.000
	10°	N	1.363	0.078	**100.00**	**0.249**	0.074	98.500	1.547	**0.048**	**100.00**	0.667	0.703	99.500
		S	1.046	0.124	**100.00**	**0.401**	0.114	**100.00**	0.596	**0.062**	**100.00**	0.771	0.183	**100.00**
	20°	N	0.423	0.082	**100.00**	**0.092**	**0.020**	99.500	0.577	0.027	**100.00**	0.769	0.029	**100.00**
		S	0.805	0.103	**100.00**	**0.160**	**0.098**	**100.00**	0.574	0.238	**100.00**	0.320	0.203	**100.00**
	30°	N	0.599	0.039	**100.00**	**0.041**	**0.012**	99.000	0.303	0.035	**100.00**	0.918	0.107	**100.00**
		S	0.641	0.158	**100.00**	**0.107**	**0.054**	**100.00**	0.302	0.100	**100.00**	0.940	0.236	**100.00**
	40°	N	**0.238**	0.033	**100.00**	0.313	0.107	**100.00**	0.275	**0.017**	**100.00**	0.386	0.023	**100.00**
		S	0.375	0.093	**100.00**	**0.100**	0.058	**100.00**	0.499	0.041	**100.00**	0.216	**0.029**	**100.00**
	50°	N	0.232	0.033	**100.00**	0.176	0.022	99.500	**0.005**	**0.006**	**100.00**	0.417	0.138	**100.00**
		S	0.252	0.082	**100.00**	**0.111**	0.039	**100.00**	0.589	**0.027**	**100.00**	0.307	0.184	99.000
	60°	N	0.089	0.031	**100.00**	**0.071**	0.018	99.000	0.131	**0.010**	**100.00**	0.299	0.239	**100.00**
		S	0.366	0.085	**100.00**	**0.067**	**0.016**	**100.00**	0.407	0.032	**100.00**	0.566	0.231	**100.00**
	70°	N	0.123	0.034	**100.00**	**0.071**	**0.017**	**100.00**	0.166	0.061	**100.00**	0.114	0.102	22.000
		S	0.184	0.024	**100.00**	**0.101**	**0.015**	**100.00**	1.511	0.021	**100.00**	0.221	0.200	83.500
	80°	N	0.183	0.136	**100.00**	**0.025**	**0.014**	99.005	ND	ND	0.000	ND	ND	0.000
		S	ND	ND	0.000	**0.039**	**0.022**	**99.500**	1.516	0.016	11.443	ND	ND	0.000

Table A.6: Single Marker Angle Step - Distance Results (Detection rate is omitted here since it is the same as in Table A.4)

Camera	Angle	Light	ARTag		April		ArUco		STag	
			mean	std	mean	std	mean	std	mean	std
Logitech	0°	N	2.370	0.052	5.097	**0.029**	2.294	0.045	**1.435**	0.138
		S	2.260	**0.027**	4.782	0.033	2.246	0.031	**1.895**	0.074
	10°	N	3.623	**0.019**	6.318	0.045	3.803	0.021	**3.004**	0.056
		S	3.741	**0.033**	6.295	0.035	**3.373**	0.040	3.514	0.128
	20°	N	5.512	**0.012**	7.934	0.013	5.585	0.021	**4.854**	0.032
		S	5.487	**0.009**	7.673	0.011	**5.067**	0.017	5.199	0.018
	30°	N	6.536	**0.013**	8.878	0.016	6.623	0.015	**5.697**	**0.013**
		S	6.498	**0.008**	8.627	0.010	6.427	0.010	**6.169**	0.027
	40°	N	7.016	**0.008**	9.346	0.010	7.232	0.011	**6.239**	0.013
		S	6.983	**0.006**	9.156	0.007	6.905	0.009	**6.755**	0.032
	50°	N	7.167	0.011	9.463	0.009	7.391	**0.008**	**6.328**	0.010
		S	7.082	**0.005**	9.262	0.008	7.035	0.009	**6.746**	0.026
	60°	N	6.891	**0.009**	9.198	**0.009**	7.224	0.011	**6.113**	0.026
		S	6.766	**0.008**	9.029	0.013	6.772	0.009	**6.498**	0.115
	70°	N	6.462	0.012	8.798	**0.011**	6.854	**0.011**	**5.701**	0.095
		S	6.252	**0.008**	8.525	0.010	6.166	0.012	**5.722**	0.126
	80°	N	5.230	0.022	7.722	**0.016**	5.917	0.021	**4.675**	0.163
		S	5.400	0.014	7.342	**0.011**	5.399	0.019	**4.824**	0.143
piCam	0°	N	1.451	**0.024**	2.599	0.025	1.972	0.048	**0.521**	0.105
		S	1.592	**0.034**	2.603	0.056	1.617	0.060	**0.490**	0.105
	10°	N	2.541	0.033	3.743	**0.013**	3.268	0.031	**2.280**	0.234
		S	2.636	0.022	3.630	0.048	**2.340**	**0.014**	2.446	0.102
	20°	N	3.636	0.029	4.664	0.015	4.030	0.041	**2.639**	**0.014**
		S	3.963	0.019	4.636	0.032	3.883	0.019	**2.608**	**0.015**
	30°	N	4.832	0.030	5.474	**0.016**	4.972	0.032	**3.580**	0.023
		S	4.996	0.019	5.557	**0.018**	4.808	0.021	**3.499**	0.020
	40°	N	5.419	0.024	6.284	**0.012**	5.886	0.026	**4.393**	0.024
		S	5.652	0.018	6.528	0.025	5.739	**0.011**	**4.426**	0.021
	50°	N	6.137	0.030	6.716	**0.013**	6.491	0.031	**4.927**	0.030
		S	6.181	0.024	7.004	**0.017**	6.236	0.026	**4.923**	0.029
	60°	N	6.322	0.039	6.991	**0.016**	6.765	0.032	**5.148**	0.067
		S	6.389	0.028	7.216	**0.016**	6.460	0.023	**5.249**	0.030
	70°	N	6.021	0.050	7.031	**0.016**	6.895	0.027	**5.198**	0.143
		S	6.412	0.030	7.252	0.030	6.421	**0.027**	**5.306**	0.035
	80°	N	5.801	**0.006**	6.637	0.015	6.431	0.029	**4.984**	0.060
		S	**5.171**	0.213	6.998	0.031	5.954	**0.020**	5.336	0.430

Table A.7: Two Marker Bundle Angle Step - Distance Results

Camera	Angle	Light	ARTag			AprilTag			ArUco			STag		
			mean	std	rate	mean	std	rate	mean	std	rate	mean	std	rate
Logitech	0°	N	3.446	8.079	80.198	2.267	0.017	78.947	2.380	0.012	81.068	1.368	0.055	79.512
		S	1.460	0.904	79.426	2.254	0.009	79.327	2.321	0.008	80.193	1.347	0.037	79.904
	10°	N	2.883	2.509	49.268	3.000	0.603	51.196	1.855	0.431	49.029	0.932	0.342	52.261
		S	1.514	1.577	51.232	2.985	0.622	49.528	1.761	0.460	50.495	0.791	0.500	50.244
	20°	N	3.344	3.585	69.903	3.480	0.016	70.388	1.516	0.008	69.565	0.653	0.011	66.505
		S	1.549	1.913	69.268	3.490	0.022	71.905	1.386	0.012	67.788	0.391	0.051	69.712
	30°	N	3.052	2.877	79.710	5.189	0.016	80.193	0.178	0.011	80.882	0.445	0.026	80.392
		S	3.492	3.081	78.469	5.162	0.008	79.147	0.023	0.014	80.676	0.376	0.012	80.097
	40°	N	2.092	2.101	50.242	5.938	0.635	48.571	0.288	0.100	47.619	0.663	0.172	49.048
		S	3.541	7.438	50.244	5.954	0.647	50.476	0.226	0.170	49.519	0.719	0.280	50.485
	50°	N	3.128	11.854	70.443	6.447	0.010	70.874	0.372	0.010	71.359	0.795	0.028	68.421
		S	1.840	1.322	72.059	6.466	0.011	71.014	0.361	0.010	71.220	0.948	0.055	69.903
	60°	N	2.286	2.653	82.178	7.422	0.009	80.861	0.439	0.006	81.643	0.809	0.023	82.353
		S	1.734	0.008	79.710	7.471	0.009	80.000	0.461	0.014	79.327	0.970	0.016	79.426
	70°	N	3.630	13.521	49.038	7.882	0.377	50.244	0.413	0.022	50.000	0.817	0.039	51.220
		S	1.697	0.033	49.029	7.910	0.364	50.000	0.426	0.037	51.442	0.974	0.054	48.804
	80°	N	2.119	3.165	67.773	8.187	0.012	70.048	0.405	0.010	68.750	0.832	0.025	70.874
		S	—	—	68.571	8.218	0.009	68.396	0.423	0.012	67.619	0.997	0.106	69.565
Pi camera	0°	N	1.744	0.010	100.00	0.020	0.011	100.00	0.032	0.006	100.00	3.162	0.025	100.00
		S	1.824	0.034	99.502	0.099	0.016	100.00	0.124	0.052	100.00	3.093	0.027	99.000
	10°	N	1.874	0.027	100.00	0.705	0.038	98.500	0.790	0.025	100.00	2.554	0.183	99.500
		S	1.872	0.034	100.00	0.391	0.013	100.00	0.344	0.037	100.00	2.980	0.042	100.00
	20°	N	1.415	0.055	100.00	1.490	0.010	99.500	1.305	0.017	100.00	2.136	0.020	100.00
		S	1.910	0.061	100.00	1.388	0.021	100.00	1.139	0.089	100.00	2.435	0.138	100.00
	30°	N	1.546	0.039	100.00	2.343	0.014	99.000	1.387	0.026	100.00	1.890	0.116	100.00
		S	1.292	0.105	100.00	2.140	0.035	100.00	0.514	0.090	100.00	2.207	0.206	100.00
	40°	N	1.240	0.034	100.00	3.124	0.016	100.00	1.426	0.023	100.00	2.168	0.027	100.00
		S	1.226	0.090	100.00	2.986	0.048	100.00	0.443	0.029	100.00	2.388	0.038	100.00
	50°	N	1.175	0.043	100.00	3.709	0.022	99.500	1.400	0.016	100.00	2.294	0.048	100.00
		S	1.033	0.067	100.00	3.632	0.061	100.00	0.439	0.051	100.00	2.624	0.137	99.000
	60°	N	1.558	0.066	100.00	4.222	0.033	99.000	1.214	0.023	100.00	2.384	0.073	100.00
		S	1.214	0.040	100.00	4.176	0.025	100.00	0.630	0.087	100.00	2.904	0.222	100.00
	70°	N	1.687	0.084	100.00	4.464	0.024	100.00	0.884	0.043	100.00	2.705	0.115	22.000
		S	1.289	0.038	100.00	4.444	0.046	100.00	0.139	0.007	100.00	3.079	0.160	83.500
	80°	N	2.056	0.104	100.00	4.554	0.027	99.005	ND	ND	0.000	ND	ND	0.000
		S	ND	ND	0.000	4.831	0.045	99.500	0.068	0.006	11.443	ND	ND	0.000

Table A.8: Non-Planar Two Marker Bundle Angle Step - Angle Results

Angle (degrees)	ARTag			AprilTag			ArUco			STag		
	mean	std	rate	mean	std	rate	mean	std	rate	mean	std	rate
-60	0.255	0.046	100.00	0.474	0.033	99.500	0.991	0.395	100.00	1.158	0.065	96.000
-50	0.038	0.025	100.00	2.925	0.020	99.000	0.055	0.026	100.00	0.420	0.206	100.00
-40	0.207	0.030	100.00	2.999	0.024	100.00	0.110	0.033	100.00	0.473	0.229	100.00
-30	0.387	0.022	100.00	3.832	0.020	100.00	0.234	0.028	100.00	0.408	0.045	100.00
-20	0.238	0.026	100.00	3.112	0.044	99.500	0.153	0.053	100.00	0.371	0.113	100.00
-10	0.053	0.044	100.00	1.863	0.023	99.500	0.538	0.028	100.00	0.726	0.078	100.00
0	0.090	0.032	100.00	0.422	0.025	99.500	0.285	0.106	100.00	0.298	0.022	100.00
10	0.456	0.027	100.00	0.385	0.039	100.00	0.272	0.020	100.00	0.042	0.045	100.00
20	0.085	0.010	100.00	1.823	0.034	99.500	0.492	0.026	100.00	0.694	0.077	99.502
30	0.020	0.014	99.502	2.021	0.019	99.500	0.284	0.033	100.00	0.095	0.045	100.00
40	0.762	9.488	100.00	1.703	0.045	99.500	0.257	0.020	100.00	0.115	0.111	100.00
50	0.475	2.309	100.00	0.944	0.024	100.00	1.022	0.026	100.00	0.193	0.053	99.500
60	0.716	0.031	100.00	8.745	0.033	100.00	0.494	0.033	100.00	0.333	0.147	97.000

Table A.9: Rotation Blur - Angle Results

Speed (degrees/sec)	ARTag			AprilTag			ArUco			STag		
	mean	std	rate	mean	std	rate	mean	std	rate	mean	std	rate
60	4.621	5.138	49.727	4.290	5.561	79.769	4.116	8.715	100.00	4.503	6.546	81.356
90	8.064	11.844	46.286	6.239	7.633	78.667	6.504	13.214	98.246	6.228	11.689	89.941
135	9.440	12.314	45.000	10.220	15.281	75.342	6.062	7.841	100.00	6.339	9.142	92.045
180	11.369	16.046	50.289	11.427	14.881	88.235	11.647	23.267	98.286	10.603	14.937	94.413
240	13.393	15.556	48.023	12.711	14.890	86.538	16.370	26.482	100.00	14.655	21.766	98.780
300	14.998	17.467	48.295	ND	ND	0.000	10.834	13.247	100.00	12.003	14.653	100.00

Table A.10: Translation Blur - X-axis Position Results

Speed (degrees/sec)	**ARTag** mean	std	rate	**AprilTag** mean	std	rate	**ArUco** mean	std	rate	**STag** mean	std	rate
30	5.540	6.756	49.189	5.330	6.254	88.525	9.994	11.481	87.647	**4.772**	**5.202**	**92.350**
60	6.022	7.438	49.718	5.993	7.155	94.972	10.318	12.133	89.326	**5.100**	**5.658**	**100.00**
90	5.865	7.213	46.067	5.955	7.029	96.089	10.093	11.675	**100.00**	**5.770**	**6.944**	98.315
135	7.246	8.772	51.124	8.047	9.739	84.746	10.407	11.800	**100.00**	**6.074**	**7.229**	94.241
180	8.009	9.712	50.595	8.679	10.173	86.127	10.239	11.841	96.703	**7.645**	**9.343**	**100.00**
240	11.075	12.105	47.126	10.574	12.686	83.333	11.749	12.756	**100.00**	**7.964**	**9.550**	**100.00**
300	10.000	11.473	48.634	9.918	11.819	87.791	10.164	12.019	**100.00**	**7.668**	**9.882**	99.471