INSTRUMENTATION AND INERTIAL NAVIGATION SYSTEMS DESIGN FOR TENSEGRITY ROBOT IMPLEMENTATIONS

Scott Edward Harper

West Virginia University, sharper9@mix.wvu.edu

Follow this and additional works at: https://researchrepository.wvu.edu/etd

Part of the Navigation, Guidance, Control and Dynamics Commons, Space Vehicles Commons, and the Systems Engineering and Multidisciplinary Design Optimization Commons

Recommended Citation

https://researchrepository.wvu.edu/etd/3682

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.
INSTRUMENTATION AND INERTIAL NAVIGATION SYSTEMS DESIGN FOR TENSEGRITY ROBOT IMPLEMENTATIONS

Scott E. Harper

Thesis submitted to the
Benjamin M. Statler College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements for the degree of
Master of Science
in
Aerospace Engineering

Yu Gu, Ph.D., Committee Chairperson
Jason Gross, Ph.D.
Powsiri Klinkhachorn, Ph.D.
Department of Mechanical and Aerospace Engineering

Morgantown, West Virginia
2018

Keywords: Tensegrity, State Estimation, INS, Zero Velocity Update, Avionics, Instrumentation, Navigation

Copyright © 2018 Scott Edward Harper
ABSTRACT

Instrumentation and Inertial Navigation Systems Design for Tensegrity Robot Implementations

by Scott E. Harper

One of the major challenges faced when developing missions for the exploration of planetary bodies is the risk these terrains pose on the science platform when using a traditional lander or wheeled rover. One means of developing platforms to traverse these harsh terrains is to utilize mobility systems comprised of tensegrity structures. These structures have the capacity to distribute loads across a network of axially loaded members such that they can be constructed in a very lightweight manner and morph their geometries when required. In literature, there has been significant progress in simulated environments to utilize tensegrity structures as mobility platforms, but the leap to real hardware has been a slow transition due to constrains in actuation design and state estimation. This thesis outlines a method of aiding an Inertial Navigation System deployed on a single strut of a tensegrity structure using zero velocity updates. The methods are implemented on a static icosahedran tensegrity structure to test the viability of the ZUPT-aided INS, how it can be improved, and other instrumentation configurations that can improve the results in future works.
I dedicate this thesis to my late grandfather, Noel Smith Justice.
Acknowledgments

I would like to thank my research adviser, and committee chair, Dr. Yu Gu for his guidance in this endeavor and the many doors he has helped me open along the way. I would also like to acknowledge my committee members Dr. Jason Gross and Dr. Powsiri Klinkhachorn for their advice throughout my graduate studies. Finally, I would like to thank the West Virginia Space Grant Consortium for granting me Graduate Research Fellowships for the 2016-17 and 2017-18 academic terms.
Contents

Abstract ii

Acknowledgments iv

List of Figures vii

1 Introduction 1

2 Tensegrity 3
  2.1 Origins and Literature Review ............................................. 3
  2.2 Tensegrity in Robotics ..................................................... 5

3 Inertial Navigation System 6
  3.1 Inertial Measurement Unit ................................................ 6
  3.2 INS Mechanization ........................................................... 6
  3.3 Zero Velocity Update ....................................................... 8
  3.4 ZUPT-aided INS .............................................................. 9

4 Avionics and Platform Design 10
  4.1 Generation VII Avionics .................................................... 10
  4.2 Applications of G7AP in Past Works .................................... 12
  4.3 Instrumentation and Platform Design for ZUPT-aided INS and Tensegrity State Es-
       timation ................................................................. 13

5 Experimental Methods and Results 16
  5.1 Data Collection Trails ..................................................... 16
    5.1.1 Boot Data Collection ................................................ 16
    5.1.2 Tensegrity Data Collection ......................................... 16
5.2 Results ................................................................. 17
  5.2.1 ZUPT Detector Tuning .......................................... 17
  5.2.2 Trajectories of ZUPT-aided INS ............................... 17

6 Conclusions and Future Works 21
  6.1 Conclusions .......................................................... 21
  6.2 Future Works ....................................................... 21

Appendices 27

A GENVII Avionic: Autopilot v1.3 Schematics 1
## List of Figures

1.1 NASA SUPERball Tensegrity Robot[1] ........................................... 2

2.1 Examples of Tensegrity structures in Art and Engineering ................... 4
2.2 Examples of Biotensegrity Models of Human Skeletal Structures ............ 4

4.1 GENVII Avionics: Autopilot and PC104 Companion Computer ................. 11
4.2 Logic flow of PWM channels. To pass through PWM signals generated by the Netburner for autopilot control, two digital signals must be activated. The first is the Autopilot engage signal from the pilot command interpreted by the ATtiny85, and the second is the channel-specific control command from the Netburner allowing for the ability to control specific actuators. For example, the autopilot may control only the surfaces, but leave the throttle command to the pilot’s discretion. ............. 12
4.3 Robotic platforms utilizing Generation VII Avionics PCBs .................... 13
4.4 Generation VII Avionics PCB designs ............................................. 14
4.5 Firefighter’s Boot instrumented with Autopilot PCB for IMU data collection. The Red arrows indicate the orientation of the PCB mounted IMU ..................... 15
4.6 Icosahedran Tensegrity Design used in this Thesis ............................... 15

5.1 ZUPT Threshold Tuning ............................................................. 18
5.2 ZUPT Threshold Tuning for Tensegrity ......................................... 19
5.3 Trajectory of Boot during first square ............................................ 20
5.4 Trajectory of Tensegrity ............................................................ 20

6.1 Concept Image of Tensegrity "Foot" .............................................. 22
6.2 Image recorded using Samsung Gear360 VR camera mounted on tensegrity strut .. 23
Chapter 1

Introduction

As humanity reaches further out into the cosmos with our unmanned probes, the idea of putting the full responsibility of an exploration mission on a single platform is not as scalable as man-kind’s curiosity. To counteract this, many research organizations are looking into new, innovative methods to distribute mission tasks across several smaller vehicles, either collaborative or independent, to pursue riskier science objective without critically risking mission success in a planetary exploration setting[2, 1, 3]. One such concept, proposed by Sunspiral et. al[1], from the NASA Ames Intelligent Robotics Group (IRG), is the utilization of tensegrity structures as the mobility platform. The IRG’s mission concept used the tensegrity robot as not only the planetary exploration vehicle, but also the landing device, reducing single-use mass such as parachutes or balloons typically of today’s missions[1]. IRG’s main test platform for this study was the SUPERball bot, shown in 1.1, which is an actuated tensegrity robot. However, control of this platform and many others has not been trivial and has been approached in a variety of ways in simulation and hardware [? , 4, 5, 6], but the physical implementations are always lacking in capability compared to their virtual counterparts. In the conclusions section of IRG’s 2015 NIAC Report[1], they stated that future works will look into new sensor technologies for improved state estimation and localization. This thesis outlines a method of improving this capability, with technologies already available on the SUPERball bot [1], through Zero Velocity Update-aided Inertial Navigation Systems using strut-mounted Inertial Measurement Units and investigate other instrumentation configurations with the potential to improve state estimation of tensegrity robots.
Figure 1.1: NASA SUPERball Tensegrity Robot[1]
Chapter 2

Tensegrity

2.1 Origins and Literature Review

Tensegrity, a concatenation of the words Tensile Integrity, describes a method of construction that develops structures by suspending a collection of discontinuous rigid elements in a network of continuous tensile cables. These configurations allow perturbations of the structure to affect all members creating morphological permutations. In their most common form, which is a structure comprised entirely of two-force members, creates a situation where each member only encounters axial loadings. This distributive property allows for these structures to be created with minimalistic material usage, and broadens the scope of possible materials to non homogeneous composites which are hard to implement in many mechanical applications. These unique structures, which began their history as sculpture, have been widely used in architecture, used to model structures in biology, and in recent years, robotics.

One of the first topics of interest was that of tensegrity geometric form-finding. Skelton et. al. [7, 8] provided a well accepted generalization of tensegrity classes and analytic tools for determining structural equilibrium, which are the conditions required by a tensegrity structure to for a stable volume. Several analytic methods can be used to develop topologies for what are known as "regular" tensegrity structures, which are typically comprised of one or more, known, symmetric, stable tensegrity patterns [7]. However, much research has been performed, with varying results to develop new methods of form-finding, typically using some form of genetic algorithm, with varied results in terms of success rate and computational complexity [9, 10, 11, 12, 13, 7].

Another topic of interest in recent literature is the controlled locomotion of tensegrity structures.
Many do this by altering the geometry to shift the center of gravity of the structure by changing the lengths of the tensile or compressive members [14, 1, 15, 16] or by moving an internal payload [1]. The control of these motions are also an active field of research with most works utilizing genetic algorithms to breed a collection of Central Pattern Generators to create actuator commands [17, 6, 1, 14] or by deploying reinforcement learning directly on actuator controls [18, 19, 20].
2.2 Tensegrity in Robotics

Utilizing the body of a robot to offset the computational load of a digital controller has been increasing in interest among the robotics community as research interests have transitioned from precise control to adaptability. The idea is that the intrinsic dynamics of a mechanical structure interacting with its environment can have emergent intelligence on its own [21]. That is, a physical body can perform computations to achieve goals such as manipulation and locomotion without the need for a controller [21]. This idea of using these analog computations using a robots physical body has been termed, morphological computation [22, 23]. Paul et al. [22] demonstrate how one can perform elementary logic operations with mechanical systems rather than their electronic counterparts. Likewise, Pfeifer et al. [23] provide an excellent example how a robotic fish, with only one degree of actuation on its tail, can perform a variety of maneuvers with just slight variations of how the tail wiggles. Instead of requiring additional fins for control maneuvers, increasing complexity to the controller, it exploits the environment it is interacting with through morphological computation [23].

A tensegrity is a concept that aligns easily with this idea of morphological computation and can be witnessed in multitudes of natural and artificial examples. Many have been utilizing tensegrity structures in robotics applications in the hopes of creating highly adaptive mobility platforms. Researchers at the Intelligent Robotics Group, out of NASA Ames, have been looking to use Tensegrity structures as a means of combining the lander and the rover for future planetary exploration missions. The unique strength properties of tensegrities would allow them to not only survive the impact of landing, but explore areas of interest that would be deemed too risky for traditional wheeled rovers. Most research on tensegrity locomotion has been performed using the icosahedron geometry [1, 17, 6, 4, 5], which is the mechanically simplest spherical tensegrity structure. While the controlled locomotion of these structures has been an active topic, as discussed in the last section, deploying these new control actions on hardware has been a challenge. As stated in the closing remarks of the final report for the Phase II NIAC SUPERball project the researchers involved had this to say about next steps for future works,”Our first steps will be to continue exploring sensor technologies that can provide critical information needed for state estimation and localization”[1]. This states a need for better state estimation, a topic that is yet still lacking in literature with relatively few publications directly addressing the issue [17, 24].
Chapter 3

Inertial Navigation System

3.1 Inertial Measurement Unit

At the heart of any INS is the IMU. The most basic of IMUs have accelerometers to measure specific forces, and gyroscopes to measure angular rates of a body. Many IMUs have additional sensors collecting useful information for navigation such as magnetic fields, altitude, and temperature. However, for this work will focus on utilizing the former style which provide 6 DOF, 3-axis acceleration and 3-axis angular rates, rigidly mounted to the body of the platform.

3.2 INS Mechanization

The INS in this report will use a typical strap-down mechanization, following the implementation presented in [25]. The IMU reports angular rates, \( \omega \), and specific forces, \( a \), at a fixed timestep, \( t \). Multiplying these reported values over the known timestep will give us the change in orientation and velocity, \( \delta \phi \), \( \delta \omega \), \( \delta r \), \( \delta v \), and \( \delta a \) respectively. These deltas are then used to update the attitude, velocity, and position states. To compliment the IMU updates, an Extended Kalman Filter (EKF) monitoring the error states of the system to account for drift introduced during integration. The error states are represented by \( \delta x = [\phi \delta \omega \delta r \delta v \delta a] \), where \( \phi \) are the attitude errors, \( \delta \omega \) are gyroscope biases, \( \delta r \) are position errors, \( \delta v \) are velocity errors, and \( \delta a \) are accelerometer biases. For each cycle, the filter produces error estimates that are accounted for in the INS, then clears the error state vector.
δx back to zero. The state transition matrix for the discrete-time dynamics model is

\[
\Phi = \begin{bmatrix}
I & -\Delta t C^n_b & 0 & 0 & 0 \\
0 & I & 0 & 0 & 0 \\
0 & 0 & I & \Delta t I & 0 \\
-\Delta t S(f^n) & 0 & 0 & I & \Delta t C^n_b \\
0 & 0 & 0 & 0 & I
\end{bmatrix}
\] (3.1)

The rotation matrix \( C^n_b \) transforms vectors from the body frame, or b-frame, to the navigation frame, or n-frame. This transformation is needed due to the IMU reporting values in the body frame of the platform, but navigation values such as velocity and position will be represented in the n-frame. The \( S(f^n) \) expression represents the skew-symmetric matrix

\[
S(f^n) = \begin{bmatrix}
0 & -f_z & f_y \\
f_z & 0 & -f_x \\
-f_y & f_x & 0
\end{bmatrix}
\] (3.2)

created from the n-frame accelerometer output vector \( f^n = C^n_b (\Delta v^b \Delta t) \). This allows for attitude errors to propagate in the n-frame allowing for corrections based on n-frame velocity and position estimates.

We start the Kalman filter process by propagating the error covariance matrix with the following equation:

\[
P_{k+1|k} = \Phi_k P_{k|k} \Phi_k^T + Q_k
\] (3.3)

Where \( P_{k+1|k} \) is the estimation error covariance matrix at time \( k+1 \) based on measurements received through time \( k \), and \( Q_k \) is the process noise covariance matrix. In this case, we do not need to propagate the states with the time update equation \( \delta x_{k+1} = \Phi_k \delta x_k \) because the EKF will transfer the error states to the INS and reset them to zero after a measurement. If we take the measurement equation measurement \( z = h(x) + v \) where \( h(x) \) is the measurement function of the states represented in \( (x) \) and \( (v) \) is additive, white, zero-mean Gaussian noise with covariance matrix \( R \), and the current estimated states of the INS are \( \hat{x} = x + \delta x \) then

\[
z = h(\hat{x} - \delta x) + v = h(\hat{x}) - H\delta x + v
\] (3.4)
where $H$ is the Jacobian $\frac{\partial h}{\partial x}$. Thus, the measurement model for the error-state filter will be

$$v = -H\delta x + v \quad (3.5)$$

where the INS innovation $v \equiv z - h(\hat{x})$ acts as the measurement for the error-state filter. After a measurement, the EFK then updates the state with

$$\delta x_{k+} = \delta x_{k} + K(\nu - (-H)\delta x_{-}) = K\nu \quad (3.6)$$

Where the Kalman gain $K$ is calculated using

$$K = P(\nu)^T P((-H)^T P (-H) + R)^{-1} \quad (3.7)$$

Finally, the covariance matrix is updated using

$$P_{k+1|k+1} = (I - K(-H))P_{k+1|k+1}(I - K(-H))^T + KRK^T \quad (3.8)$$

### 3.3 Zero Velocity Update

A common method for reducing INS drift over time is to implement a Zero Velocity Update (ZUPT). ZUPTs are very common in pedestrian navigation systems where at some point in the motion it can be inferred that there are regular intervals of no translational moment. In the case of pedestrian navigation, for example, there are instances in the walking gate where the foot is resting on the ground and has no translational movement. When a zero-velocity update is initiated, it is assumed that the velocity during the period the foot is stationary, is zero. Thus, the velocity states in the state estimator are reset to zero. This in turn eliminates any error accumulated in the velocity states between ZUPTs. To implement these updates, there must first be a method to detect when these moments of zero velocity. There are a variety of methods used to build zero velocity detectors in literature, but the method that will be implemented in this study, based on the evaluation study performed by Skog, et al [26], will be the Angular Rate Energy (ARE) detection method due to its computation simplicity and performance compared to other similar detection methods also review in [26]. The implementation of the detector is a binary hypothesis testing problem, where we are testing to see if the IMU is in one of two states, moving or stationary. In the following statements the nomenclature used will also follow that of [26]. The IMU is determined to be stationary if
\( T(z_n^\omega) < \gamma \). Here, \( n \) is the time instant, \( z_n^\omega \) is the measurement sequence of the IMUs gyroscopes, \( T(z_n^\omega) \) represents the quantification metric for the detector implementation, and \( \gamma \) is the detection threshold. The detector metrics are then represented as

\[
T(z_n^\omega) = \frac{1}{\sigma^2_\omega W} \sum_{k=n}^{n+W-1} \|y_k^\omega\|^2
\]

(3.9)

Here, \( \sigma^2_\omega \) is the noise variance of the gyroscopes, \( W \) is the number of observations to be considered, and \( y_k^\omega \) is the measured angular rate vector from the gyroscopes. This detector is tuned using the \( W \) and \( \gamma \) parameters. Increasing the observation window will reduce false detections but reducing the detector response time. Modifying the detection threshold sets the noise tolerance on the detector, which can allow for some leniency for underdamped, residual vibrations that can occur when coming to a stop after an impact.

### 3.4 ZUPT-aided INS

The implementation of the zero-velocity update will also follow the procedures used by [25]. When a stationary period is detected, the update procedure in the EKF is modified to take into considerations the extra assumptions associated with the ZUPT. During a ZUPT we assume a velocity pseudo measurement of \( z = [0\ 0\ 0]^T \) causing the velocity component of our INS innovation, \( \nu \), to be the negative of the most current velocity. With \( H = [0\ 0\ 0\ I_0] \), to select the states corresponding to the velocity, the covariance matrix is set to

\[
R = I \times \max(\text{trace}(HPH^T), (1 \text{ mm/sec}^2))
\]

(3.10)

to avoid instability that could occur during such a sudden reduction in covariance.
Chapter 4

Avionics and Platform Design

This chapter will outline the design of WVU IRL’s Generation VII Avionics: Autopilot, which is used by this research, as well as many other tasks at IRL.

4.1 Generation VII Avionics

The design of and utilization WVU’s avionics system for sub-scale fixed-wing UAVs began through efforts made by the WVU Flight Control Systems Lab, and has continued to evolve and be used by IRL and the WVU Navigation Laboratory. Historically, the avionics system has comprised of multiple PCBs collaborating to collect sensor data, process information, and distribute actuator commands. Each iteration of the avionics system allowed for a wider selection of instrumentation options, faster processing, and more efficient packaging.

The Generation VII Avionics system had an additional design goal over its predecessors: the ability to be used by many robotic vehicle configurations, including, but not limited to: fixed-wing UAVs, multi-rotor UAVs, and rovers. To meet these goals, the flagship PCB design for the Generation VII Avionics, the Autopilot (G7AP) was developed. This was made possible due to the highly configurable Netburner MOD54415 microcontroller used as the main processing unit of the Autopilot. The MOD54415 has a 250 MHz processor and a wide range of I/O options including: 10/100 Mbps Ethernet, 8 UART, 4 I2C, 2 CAN, 3 SPI, 42 GPIO, 12 ADC, 2 DAC, and 5 PWM. This vast flexibility in connectivity makes it the perfect candidate for the general-use platform the Autopilot was designed to be. The PCB design breaks out many of these I/O for pre-allocated uses such as the IMU interface, companion computer connectivity, and PWM outputs for motor commands, as well as many extra ports, such as GPIO, UART, and ADC, for expanded
The PCB was designed using Altium Designer, the full schematic can be found in A. It consisted of a 4 layer design with two signal layers and two power planes. The PCB had onboard power regulation using switching Traco Power dc-dc converter modules that can be come in a variety of input voltages. These regulators also vary in the power output capacity, allowing for up to 30 watts of power available for external devices. This style of modular regulation also allows for the design to be flexible with the power source, and is compatible with most battery technologies used in robotics. The G7AP can also be powered via the USB debug port if it does not have to support any other devices that are not on-board components. The PCB has an on-board IMU interface that is compatible with many Analog Devices IMUs such as the ADIS16490 and similar models. For other external devices it breaks out several common interfaces such as ADC/DAC, GPIO, PWM, as well as TTL and RS-232 logic levels for UART serial communication. Other standards can also be achieved using external circuitry.

For vehicle safety, there is intelligent Autopilot to Human-pilot RC switching circuitry to ensure that if a pilot needs to take control of the vehicle away from any Autopilot software with commands generated by the Netburner Module, they can do so. There is an additional ATtiny85 microprocessor on the PCB monitoring a specific RC channel, dedicated to switching the autopilot control on and off, regardless of any processes happening on the main processing unit. This ATtiny85 will activate a set of AND logic gates, with the other input coming from the Netburner requesting to command specific motor outputs, illustrated in Figure 4.2. Unless the AND gates are receiving TRUE signals from both the Netburner requesting access to a specific motor, and the ATtiny determining the pilot is relinquishing control, the motor commands will be directly coming from the RC receiver.
Figure 4.2: Logic flow of PWM channels. To pass through PWM signals generated by the Netburner for autopilot control, two digital signals must be activated. The first is the Autopilot engage signal from the pilot command interpreted by the ATtiny85, and the second is the channel-specific control command from the Netburner allowing for the ability to control specific actuators. For example, the autopilot may control only the surfaces, but leave the throttle command to the pilot’s discretion.

This default action of passing through the pilot commands works even if power is removed from the Autopilot, so the vehicle can be remotely controlled even during a power failure of the PCB.

4.2 Applications of G7AP in Past Works

Since development of the G7AP, it has been involved in a variety of applications. Initially, it was installed on a Quad-rotor UAV used in the study conducted in [27]. This low cost platform, as shown in 4.3(a), was the perfect development environment for testing earlier iterations of the PCB, and work out any design flaws. It was later used to modernize the avionics payload of the flight of Phastball UAVs, 4.3(b), used to conduct research operations at WVU such as in [28]. It was used to in ground vehicles as well, including the Mountaineer Mining Vehicle, pictured in 4.3(c), that was entered into the 2017 NASA Robotic Mining Competition.

After several successful implementations of the G7AP, a line-up of companion, and spin-off, designs were developed for more platform-specific implementations. For the Phastball UAVs, a PCB was developed to be installed in the nose cone of the vehicle, as seen in 4.4(a) to house a Novatel GPS receiver, RF modem for ground station communications, pressure sensors, wind vanes, and a temperature/humidity sensor. an additional Netburner MOD5213 was included on the PCB to collect sensor data then either log it to an SD or and communicate the information an accompanying Autopilot or companion computer. A complete overview of the Phastball electronics package is explained in further detail in [28]. Another Autopilot-inspired design was the Navigation Sensor
Figure 4.3: Robotic platforms utilizing Generation VII Avionics PCBs

Interface (G7NRI), 4.4(b), used by the Cataglyphis rover, 4.3(d), for WVUs entry into the Sample Return Robot Challenge (SSRC) [29, 30]. This PCB included that capacity to house six tactile-grace IMUs, interact with a Velodyne 3D lidar, and facilitate required robot features dictated by the challenge rules such as the remote pause switch and hazard light. Following the SRRC, a new PCB was designed combining features of both the G7AP and G7NRI for use on future ground-vehicle, or large aerial UAV designs. This new PCB, named the Autopilot+, shown in 4.4(c), contained all the features included in the previous Autopilot PCB, but also replaced the MOD54415 with a Netburner NANO54415 module, and included the 3D lidar and remote pause interfaces included on the G7NSI.

4.3 Instrumentation and Platform Design for ZUPT-aided INS and Tensegrity State Estimation

For the procedure carried out in the later chapters of this document, two hardware platforms were developed. The first, used for collecting IMU data for zero-velocity detector development and ZUPT-aided INS implementation on a system with periodic intervals of rest, was a firefighter’s boot outfitted with a G7AP, as shown in 4.5. The PCB was powered using a USB battery pack for charging mobile
Figure 4.4: Generation VII Avionics PCB designs

devices, and included a toggle switch for indicating when to log data. The IMU used was an analog devices ADIS16485, collected at a rate of 100 Hz, and saved on the on-board SD card of the MOD54415. When the toggle was switched to the LOG position, a new log file would be created and IMU data would begin to be recorded to the new file. When the switch is toggled back to the OFF state, the data log file is saved and the MOD54415 will wait until the switch is toggled again.

For a tensegrity platform, it was desired to have a design that was large enough to mount several devices to as well as sturdy enough to withstand some rather harsh testing, such as rapidly rolling down a hill. The final mechanical design is shown in 4.6. It is comprised of a collection of 4 foot, 1-inch diameter, aluminum tubes suspended by 1/4 inch steel cables. Then, to instrument the structure, the same G7AP setup conceived for the boot is attached to a strut.
Figure 4.5: Firefighter’s Boot instrumented with Autopilot PCB for IMU data collection. The Red arrows indicate the orientation of the PCB mounted IMU.

(a) Exploded view of mechanical design of Strut Endcap
(b) Tensegrity Structure used for Data Collection

Figure 4.6: Icosahedran Tensegrity Design used in this Thesis
Chapter 5

Experimental Methods and Results

5.1 Data Collection Trails

As mentioned in chapter 4, the G7AP is the main data collection device used for gathering inertial data in this study. The IMU used is the ADIS16485 by Analog devices, and is captured at a rate of 100 Hz. To validate our INS solution, a Vicon External motion capture system was used to represent our ground truth.

5.1.1 Boot Data Collection

For the first data collection campaign, conducted using the boot platform seen in Figure 4.5 trails were conducted in three configurations: stationary, square, and irregular. For the stationary IMU data configuration, the boot was left in a single location undisturbed for 20 minutes, no motion capture information was collected for this trial and the change in position and orientation was assumed zero. The second trial collected data continuously for an hour walking in a two-meter square for 1 hour, followed by a short 5-minute period of irregular movements.

5.1.2 Tensegrity Data Collection

For the second set of experiments, the tensegrity structure, shown in Figure 4.6 was used. One of the struts was then outfitted with the Autopilot PCB used on the boot platform. The first trial conducted on this platform involved slowly transitioning the object, by hand, to different triangles of the structure. The second trial involved more forceful disturbances, throwing the structure from one side of the room to the other, followed by short periods of rest to allow for vibrations to settle.
5.2 Results

5.2.1 ZUPT Detector Tuning

To gain an initial understanding of the zero-velocity detector, the minimal threshold and window size were found so that on the stationary dataset of the boot, the ARE detector was always determining that the body was stationary. A window size of 3 was selected so that the threshold term could be used as the sole tuning parameter. Through trial and error, the threshold term was found to be stable at a value of 25. After the initial settings were decided upon, the detector was tested on a segment of the larger 1-hour trail to see if the ZUPT updates were triggered at the correct moments in the gait. Initially, the Zero velocity detector did not detect any of the resting moments. This is likely due to small residual motion in the system after each step, and slight movements. Thus, a second round of tuning was required for a walking experiment. In Figure 5.1, you can see the results of different values for the ARE detector. The initial value of never detected a rest period as explained before. The 10,000 threshold was characterized the resting period well, but triggered ZUPTs when there were still obvious vibrations in the system after a step and periods with slight movements about one or more axis. The 1000 threshold was the best compromise as it triggered only when the system was truly at rest. Also, as expected the parameters for the boot also did not immediately work on the Tensegrity structure. However, instead of increasing the threshold, it had to be decreased significantly in order to cancel out oscillations within the structure. With a threshold of just 1, an acceptable zero-velocity response was achieved. As seen in Figure 5.2, There are points in the trajectory where a ZUPT can be performed, but due to the more irregular intervals in the motion, they are triggered less often than on the boot platform. On the aggressive dataset, the same threshold had an acceptable response, but was given even fewer opportunities to achieve zero-velocity due to oscillations in the system.

5.2.2 Trajectories of ZUPT-aided INS

When the tuning was complete the trajectories of the INS without ZUPT and with were compared. However, the EKF was unstable without the use of ZUPT, and at the time of writing this report has not been resolved. However, there was stability when ZUPTs were taken into consideration, as seen in Figure 5.3. There is significant drift in the z and yaw, more so than expected. This may indicate some deficiency in the sensor modeling in the EKF implementation and/or misalignment in the sensor axis.
Figure 5.1: ZUPT Threshold Tuning
Figure 5.2: ZUPT Threshold Tuning for Tensegrity
After the Boot trials, The ZUPT-aided INS was implemented on the tensegrity datasets. And the instabilities due to less frequent ZUPTs in the INS are very apparent in the trajectories in the Tensegrity trials, as seen in Figure 5.4. The first slow experiment started to diverge quite early, and the aggressive experiment was completely unstable.
Chapter 6

Conclusions and Future Works

6.1 Conclusions

This thesis showed a potential implementation of a ZUPT-aided INS on a tensegrity platform to improve state estimation performance. The approach of using ZUPT-aided INS was shown to be plausible based on the data set collected using the boot platform, as seen in the ZUPT detector tuning of Figure 5.1 and trajectory tracking shown in Figure 5.3. However, using the tensegrity configuration demonstrated in chapter 5 is not advised due to the infrequency of moments of zero velocity. Due to sensor placement, and the internal under-damped dynamics of struts creating oscillations up to tens of seconds after a disturbance, as seen in Figure 5.2, the IMU rarely encounters an instant of rest in which to initiate a zero velocity update.

6.2 Future Works

To address the shortcomings displayed in this thesis, future works will proceed as follows: modify sensor placement and mechanical design for improved ZUPT, and look into different sensors and sensor-fusion techniques to further improve the state estimate capabilities for tensegrity robots.

To improve Zero-velocity Update performance, future design will incorporate "feet" which will be passive platforms with a pivot located at the end-cap of each strut. These feet will lay flat on the ground and allow the strut to move freely in relation to it through a ball-joint. This will provide a platform to house an IMU that will have periods of zero velocity much more frequently than an IMU mounted mid-strut. A preliminary design can be seen in Figure 6.1.

Another set of instruments of interest would be the incorporation of visual systems for visual
odometry and visual slam purposes. The vision systems can be combined with the INS through the use of a Kalman filter for form a Visual-Inertial Odometry (VIO) to add further corrections in the localization estimates in addition to the ZUPT. Preliminary video was recorded at the same time of the tensegrity trials in chapter 5 for future use. The camera used was a Samsung Gear360 camera for virtual reality (VR) recordings. The benefit to using this VR camera is that it combines images of two cameras with fisheye lenses to create a view of the entire area surrounding the camera, as seen in fig:POV. For future development, QR codes were mounted on selected struts in view of the camera with the idea that the pose of each QR code can be estimated and used to reconstruct the pose of a particular strut in relation to the camera.
Figure 6.2: Image recorded using Samsung Gear360 VR camera mounted on tensegrity strut
Bibliography


[27] Caplinger, T. W., Path Planning and Control of an Autonomous Quadrotor Testbed in a Cluttered Environment, West Virginia University, 2015.


Appendices
Appendix A

GENVII Avionic: Autopilot v1.3

Schematics