Theses and Dissertations

2015-03-01

UAV Navigation and Radar Odometry

Eric Blaine Quist
Brigham Young University - Provo

Follow this and additional works at: https://scholarsarchive.byu.edu/etd

Part of the Electrical and Computer Engineering Commons

BYU ScholarsArchive Citation
https://scholarsarchive.byu.edu/etd/4439

This Dissertation is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.
UAV Navigation and Radar Odometry

Eric Blaine Quist

A dissertation submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Randal W. Beard, Chair
Timothy W. McLain
Karl F. Warnick
Dah Jye Lee
David G. Long

Department of Electrical and Computer Engineering
Brigham Young University
March 2015

Copyright © 2015 Eric Blaine Quist
All Rights Reserved
ABSTRACT

UAV Navigation and Radar Odometry

Eric Blaine Quist
Department of Electrical and Computer Engineering, BYU
Doctor of Philosophy

Prior to the wide deployment of robotic systems, they must be able to navigate autonomously. These systems cannot rely on good weather or daytime navigation and they must also be able to navigate in unknown environments. All of this must take place without human interaction.

A majority of modern autonomous systems rely on GPS for position estimation. While GPS solutions are readily available, GPS is often lost and may even be jammed. To this end, a significant amount of research has focused on GPS-denied navigation. Many GPS-denied solutions rely on known environmental features for navigation. Others use vision sensors, which often perform poorly at high altitudes and are limited in poor weather. In contrast, radar systems accurately measure range at high and low altitudes. Additionally, these systems remain unaffected by inclimate weather.

This dissertation develops the use of radar odometry for GPS-denied navigation. Using the range progression of unknown environmental features, the aircraft’s motion is estimated. Results are presented for both simulated and real radar data.

In Chapter 2 a greedy radar odometry algorithm is presented. It uses the Hough transform to identify the range progression of ground point-scatterers. A global nearest neighbor approach is implemented to perform data association. Assuming a piece-wise constant heading assumption, as the aircraft passes pairs of scatterers, the location of the scatterers are triangulated, and the motion of the aircraft is estimated. Real flight data is used to validate the approach. Simulated flight data explores the robustness of the approach when the heading assumption is violated.

Chapter 3 explores a more robust radar odometry technique, where the relatively constant heading assumption is removed. This chapter uses the recursive-random sample consensus (R-RANSAC) Algorithm to identify, associate, and track the point scatterers. Using the measured ranges to the tracked scatterers, an extended Kalman filter (EKF) iteratively estimates the aircraft’s position in addition to the relative locations of each reflector. Real flight data is used to validate the accuracy of this approach.

Chapter 4 performs observability analysis of a range-only sensor. An observable, radar odometry approach is proposed. It improves the previous approaches by adding a more robust R-RANSAC above ground level (AGL) tracking algorithm to further improve the navigational accuracy. Real flight results are presented, comparing this approach to the techniques presented in previous chapters.

Keywords: Radar navigation, GPS-Denied navigation, Kalman filter, RANSAC, SAR
ACKNOWLEDGMENTS

This dissertation is dedicated to my children: Paige Alice Quist, Benson Blaine Quist, Everett Dale Quist, and Coleman Spencer Quist. While they had no say in my educational pursuits, they have far too often gone without the time, attention, and patience of their Dad.

I thank my wife, Melinda Kay Quist, for her patience and long-suffering throughout the entire process. She has carried the majority of the burden, while I have received a majority of the credit.

I also thank my Savior Jesus Christ. Without him I am nothing.

The list of others who have helped me throughout this process is too long to enumerate here. Specifically, I’d like to thank IMSAR for their flexibility and specifically Dr. Bryce Ready for his help. I would also like to thank those at BYU who have helped me along the way, specifically Dr. Randal Beard for his help and patience and Peter Niedfeldt for his insight.
Contents

List of Tables vii

List of Figures viii

1 Introduction 1
   1.1 Summary of Contributions ................................................. 2
   1.2 Literature Review .......................................................... 3
      1.2.1 GPS Denied Navigation .............................................. 4
      1.2.2 Optical Sensors and Visual Odometry ............................ 4
      1.2.3 Laser Range Finders ................................................. 5
      1.2.4 Radar Beacons ......................................................... 6
      1.2.5 Sonar ................................................................. 6
      1.2.6 Radar ................................................................. 6
         1.2.6.1 Resolving Bearing to Scatterers ............................. 7
         1.2.6.2 Navigation with Radar ........................................ 7
      1.2.7 Mapping ............................................................... 8
         1.2.7.1 SLAM ........................................................... 8
         1.2.7.2 View-Based Maps .............................................. 9
   1.3 Literature Summary ...................................................... 10

2 Radar Odometry on Small Unmanned Aircraft - A Greedy Approach 11
   2.1 Radar Range Compression .............................................. 12
      2.1.1 Radar Range and LFM-CW Radar .................................. 12
      2.1.2 Range to Scatterer During Flight ................................ 15
   2.2 Radar Odometry ........................................................... 17
      2.2.1 AGL Estimation ....................................................... 19
      2.2.2 Range-Compressed Image Pre-Filter .............................. 19
      2.2.3 Scatterer Identification ........................................... 21
      2.2.4 Range Estimation .................................................... 24
      2.2.5 Relative Drift Estimation ........................................ 25
   2.3 Extended Kalman Filter ................................................ 28
      2.3.1 Prediction Model .................................................... 29
      2.3.2 Measurement Model ................................................ 32
   2.4 Results ................................................................. 33
      2.4.1 Simulation Results .................................................. 34
      2.4.2 Flight Test Results ................................................ 36
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2.2</td>
<td>Scatterer Track Filtering</td>
<td>79</td>
</tr>
<tr>
<td>4.2.2.1</td>
<td>Increasing Scatterer Measurements</td>
<td>80</td>
</tr>
<tr>
<td>4.2.2.2</td>
<td>Identifying and Removing Clutter</td>
<td>81</td>
</tr>
<tr>
<td>4.2.3</td>
<td>AGL estimation with R-RANSAC</td>
<td>82</td>
</tr>
<tr>
<td>4.3</td>
<td>EKF and the Applied State-Space Model</td>
<td>84</td>
</tr>
<tr>
<td>4.3.1</td>
<td>System State</td>
<td>85</td>
</tr>
<tr>
<td>4.3.1.1</td>
<td>Aircraft State</td>
<td>85</td>
</tr>
<tr>
<td>4.3.1.2</td>
<td>Point Scatterer State</td>
<td>86</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Prediction Model</td>
<td>86</td>
</tr>
<tr>
<td>4.3.2.1</td>
<td>Aircraft Dynamics</td>
<td>87</td>
</tr>
<tr>
<td>4.3.2.2</td>
<td>Scatterer State Dynamics</td>
<td>89</td>
</tr>
<tr>
<td>4.3.2.3</td>
<td>IMU Dynamics</td>
<td>89</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Update Sensor Dynamics</td>
<td>90</td>
</tr>
<tr>
<td>4.3.3.1</td>
<td>GPS</td>
<td>90</td>
</tr>
<tr>
<td>4.3.3.2</td>
<td>Magnetometer</td>
<td>91</td>
</tr>
<tr>
<td>4.3.3.3</td>
<td>Coordinated Turn</td>
<td>92</td>
</tr>
<tr>
<td>4.3.3.4</td>
<td>AGL Measurement</td>
<td>92</td>
</tr>
<tr>
<td>4.3.3.5</td>
<td>Scatterer Range Measurements</td>
<td>93</td>
</tr>
<tr>
<td>4.3.3.6</td>
<td>Relative Range Measurement</td>
<td>93</td>
</tr>
<tr>
<td>4.3.3.7</td>
<td>Range Rate</td>
<td>94</td>
</tr>
<tr>
<td>4.4</td>
<td>Results</td>
<td>95</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Simulated Results</td>
<td>95</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Real Results</td>
<td>97</td>
</tr>
<tr>
<td>4.4.3</td>
<td>Conclusion</td>
<td>99</td>
</tr>
<tr>
<td>5</td>
<td>Conclusion</td>
<td>100</td>
</tr>
<tr>
<td>5.1</td>
<td>Summary</td>
<td>100</td>
</tr>
<tr>
<td>5.2</td>
<td>Future Work</td>
<td>101</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Remove system constraints</td>
<td>101</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Explore the Use of Traditional Radar Algorithms</td>
<td>102</td>
</tr>
<tr>
<td>5.2.2.1</td>
<td>Ground Motion Tracking Indicator (GMTI) Algorithms</td>
<td>102</td>
</tr>
<tr>
<td>5.2.2.2</td>
<td>SAR Image Formation Algorithms</td>
<td>102</td>
</tr>
<tr>
<td>5.2.3</td>
<td>Mapping</td>
<td>103</td>
</tr>
<tr>
<td>5.2.4</td>
<td>Real Flight Testing Using Known and Unknown Scatterers</td>
<td>103</td>
</tr>
<tr>
<td>5.2.5</td>
<td>Estimate Pitch and Roll from the Radar Return</td>
<td>103</td>
</tr>
<tr>
<td>5.3</td>
<td>Discussion</td>
<td>103</td>
</tr>
</tbody>
</table>

Bibliography: 105
List of Tables

2.1 Position error for Monte-Carlo simulations as a function of different commanded heading changes. .................................................. 35

4.1 Comparison of previous radar odometry approaches using the same data set. The unaided IMU, greedy RO algorithm [1], the RO with R-RANSAC algorithm [2], and the enhanced results presented in this chapter (all results are from real flight data). The greedy RO algorithm did not require a Monte Carlo run for their real results, as the Hough transform exhaustively considers all data, whereas both the RO with R-RANSAC and these results each ran 10 Monte Carlo runs. ............ 99
## List of Figures

2.1.1 Range-compressed image of radar return during a straight and level aircraft flight. .................................................. 15
2.1.2 The geometry that shows how a ground scatterer is displayed in the range-compressed image. .................................................. 16
2.2.1 Block diagram outlining the aircraft state estimation using an IMU and the optional radar odometry algorithm. .................................................. 18
2.2.2 One column of the range-compressed image, showing the dramatic increase in signal and noise floor beyond the range associated with nadir. The peaks prior to nadir are noise in the radar itself. .................................................. 19
2.2.3 Magnified segment of the range-compressed image of radar return during an aircraft flight. .................................................. 20
2.2.4 Range-compressed image after pre-filtering. .................................................. 21
2.2.5 The Hough-space image $I_{V_g}$ for a fixed $\hat{V}_g$, with gray-scale indicating the number of votes. .................................................. 22
2.2.6 Thresholded Hough-space image $I_m$. .................................................. 24
2.2.7 Pre-filtered range-compressed image (shown in gray) with super-imposed hyperbola estimates shown in black. .................................................. 24
2.2.8 Pre-filtered range-compressed image (shown in gray) with super-imposed measured range values in black. .................................................. 25
2.2.9 Geometry associated with calculating the relative drift from the range-compressed image. .................................................. 27
2.4.1 Average position error when the commanded to fly three flight paths: straight, 10 degree change in heading, and 40 degree change in heading. The average IMU and RO+IMU drift error is shown for all changes in heading. .................................................. 34
2.4.2 Post-processed SAR image of the scatterers as seen by the aircraft. .................................................. 36
2.4.3 Flight results using a navigation-grade IMU. .................................................. 37
2.4.4 Flight results using an emulated commercial-grade IMU, note the scale change as compared to Figure 2.4.3. .................................................. 37
3.1.1 Radar odometry block diagram .................................................. 41
3.1.2 Range-compressed image from a real flight. The bright hyperbolas in the image are due to the range progression of strong point scatterers. The dark portion at the top of the image indicates no objects reflecting at that range. At nadir the return power drastically increases as ground scatterers reflect the radar signal back to the aircraft. .................................................. 42
3.1.3 Accurate association and tracking for real range-compressed image. The real range-compressed image is shown in gray, while the identified and tracked features are shown in black.

3.1.4 Errors associated with incorrectly distinguishing clutter and features (3.1.4a), track drift (3.1.4b), and feature identification (3.1.4c). All show a real range-compressed image gray, while the identified and tracked features are shown in black.

3.1.5 Simulated error between truth and tracked measurements (3.1.5a) and the number of simulated chirps where the feature was not tracked (3.1.5b).

3.2.1 Azimuth angle and range relative to the aircraft’s vehicle frame.

3.3.1 Radar odometry simulated flight error. The solid black line indicates the drift error with the navigation-grade IMU alone, while the blue line indicates the average of 10 Monte Carlo results. The best and worst results are shown in red and green respectively.

3.3.2 Processed image formed from real flight data.

3.3.3 Radar odometry real flight error. The solid black line indicates the drift error with the navigation-grade IMU alone, while the blue line indicates the average of 10 Monte Carlo results. The best and worst results are shown in red and green respectively.

4.2.1 R-RANSAC based radar odometry block diagram.

4.2.2 (4.2.2a) shows all point scatterers track measurements after being labeled as a track using the R-RANSAC algorithm. (4.2.2b) shows the tracks, while adding the inlier measurements that occurred prior to track labeling.

4.2.3 The results of the track filtering, plotting all measurements and tracks used by the EKF.

4.2.4 (4.2.4a) shows a range-compressed single chirp and shows that the return power drastically increases at nadir. (4.2.4b) shows a range-compressed image. There is clutter or point scatterers closer than nadir throughout much of the image (specifically between 10 and 40 second and again between 48 and 55 seconds).

4.2.5 The AGL error for a thresholding approach and a R-RANSAC estimation approach. The thresholded estimation error is in blue, while the R-RANSAC estimation error is lower and more consistent, as is shown in green.

4.3.1 Azimuth angle relative to the aircraft’s vehicle frame.

4.4.1 Average drift error for simulated flight as the aircraft underwent varying commanded banking angles (each with 36 Monte Carlo runs).

4.4.2 Simulated GPS-denied navigational error over a prolonged 10 minutes simulated flight. The blue, solid line is the IMU drift and the green, dotted line is the drift error of the radar odometry algorithm.

4.4.3 Position estimation error from real flight with 10 Monte Carlo runs.

4.4.4 Comparison of real flight results (shown as a solid line) and simulated flight (shown as a dashed line). The IMU results are shown in bold, while the radar odometry results with IMU are shown with a narrow line.
Chapter 1

Introduction

Autonomous navigation is predicated on the unmanned vehicle’s ability to estimate its position as it navigates its environment. Typical solutions use an inertial measurement unit (IMU) to measure the acceleration and rotation of the vehicle, in conjunction with a global positioning system (GPS) sensor to compensate for drift and error resulting from the IMU. Such solutions are very accurate but cannot operate when GPS signal is lost or jammed.

A variety of approaches attempt to provide GPS-denied navigation. Some systems, such as those employed on intercontinental ballistic missiles, rely exclusively on highly precise IMUs for navigation. These systems navigate accurately, but the large size, weight, power, and cost renders them intractable for commercial small unmanned aircraft vehicles (UAVs). Accordingly, most approaches rely on exteroceptive sensors to mitigate the drift of smaller IMUs.

A single optical sensor may be used to measure the bearing to environmental visual features. The accuracy of these systems is a function of the pixel count and the focal length of the camera. As the range to a feature increases, the azimuth and/or elevation angle may not change, but the area covered by each pixel increases, resulting in a degraded position estimate. To resolve the range ambiguity, many optical sensors utilize stereo cameras, which are adequate at short ranges, but for long-range aircraft, their range estimates become unreliable [3].

Many GPS-denied navigation solutions resolve depth using light detecting and ranging (LiDAR) devices [4–6]. These sensors perform accurate range measurements, though long-range LiDARs often have large size, weight, and power (SWaP), rendering them a poor solution for small UAVs. Additionally, LiDARs are limited in that their focused beam often requires gimballing.

In contrast to optical sensors which are hampered by poor weather conditions, such as rain, dust, and fog, radar remains unaffected. Additionally, as radar is an active sensor, it remains unaffected by night-time navigation. Radar, as a range-only sensor, may be viewed as the dual to
optical sensors [7]. It’s resolution is range independent. With a single antenna, the only orientation measurement is the appearance of the feature within the antenna’s beam-pattern. The azimuth and/or elevation angle to features may be estimated when multiple antennas are used, though the accuracy of this measurement is heavily affected by the feature’s geometry. These systems often require the radar to operate coherently and typically require increased computational capability.

Historically, radar’s use has been limited by the radar’s large SWaP, while recent technological advancements [8–10] have resulted in significantly smaller systems. For example, IMSAR’s NanoSAR C, the radar used for testing in our work, the antenna, GPS, radar, and IMU weighs 2.6lb, consumes less than 25W, with a volume of 86in$^3$, while the radar itself is less than 2lb with a volume of 38.5 in$^3$ [11], thus rendering it a potential payload for small UAVs.

This dissertation seeks to develop a radar odometry approach for the navigation of GPS-denied environments. While the use of radar for motion estimation has been previously explored [12–21], the work presented in this paper is the first to present real radar flight results. Specifically, we use a small synthetic aperture radar (SAR) that is capable of operating on small or large unmanned air platforms. No coherence assumptions are made and unknown, ground point scatterers are used for all navigation, though relatively flat terrain is assumed.

A brief description of our contributions is considered in Section 1.1, following which a brief survey of related GPS-denied navigation approaches is presented in Section 1.2.

1.1 Summary of Contributions

This work develops and tests the use of unknown, environmental ground point scatterers to estimate motion onboard a small UAV. The algorithms presented are designed for use on a low SWaP radar system, and as such, assume no phase coherence, use a single antenna SAR, and allow for dense radar return. The main contributions of this dissertation are grouped by chapter as follows:

- A greedy radar odometry algorithm is presented in Chapter 2. Contributions include:
  - The Hough transform is used to identify ground point scatterers and to perform data association,
– The above ground level (AGL) of the aircraft is resolved from the range return from the radar,
– Using pairs of scatterers, the aircraft’s motion is estimated,
– The implemented approach is validated with real flight data, demonstrating decreased drift when GPS is lost.

• Chapter 3 presents a novel Recursive-RANSAC (R-RANSAC) tracking radar odometry algorithm. Contributions include:
  – R-RANSAC is used for point scatterer identification, association, and tracking,
  – The state model and tracking techniques are enhanced to allow for dynamic flight,
  – The implemented enhancements demonstrate reduced drift.

• An enhanced radar odometry algorithm is presented in Chapter 4 and includes the following contributions:
  – Observability analysis for a range-only radar is performed,
  – An observable approach for performing radar odometry is presented,
  – A more robust, R-RANSAC AGL tracking scheme is developed,
  – Techniques for more accurate tracking are discussed,
  – The improvements are validated with real flight data.

1.2 Literature Review

This section contains a literature review of related fields. A variety of GPS-denied navigation techniques are reviewed, followed by a brief discussion of a variety of sensors that have been used for navigation, such as optical sensors, LiDARs, radar beacons, and radar. Many navigation solutions integrate a variety of sensors to create a consistent map, and as such, a class of mapping algorithm is briefly mentioned.
1.2.1 GPS Denied Navigation

With the advancement of GPS, as well as Galileo and GLONASS, UAVs are provided with highly accurate absolute position information. While the availability of such solutions has led to readily available and accurate navigation solutions, reliance on them can be problematic as they are easily jammed. Additionally, indoor navigation or areas with partially obscured views of the sky often result in GPS signal loss.

Prior to the advent of the global positioning system, all navigation was “GPS denied.” These systems seek to estimate motion using a wide variety of sensors. Introceptive sensors, such as inertial measurement units (IMU) and gyros, estimate the acceleration and rotation of the agent relative to itself. While these sensors vary in size, cost, and accuracy, affordable sensors that fit on small UAVs are subject to significant sensor drift. Accordingly, a majority of GPS Denied solutions use exteroceptive sensors to bound or mitigate the drift experienced by relying exclusively on introceptive sensors.

Some exteroceptive sensors, such as magnetometers and pitot tubes, measure environmental conditions. Alternately, other exteroceptive sensors, such as cameras [22–28], laser range-finders [29–32], radio beacons [33–37], sonar [7, 38–42], and radar [13, 15, 19, 43, 44], sense specific features in the surrounding environment. These feature sensors acquire relative measurements, such as range, azimuth angle, or heading angle, which are used to relate the specific feature to the vehicle. Motion is estimated as the measurements change over time. While each of these sensors has strengths, as well as limitations, their commonality provides value when comparing the various dead-reckoning approaches.

1.2.2 Optical Sensors and Visual Odometry

Optical sensors measure the relative orientation of features as observed by one or more cameras. Each feature typically represents a location with distinct change in lighting or an area with little to no change in lighting, and is represented by the \([x, y]\) pixel in the image, which may be converted to azimuth and elevation angle by relating the pixel to the camera’s focal length. Using a second camera, known as stereo vision, the feature range may be estimated, as long as the distance between cameras isn’t significantly smaller than the range to the feature (see [45, 46]). Alternately range may be estimated using a sequence of camera images using a technique...
called bundle adjustment [47]. With either approach, as the range to a feature is increased, the fixed resolution of the camera results in an increasingly erroneous estimate, which is particularly troublesome for unmanned aircraft.

Recovering relative motion between two consecutive images has been researched for more than a century [48] and is known in the vision community as structure from motion (SFM). While much of SFM is focused on reconstructing three dimensional environments from multiple images [49, 50], Visual Odometry (VO) seeks to estimate the motion using changes in perspective from consecutive images, as observed in the imagery [22–24]. Motion is estimated by identifying corresponding features or landmarks in each image, and estimating the motion that would result in the change in landmark orientation. The specific motion estimation technique is greatly influenced by the vehicle’s camera configuration (see [22, 51] for more detail).

VO approaches have been shown to provide successful navigation [52–54] and are cheap and readily available. The main limitation with vision systems is that at the range to a feature increases, each pixel covers an increasingly larger area, fundamentally constraining their ability to track and detect objects [55, 56]. Additionally, they are unable to operate in bad lighting and in bad weather, such as fog or rain.

1.2.3 Laser Range Finders

Laser range finders, or light detection and ranging devices (LiDARs), actively transmit an optical, or infrared, laser and measure the reflection off of the environment to determine range. In doing so, they measure range to a focused point and resultingy are often gimballed so as to collect the range to a variety of locations or features. Range finders are often used in conjunction with VO and are commonly used for GPS-denied automotive applications [30].

LiDAR use for UAVs has been explored [32, 57, 58] and is often referred to as terrain reference navigation (TRN). TRN systems estimate the vehicle’s terrain and compare it to previously acquired, accurate digital elevation model (DEM) data. While accurate, LiDARs are typically range-limited and systems that operate for farther ranges (> 100m) are typically over 10lbs. Unlike cameras, their active nature allows them to operate at night, though like cameras, they operate poorly in poor weather.
1.2.4 Radar Beacons

A variety of navigation systems use radar to communicate with beacons, used as landmarks, that are distributed throughout the navigation environment [21, 59–62]. Each beacon transmits a radio signal, which is received, decoded, and converted to a range measurement, as is done on a global scale with GPS. These sensors are often very accurate, though typically have low update rates.

Much of the research surrounding them builds on a range-only [37, 63, 64] model that is very similar to that used by single-antenna radar systems. In comparison with other radar systems, beacons provide an inherent feature descriptor that is unavailable with other non-beacon radar systems. Further, their dependency on active distributed beacons limits their widespread use as a GPS-denied navigation solution.

1.2.5 Sonar

Sonar is a sensor that uses transmitted sound waves, and the echo from environment features, to detect range. Modern sonar systems typically center on the navigation of unmanned underwater vehicles (UUVs), though their use for airborne sensing has been explored [65].

Many sonar navigation techniques have been implemented. A common approach uses sonar transceivers, which behave similarly to radar beacons [7]. Alternately, a synthetic aperture sonar (SAS) has been used for terrain-based navigation [66]. Another approach has involved performing sonar feature identification and tracking [40].

1.2.6 Radar

Radar is an active sensor that transmits a succession of electromagnetic pulses. As the pulses are reflected by the surrounding environment, many features, called point scatterers, reflect the transmitted pulses straight back to the radar. Using the delay between the transmitted and received pulses, the radar is able to detect the range to these features. Using a single receive aperture, or antenna, radar is only able to measure range.
1.2.6.1 Resolving Bearing to Scatterers

While each antenna alone is unable to measure bearing, by correlating the return and accounting for the relative positioning of the antennas, beam-forming techniques may be used to estimate orientation to environmental point scatterers [67–72]. Using this technique, some have explored the use of radar feature descriptors [70, 72]. It is worth noting that the use of multiple antennas for target tracking (or in our work, scatterer tracking), is a field of research of its own. Space time adaptive processing (STAP) [73, 74] relies on coherence, multiple radar apertures, and sequential chirps to precisely track features in the presence of clutter, noise, or even jamming. As a whole, multiple apertures provide more accurate motion estimates, but also require a significant increase in processing, power, size, and weight.

Other radars, often referred to as 360 degree field-of-view radars, or a radar scanners, operate like a LiDAR, using a gimballed, narrow beamwidth antenna to measure the orientation of point scatterers [59, 61, 62, 75–83]. These systems simpler than using multiple apertures, but require additional, very precise mechanical gimbaling. This technique is often used on ground platforms and is also performed in conjunction with mapping. In contrast, synthetic aperture radar (SAR) systems use a single, side-facing receive antenna and the motion of the aircraft to synthetically mimic multiple aperture systems. These simple systems are often used for image formation and have no gimballing requirements. Further, recent technological enhancements have allowed for the historically large SWaP of SAR systems to drastically decrease [84, 85].

1.2.6.2 Navigation with Radar

The variety of radar configurations results in varying motion estimation techniques, as each radar configuration lends itself to a different navigational approach. Our work focuses on airborne navigational systems, though other work has explored the use of radar for ground-based navigation (see [19, 86–90]).

Many motion estimation techniques, specifically with SAR image formation, seek to refine the motion estimates provided by GPS and an IMU to improve the image resolution [91–94]. These approaches exclusively focus on forming a quality SAR image, rather than estimating the motion itself. Additionally, they often rely on GPS, while also making undesired phase coherence assumptions.
Terrain aided navigation (TAN) systems bypass scatterer position entirely as they use a narrow beamwidth radar as an altimeter in conjunction with a known contour map of the earth to estimate the aircraft’s position [95–97].

Recently, research has started focusing on the use of single aperture SAR systems as a navigational aid. Initial work developed the sparse target method (STM), a target tracking algorithm which relies on sparse return, a reflector’s range return, and phase history to identify ground reflectors and estimate their track in an image [12, 14]. Later, an INS was integrated [18] with an extended Kalman filter (EKF) to estimate two dimensional motion (assuming constant elevation) [15, 16].

To improve the algorithm a M-out-of-N detection scheme was added in conjunction with a global nearest neighbor (GNN) algorithm for feature identification and association [13, 17]. This work culminated in a simulation with a 0.16% drift rate over a 120km simulated flight. Using the same technique, recent tests involved moving a radar in an indoor environment, with stationary, ideal corner reflectors. The radar moved 9 meters on a wheeled platform, resulting in a total error (using a Tactical-grade IMU) of .36 meter or 4% drift [19], significantly less than the 5 meter drift experienced by the IMU alone, though the results are inconsistent with the simulations.

1.2.7 Mapping

Although this dissertation develops an odometry approach, a variety of GPS-denied algorithms improve the consistency of odometric navigational systems by creating a catalog of known or observed features, referred to as a map. These maps maintain feature estimates long after they leave the sensors field of view with the hope that the feature will be re-observed. Once a previously known feature is identified, also referred to as loop closure, mapping algorithms perform an additional estimation step over the entire state model. Though various mapping algorithms exist, simultaneous localization and mapping (SLAM) and view-based maps are specifically discussed.

1.2.7.1 SLAM

The foundation of SLAM involves using observed landmarks to estimate geometric uncertainty, a framework that was developed by both Smith and Cheeseman [98] and Durrant-Whyte
Later work developed an approach to represent the spatial relationships of observed landmarks in a stochastic map [100], a state-space model representing the vehicle pose and landmark positions. The argument for SLAM was strengthened when Csorba [4, 100, 101] showed that by combining localization and mapping into a single estimation problem resulted in a convergent solution. Similarly, Thrun developed a convergent probabilistic approach [65]. With either approach, convergence depends on loop closure.

Much of the original SLAM work was model-based, and thus sensor agnostic [98–100]. Thrun’s probabilistic model assumed a sensor that was able to measure the type of landmark and approximate distance and relative angle from the vehicle and to the landmark [65]. Leonard and Durrant-Whyte tested their EKF localization technique using sonar beacons [33], while Csorba used a laser rangefinder [4].

Multiple approaches to SLAM exist. EKF-SLAM, the most popular implementation, models noise as Gaussian and uses a extended Kalman filter (EKF) to estimate the map and vehicle state. FastSLAM uses a Rao-Blackwellised particle filter, which more readily handles nonlinear process models [102]. GraphSLAM is similar to EKF-SLAM, but composes the state matrix in an information-state form that better handles the large state models present in large maps [103].

While SLAM is successful for navigation, the algorithm is often overly confident in it’s estimate, often trusting inaccurate estimates. Additionally, the state model, with its growing number of features, often becomes prohibitively large. Much of the current SLAM research involves simplifying, or pruning, the feature space. For a thorough treatment of SLAM, see [6] and [5].

As SLAM is sensor agnostic, it is no surprise that many have explored the use of radar with SLAM. This work has typically involved either radar beacons [59, 62], or a 360 degree field of view radar [59, 61, 62, 75–79, 81, 83, 104].

1.2.7.2 View-Based Maps

One noteworthy SLAM approach is view-based mapping [105, 106], though it is specifically developed for use with optical sensors. The view-based maps algorithm builds on visual odometry, while also implementing many of the techniques developed for FrameSLAM [26, 107, 108]. Specifically, it describes visual features using a bag-of-words approach [109] and adds a vocabulary tree [110] for speed improvement. Additionally, it implements a skeleton graph, a
trimmed constraint graph relating selected, consecutive image frames. The use of a skeleton graph, rather than a typical SLAM graph, lessens the number of nodes on the graph, thus allowing for improved performance over long maneuvers.

1.3 Literature Summary

While brief, the literature review provides significant insight into the field of GPS-denied navigation. Specifically, it demonstrates that a wide variety of sensors are used for GPS-denied navigation. In fact, autonomous navigation has been demonstrated in hardware with vision, LiDAR, beacons, and sonar systems, though each of these approaches has significant limitations, particularly for UAVs. Vision and LiDAR sensors are unable to operate in poor weather and are very limited at night, particularly on air platforms. Sonar systems are ineffective at high altitudes, while beacons cannot be used in navigate new, unknown environments.

Radar provides significant robustness to weather and is able to operate day or night, though its use for navigation has been surprisingly limited. Some radar terrain-based navigation systems have demonstrated accurate navigation, though a known map is required, while other SAR-based navigation techniques have shown some ground results, but have never been demonstrated on aircraft. Further, while the simulation of radar provides value for testing, radar is particularly difficult to emulate, particularly when considering environmental point scatterers (as compared to ideal corner reflectors), and as such real results are necessary to demonstrate the validity of any radar-based navigation approach.

This dissertation presents a novel radar odometry approach to GPS-denied navigation. While some previous work has been performed using simulation, the work in this dissertation is the first to demonstrate the value of a single aperture radar for navigating unknown environments. It demonstrates effective altitude estimation using nadir. Further, this work demonstrates that both the Hough transform and Recursive-RANSAC algorithms are capable of identifying features from cluttered, real, range-compressed imagery.
Chapter 2

Radar Odometry on Small Unmanned Aircraft - A Greedy Approach

This chapter develops a novel radar odometry solution that uses stationary, ground point scatterers to reduce the GPS-denied drift rate of an IMU. The effectiveness of the proposed radar odometry algorithm is demonstrated using actual radar data captured during outdoor flight. Relative to the existing literature, the contributions of this chapter are as follows. First, in this chapter the Hough transform is used to identify scatterers and perform data association. Second, the altitude above ground level (AGL) is estimated using the radar return from the ground, as opposed to [15] which assumes a fixed AGL, [18] which estimates motion strictly from INS, or other approaches which use a barometer to estimate elevation independent of the ground level. Third, this chapter explores the radar odometry navigational accuracy of non-straight flight for various banking angles. The results in this chapter demonstrate accurate navigation when using unknown, environmental point scatterers (such as cars and buildings), which often are smeared and unfocused in the range-compressed imagery. Finally, this dissertation is the first to present successful radar odometry results from a real, outdoor flying platform.

Preliminary portions of this dissertation have appeared in [44], but the current work extends [44] in several ways. First, [44] assumes that the aircraft is flying due North. The current work allows for more general flight paths that are not necessarily in a straight line. Second, the filtering, data association, and noise cancellation techniques described in [44] have been significantly modified and revised. Third, this dissertation explores the behavior of the aircraft by simulating a variety of banking angles. Finally, this dissertation contains flight test results as opposed to [44] which presented only simulation results.

The chapter is organized as follows. Section 2.1 describes radar-generated range-compressed imagery, which is the input to the radar odometry algorithm. The radar odometry algorithm is then described in Section 2.2. The output of the radar odometry algorithm is an estimate of the drift rate
of the vehicle and an AGL estimate. Section 2.3 describes the design of an extended Kalman filter that uses these inputs together with an IMU to estimate the position, velocity, Euler angles of the vehicle, and the INS bias. Section 2.4 presents results that demonstrate improved motion estimates during real flight. While the radar odometry algorithm presented in this chapter assumes that over any small window of time (roughly 2-3 seconds), there are no changes in heading, simulation results are also presented in Section 2.4 that quantify the accuracy of the algorithm during various banking maneuvers.

2.1 Radar Range Compression

In this dissertation, a Linear Frequency Modulated Continuous Wave (LFM-CW) Radar is used, which returns the range to scatterers observed in the beamwidth of the radar’s single aperture (antenna) during a single chirp.

2.1.1 Radar Range and LFM-CW Radar

LFM-CW radar involves repeatedly transmitting a chirp signal and receiving the return from scatterers in the environment. The frequency of chirp repetition is referred to as the pulse repetition frequency,

\[ PRF = \frac{1}{\tau}, \]  

(2.1.1)

where \( \tau \) is the chirp duration. The transmitted chirp signal is given by

\[ x_t (t) = a_t (t) \cos (2\pi t F (t) + \Phi_t) \]  

(2.1.2)

where \( \Phi_t \) is a fixed phase offset and

\[ a_t (t) = u (t) - u (t - \tau) \]  

(2.1.3)

represents the transmit pulse amplitude, \( u(t) \) is the unit step function, and where

\[ F (t) = F_0 + \frac{\beta}{2\tau} t \]  

(2.1.4)
indicates the frequency as a function of the initial transmit frequency $F_0$, the transmit bandwidth $\beta$, and the time $t$.

The transmitted signal reflects off various scattering centers in the environment, including the ground, and the time it takes for the signal reflected by scatterer $i$ to return to the transmitter is denoted as

$$\Delta t_i = \frac{r_i}{c},$$

(2.1.5)

where $r_i$ is the range to scatterer $i$ and $c$ is the speed of light. The reflected signal received by the radar is

$$x_r(t) = a_r(t) \sigma_i x_t(t - \Delta t_i),$$

(2.1.6)

where

$$a_r(t) \approx a_t(t)$$

(2.1.7)

is the receive window and $\sigma_i$ is the radar return from the $i^{th}$ scatterer.

The transmit and receive signals are mixed at the transmitter, resulting in

$$x_r(t) \times x_t(t) = a_t(t) \cos(2\pi t F(t)) \sigma_i x_t(t - \Delta t_i)$$

(2.1.8)

$$= a_\sigma(t) \cos(2\pi t F(t)) \cos(2\pi t_{\Delta,i} F(t_{\Delta,i})),$$

(2.1.9)

where

$$t_{\Delta,i} = t - \Delta t_i$$

(2.1.10)

and

$$a_\sigma(t) = a_t(t) a_t(t_{\Delta,i}) \sigma_i.$$  

(2.1.11)

Taking the Fourier Transform of the mixed transmit and receive signals is referred to as range-compressing the chirp. As the scatterer radar cross-section is unknown and range dependent, the
mixed transmit and receive signal is approximated as

\[ x_r(t) \otimes x_t(t) \approx \cos(2\pi F(t)) \cos(2\pi t_{\Delta,i} F(t_{\Delta,i})) , \]  

resulting in the approximate range-compressed signal

\[ X_m(j\omega) = \mathcal{F} \{ x_r \otimes x_t \} \approx \left( \frac{\sin\left(\pi \left(\omega - \frac{\beta}{2\tau} \Delta t_i \right) \tau \right)}{\pi \left(\omega - \frac{\beta}{2\tau} \Delta t_i \right) \tau} \right)^2 \implies \left( \text{sinc}\left(\omega - \frac{\beta}{2\tau} \Delta t_i \right) \right)^2 , \]

where \( \mathcal{F} \{ \cdot \} \) is the Fourier transform and the sinc function is centered at the range-dependent frequency

\[ \omega_i = \frac{\beta}{2\tau} \Delta t_i = \frac{\beta}{\tau} \frac{r_i}{c} . \]

The range-compressed chirp represents the accumulative strength of all reflected returns for a given range during the specified chip. Radar signal processing algorithms typically quantize the range into a finite set of range bins. The mapping from range bin index \( b \) to range is defined as

\[ r = r_0 + b r_{\text{res}} , \]

where \( r_0 \) is the minimum range bin visible to the radar, and \( r_{\text{res}} \) is the radar’s range resolution.

The radar measurement over the time window \([t - \tau, t]\) of the chirp is represented as a column vector where each row index (range bin) represents a particular range, and the value at that index represents the strength of the radar return at that range for that chirp. Since chirps occur sequentially in time, we define the chirp index

\[ s = \left\lfloor \frac{t - t_0}{\tau} \right\rfloor \]

\( \left( \text{2.1.17} \right) \)

\[ \text{to be the index of the chirp that occurs over the time window } [t_0 + (s - 1) \tau, t_0 + s \tau] , \text{ where } t_0 \text{ corresponds to the absolute time the first chirp in the image was started.} \]
Stacking the range measurements at consecutive chirps results in a positive matrix $I_{RC}[s,b]$ called the range-compressed image which can be displayed and visualized as an image. A range-compressed image from an actual flight is shown in Figure 2.1.1.

![Range Compressed Image](image)

**Figure 2.1.1:** Range-compressed image of radar return during a straight and level aircraft flight.

The top row of the image in Figure 2.1.1 is the radar return at the closest range bin $r_0$. Range increases as the row number increases down the image. The bright line about one third of the way down Figure 2.1.1 is the reflection due to the ground directly below the aircraft, where bright pixels correspond to a strong return at that range bin for each chirp. The dark pixels in the top third of Figure 2.1.1 are due to the fact that the radar is onboard an aircraft and that there are no scatterers between the aircraft and the ground.

A more thorough treatment of range compression is found in [84].

### 2.1.2 Range to Scatterer During Flight

Note that in Figure 2.1.1 there are multiple bright curves in the range-compressed image. Each of these hyperbolic curves correspond to a strong point scatterer in the environment, as is explained in this section.
Figure 2.1.2: The geometry that shows how a ground scatterer is displayed in the range-compressed image.

Figure 2.1.2 depicts an aircraft at position $p(t)$ as it flies in a straight line past scatterer $i$ located at $m_i$. The position of the aircraft when it is closest to $m_i$, as identified by the aircraft, is defined as $p_i^*$ and occurs at time $t_i^*$. If the velocity of the aircraft is denoted as $\dot{p}$, then assuming constant velocity, straight flight, the position of the aircraft as a function of $t$ can written as

$$p(t) = p_i^* + \dot{p} (t - t_i^*). \quad (2.1.18)$$

The range to scatterer $i$ is therefore given by

$$r_i(t) = \|p(t) - m_i\| = \sqrt{(p(t) - m_i)^T (p(t) - m_i)} \quad (2.1.19)$$

$$= \sqrt{\|p_i^* - m_i\|^2 + \|\dot{p}\|^2 (t - t_i^*)^2}, \quad (2.1.20)$$

where we have used the fact that for straight flight and constant velocity, $\dot{p}$ is orthogonal to $(p_i^* - m_i)$. Defining the ground speed as $V_g \overset{\triangle}{=} \|\dot{p}\|$ and the minimum range to the scatterer as
Equation (2.1.20) can be rearranged to get the hyperbolic equation

\[
\frac{r_i^2(t)}{r_i^*} \frac{V_g^2 (t - t_i^*)^2}{r_i^*} = 1,
\]  

indicating that under ideal, straight flight conditions, the range to each scatterer appears as a hyperbola in the range-compressed image. Note from Figure 2.1.1 the presence of many hyperbolic looking curves, each indicating the presence of a strong scatterer. The key idea in this section is to use these hyperbolic curves to estimate the motion of the aircraft.

Since the range-compressed image is parametrized by discrete range indices \( b \) and chirp indices \( s \), Equation (2.1.21) is expressed using these indices. From Equation (2.1.17),

\[
t - t_i^* = \tau (s - s_i^*),
\]

where \( s_i^* \) is the chirp index at the time of closest approach to the \( i^{th} \) scatterer. From Equation (2.1.16), the range at chirp \( s \) is

\[
r_i[s] = r_0 + b_i[s] r_{res}.
\]

Letting \( r_i^* \) be the range to the \( i^{th} \) scatterer at the point of closest approach, and \( b_i^* \) the associated range bin, gives

\[
r_i^* = r_0 + b_i^* r_{res}.
\]

Therefore converting Equation (2.1.21) to discrete range bin and chirp parameters gives

\[
\frac{(r_0 + b_i[s] r_{res})^2}{(r_0 + b_i^* r_{res})^2} - \frac{V_g^2 (s - s_i^*)^2 \tau^2}{(r_0 + b_i^* r_{res})^2} = 1.
\]

### 2.2 Radar Odometry

A block diagram of the radar odometry algorithm is shown in Figure 2.2.1. The input to the algorithm is the range-compressed image from the radar. The range-compressed image can be used to estimate the altitude above ground level (AGL) of the radar. This process is described in
Section 2.2.1. As shown in Figure 2.2.1, the first element of the radar odometry algorithm is to pre-filter the range-compressed image. The pre-filtering algorithm will be discussed in Section 2.2.2. The filtered image is then processed using a Hough transform to identify and characterize the scatterers with large radar cross section. This process is identified as scatterer identification in Figure 2.2.1 and is described in detail in Section 2.2.3. The tracks of the scatterers are passed to the range estimation block in Figure 2.2.1 for an initial estimate of the range based on the resulting hyperbolic shape in the range-compressed image. Details of this process are described in Section 2.2.4.

The AGL estimate and the range estimate are used to estimate the relative drift of the aircraft in the along-track and cross-track directions. This process is described in Section 2.2.5 and results in a relative drift estimate that is the output of the radar odometry algorithm. The drift motion estimate and AGL estimate of the aircraft provided by radar odometry are used as measurement inputs to an extended Kalman filter (EKF), that fuses the radar odometry measurement with IMU measurements to produce an estimate of the state of the aircraft relative to its initial position. The EKF is described in Section 2.3.
2.2.1 AGL Estimation

As shown in Figure 2.2.1, the range-compressed image can be used to estimate the altitude above ground level (AGL). The radar return from the ground located immediately below the UAV is called nadir, and is the signal used to estimate AGL. Figure 2.2.2 shows one column of the range-compressed image on a dB scale, where it can be seen that the noise floor dramatically increases at the range that coincides with ground level.

This is further visualized by Figure 2.1.1, where nadir is observed as the bright line in the middle of the image, that gradually decreases in range, moving up the image, from left to right. At each chirp the AGL is calculated as the first range bin, with a signal larger than the AGL threshold $T_{AGL}$, and is identified as $d_{AGL}$.

2.2.2 Range-Compressed Image Pre-Filter

Any range-compressed image generated by a LFM-CW synthetic aperture radar contains multiple noise sources. Multi-path radar returns and system noise result in a significant amount of noise speckle. Figure 2.2.3 shows a portion of Figure 2.1.1 that has been magnified to show additional detail. The image pre-filter algorithm attempts to remove much of the noise in the range-compressed image by using the following steps.

**Figure 2.2.2:** One column of the range-compressed image, showing the dramatic increase in signal and noise floor beyond the range associated with nadir. The peaks prior to nadir are noise in the radar itself.
Figure 2.2.3: Magnified segment of the range-compressed image of radar return during an aircraft flight.

Step 1. The bias in the image is removed by subtracting from each pixel, the weighted average of its neighbors.

Step 2. As the ranges to each scatterer change very slowly in comparison to the chirp index, a weighted horizontal corner kernel,

\[ k_{hc} = \begin{bmatrix} -1 & 2 & -1 \end{bmatrix}, \]

(2.2.1)

is used to detect features from noise by convolving the kernel with the range-compressed image [46, 111].

Step 3. Range dependent noise sources are removed from the image by calculating and removing the weighted average value of each range bin.

Step 4. In a range-compressed image, the brightest return is almost always at, or immediately after, nadir, with decreasing return as the range past nadir increases. If nadir changes over time, compensating for the average range return improperly weights range bins which are near the changing nadir. Accordingly, define a nadir-adjusted range bin

\[ b_{AGL}[s] = b - d_{AGL}[s]. \]

(2.2.2)
This adjusted range is used to perform a weighted, AGL-adjusted, nadir-adjusted mean removal.

**Step 5.** To identify clutter from scatterer return, the resulting image is then thresholded [46, 111, 112].

The result of these steps is the filtered image denoted as $I_F$. Applying the pre-filter step to the section of the range-compressed image shown in Figure 2.2.3 results in Figure 2.2.4.

### 2.2.3 Scatterer Identification

While pre-filtering the range-compressed image removes much of the noise, it is still necessary to identify individual scatterers. Many data association techniques, such as least-squares, or the random sample consensus algorithm (RANSAC) [113] are only able to identify single models from a data set. Other approaches, such as a M-out-of-N detector, are limited in dense data sets or in the presence of clutter. The Hough transform is a voting algorithm which is able to identify a large number of parameter models from within a single data set [114].

Based on the discussion in Section 2.1.2 the Hough transform identifies scatterers by searching for hyperbolas in the range-compressed image. With reference to Equation (2.1.25), the range resolution $r_{res}$ and the minimum range $r_0$ are constant. Therefore, the hyperbola associated with each scatterer is uniquely identified by the parameters $V_g^*$, $b_i^*$, and $s_i^*$. A hyperbolic Hough transform is used to identify and provide initial parametrization for each scatterer.
Toward that end, define the possible parameter set

$$\mathcal{H} = \mathbb{V} \times \mathbb{B} \times \mathbb{S}, \quad (2.2.3)$$

where $\mathbb{V} \subset \mathbb{R}^+$ is a finite range of possible ground speeds, $\mathbb{B} \subset \mathbb{N}^+$ is a finite set of possible range bins, and $\mathbb{S} \subset \mathbb{N}^+$ is a finite range of possible chirp indices. The vector

$$H = (V_g, b_i^*, s_i^*) \in \mathcal{H}, \quad (2.2.4)$$

are the parameters associated with a unique hyperbola. The Hough transform scans the thresholded range-compressed image and, when a pixel is illuminated, assigns a vote to all parameters in $\mathcal{H}$ that result in a hyperbola that passes through that pixel.

Figure 2.2.5 shows a two dimensional slice of the Hough transform of Figure 2.2.4 where $\hat{V}_g$ has been held constant at the estimated ground velocity of the aircraft. The intensity of each pixel in Figure 2.2.5 indicates the number votes for that parameter. Figure 2.2.4 shows three distinct hyperbolas corresponding to strong scatterers on the ground. The corresponding hyperbolic parameters are the three dark pixels in Figure 2.2.5.

The clutter in the Hough transform is due to radar smearing and noise resulting in hyperbolas in the range-compressed image that are not one pixel wide, non-straight flight, non-constant speed
airspeed, measurement inaccuracies, and multiple scatterers in the environment. To isolate the peaks in the Hough transform, and thereby identify the scatterers, the following steps are taken.

Step 1. The parameter $H_{\text{max}} \in H$ with the largest number of votes is identified. The ground speed $\hat{V}_g$ of the sensor platform is estimated to be the ground speed index containing $H_{\text{max}}$. The remaining parameters ($b_i^*$ and $s_i^*$) are estimated using the resulting two dimensional slice of $H$ which is denoted as $I_{\hat{V}_g}$.

Step 2. The average pixel value is removed from the Hough transform resulting in the mean-removed image

$$I_{mr} = I_{\hat{V}_g} - \bar{I}_{\hat{V}_g},$$

(2.2.5)

where $\bar{I}_{\hat{V}_g}$ is the average pixel value in the image $I_{\hat{V}_g}$.

Step 3. The mean-removed image $I_{mr}$ is normalized to produce the normalized image

$$I_n = \frac{I_{mr}}{\max_{b,s}(I_{mr}[b,s])},$$

(2.2.6)

Step 4. The normalized image is thresholded to produce

$$I_m[b,s] = \begin{cases} 
1 & I_n[b,s] > T_n \\
0 & I_n[b,s] \leq T_n,
\end{cases}$$

(2.2.7)

where $T_n$ is the minimum number of votes allowed for a the identification of a feature. The thresholded image corresponding to Figure 2.2.5 is shown in Figure 2.2.6.

Step 5. The pixels illuminated in the thresholded image are then segmented into connected groups $G_{\hat{V}_g}$. For example, in Figure 2.2.6 there are three connected groups.

Step 6. Each connected group corresponds to a single parameter $H_i \in H$, where $i \in G_{\hat{V}_g}$. To find $H_i$ for each connected group, $I_m$ is convolved with a $5 \times 5$ smoothing kernel. After smoothing, the parameters of the pixel with the maximum value for each subgroup is assigned as $H_i$. To reduce the likelihood of mis-identifying clutter as a scatterer, only groups containing more than $T_g$ pixels are considered.
Figure 2.2.6: Thresholded Hough-space image $I_m$.

Figure 2.2.7: Pre-filtered range-compressed image (shown in gray) with super-imposed hyperbola estimates shown in black.

Figure 2.2.7 shows the hyperbolas estimated using the six steps described above, superimposed on the range-compressed image.

2.2.4 Range Estimation

As can be seen from Figure 2.2.7, the hyperbolas produced by the algorithm described in the previous chapter are overly optimistic in two ways. First, if the flight path of the sensor is not actually a straight line, then the range to the scatterer is not be an exact hyperbola. Second, the hyperbolas shown in Figure 2.2.7 extend to areas of the range-compressed image that do not have
any supporting measurements. This happens, for example, when the scatterer leaves the antenna beamwidth of the radar. To correct for these issues, the final range estimate for each scatterer at each chirp bin is obtained by traversing the hyperbolas produced by the algorithm described in the previous chapter and making corrections based on the supporting pixels in the pre-filtered range-compressed image.

To be precise, let $h_i^-(s)$ denote the hyperbola associated with the $i^{th}$ scatterer, where $s$ is the chirp number. Note that $h_i^-(s)$ can be thought of as a one pixel width line running through the pre-filtered range-compressed image, where $h_i^-(s)$ denotes the estimated range to the $i^{th}$ scatterer at chirp $s$. At each chirp index, the range to the $i^{th}$ scatterer $h_i(s)$ is determined as the center of the illuminated range pixels that are contiguous with $h_i^-(s)$. If no range bins contiguous to $h_i^-(s)$ are illuminated and this persists over several chirp indices, then the $i^{th}$ scatterer is assumed to be out of the field of view, and $h_i(s)$ is removed for all subsequent chirps. The result of following this process is shown in Figure 2.2.8. The result of this process is identified in this paper as the measured range to each scatterer.

2.2.5 Relative Drift Estimation

In this chapter we assume that the lateral position of the aircraft may drift due to wind and other environmental factors, but that its heading direction remains constant over a small window of time. Specifically, a constant heading is assumed at the time at which the aircraft passes the
minimum range of selected pairs of scatterers, a window that is typically less than 2 to 3 seconds.

The objective of this section is to show how the relative drift of the aircraft can be determined from a window of the range-compressed image. Recall from Figure 2.1.2 that the vertex of the hyperbola corresponds to that point in time when the scatterer is aligned with the point of closest approach, or in other words, at the point in time when the flight path of the aircraft is perpendicular to the line of sight to the scatterer. Let the position of the \(i^{th}\) scatterer be \(m_i\) and let \(p(t)\) be the position of the aircraft at time \(t\). Define the range to the \(i^{th}\) scatterer at time \(t\) to be

\[ r_i(t) = \|p(t) - m_i\|. \tag{2.2.8} \]

Then the time of closest approach is given by

\[ t^*_i = \arg\min_t r_i(t). \tag{2.2.9} \]

It is assumed in this section that the estimate of the motion is required at a sample rate \(T_s\) that is significantly lower than the chirp rate \(T_c\) of the radar. In that case there are \(T_s/T_c\) columns of the range-compressed image between each sample time. We assume that the range-compressed image between sample times contains the vertex of the hyperbolas corresponding to at least two scatterers. If that is not the case, then additional past columns need to be appended until there are two vertexes and the range to the other scatterer is available at those vertices. Let \(m_i\) and \(m_j\) correspond to the positions of the two scatterers. Then the range-compressed image can be used to compute \(r_i(t^*_i), r_j(t^*_i), r_i(t^*_j),\) and \(r_j(t^*_j)\).

The geometry associated with calculating the relative drift is shown in Figure 2.2.9.

The objective is to calculate the relative drift \((\Delta x, \Delta y)\) of the aircraft in the along-track \((x)\) and cross-track \((y)\) directions. From the geometry of Figure 2.2.9,

\[ \Delta x^2 + (r_j(t^*_j) - \Delta y)^2 = r_i(t^*_i) \tag{2.2.10} \]

\[ \Delta x^2 + (r_i(t^*_i) + \Delta y)^2 = r_j(t^*_j). \tag{2.2.11} \]
Solving for the cross track drift $\Delta y$ gives

$$\Delta y = \frac{r_j^2(t_j^*) - r_i^2(t_i^*) + r_i^2(t_j^*) - r_j^2(t_i^*)}{2 (r_i(t_i^*) + r_j(t_j^*))}. \quad (2.2.12)$$

Given $\Delta y$, the along track drift $\Delta x$ from Equation (2.2.10) is calculated as

$$\Delta x = \sqrt{r_j^2(t_j^*) - \left(r_j(t_j^*) - \Delta y\right)^2}. \quad (2.2.13)$$

If the vertex of more than two scatterers is visible in the time window of the range-compressed image, then every pair of scatterers is used to estimate the relative motion over the past time sample $T_s$ and the results are averaged.

Selecting scatterer pairs located at $\mathbf{m}_i$ and $\mathbf{m}_j$ such that the scatterers are both visible at $t_i^*$ and $t_j^*$ involves sorting the scatterers by their respective $t^*$. Sequential scatterers are then selected and the resulting velocity is estimated as

$$\begin{pmatrix} \dot{v}_x \\ \dot{v}_y \end{pmatrix} = \frac{1}{|t_j^* - t_i^*|} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}. \quad (2.2.14)$$
In the event that the resultant speed estimate is not on the same scale as $\hat{V}_g$, inconsistent values are discarded. When $\hat{V}_g$ is on the same scale then the final velocity estimate is given by

$$\begin{bmatrix}
\nu^d_x \\
\nu^d_y
\end{bmatrix} = \frac{\hat{V}_g}{\sqrt{(\nu^d_x)^2 + (\nu^d_y)^2}} \begin{bmatrix}
\nu^d_x \\
\nu^d_y
\end{bmatrix}.$$  \hspace{1cm} (2.2.15)

### 2.3 Extended Kalman Filter

This section describes the extended Kalman filter (EKF) shown in Figure 2.2.1. For navigation, many GPS-denied solutions use the indirect Kalman filter (or error-state Kalman filter) [16, 115–117]. This approach optimizes the processing for a system containing a high measurement rate sensor (like INS), with low frequency noise, when used in conjunction with low measurement rate sensors (such as GPS or vision), with high frequency noise. Our situation is different in that the radar is also a high rate sensor. Therefore, the computation benefits that are often associated with indirect Kalman filters are not achieved for our scenario. Therefore, since the performance and computational requirements for the direct and indirect Kalman filters are similar, we present here the direct filter.

The input to the filter are the IMU, a digital compass, the AGL estimate $d_{AGL}$ described in Section 2.2.1, and the drift velocity given by Equation (2.2.15). The output of the filter is the estimate of the state including the inertial position and velocity, and the Euler angles of the vehicle.

We assume that the system model is given by

$$\dot{x} = f(x, u + \mu) + \xi$$  \hspace{1cm} (2.3.1)

$$y_k = h(x_k) + v_k,$$  \hspace{1cm} (2.3.2)

where $v_k$ is a zero mean random sequence with covariance $R$, and where $y_k$ denotes the measurement at discrete sample time $k$. The process noise is modeled as two separate components. The IMU noise is modeled by $\mu$, a zero mean Gaussian process with sensor-specific covariance $M$, while $\xi$ models the remaining system noise and is represented as a zero mean Gaussian process with covariance $Q$. Since the covariance $Q$ represents model uncertainty and is unknown, $Q$ is used
as a tuning parameter. In this section we use the continuous state, discrete measurement, extended Kalman filter described in e.g. [118]

**Prediction Step (between measurements):**

\[
\dot{\hat{x}} = f(\hat{x}, u)
\]

\[
\dot{P} = \frac{\partial f}{\partial x} P + P \frac{\partial f^T}{\partial x} + Q + \frac{\partial f}{\partial u} M \frac{\partial f^T}{\partial u},
\]

**Measurement Update (at measurements):**

\[
L = P^{-1} \frac{\partial h}{\partial x} \left( R + \frac{\partial h}{\partial x} P^{-1} \frac{\partial h}{\partial x}^T \right)^{-1}
\]

\[
P = \left( I - L \frac{\partial h}{\partial x} \right) P^{-}
\]

\[
\hat{x} = \hat{x}^- = L (y_k - h(\hat{x}^-)),
\]

where \(\hat{x}^-\) and \(P^-\) denote \(\hat{x}\) and \(P\) at the end of the prediction step, just prior to the measurement update, and \(\hat{x}^+\) and \(P^+\) denote \(\hat{x}\) and \(P\) after the measurement update.

### 2.3.1 Prediction Model

If the position expressed in the inertial frame is given by

\[
p^i = [n \ e \ d]^T,
\]

where \(n\), \(e\), and \(d\), indicate the north, east, and down position of the aircraft, the velocity is given by

\[
v^i = [\dot{n} \ \dot{e} \ \dot{d}]^T,
\]

and the 3-2-1 Euler angles are given by

\[
\Theta = [\phi \ \theta \ \psi]^T,
\]
where $\phi$ is the roll angle, $\theta$ is the pitch angle and $\psi$ is the heading angle. Additionally, the gyroscope bias states is represented are

$$ \delta b = \begin{bmatrix} b_p & b_q & b_r \end{bmatrix}^T, $$

(2.3.11)

where $b_p$, $b_q$, and $b_r$ are the body angular drift rates. The resulting system state is defined by

$$ x = \begin{bmatrix} p^i \\ v^i \\ \Theta \\ \delta b \end{bmatrix}. $$

(2.3.12)

The output of the IMU is given by

$$ u = \begin{bmatrix} a^b \\ \omega \end{bmatrix}, $$

(2.3.13)

where $a^b = \begin{bmatrix} a_x & a_y & a_z \end{bmatrix}^T$ are the accelerometer measurements that express the specific acceleration in the vehicle body frame, and $\omega = \begin{bmatrix} p & q & r \end{bmatrix}^T$ are the body angular rates as measured by rate gyros. The system dynamics can then be expressed as

$$ \dot{x} = f(x, u), $$

(2.3.14)

where

$$ f(x, u) \triangleq \begin{bmatrix} \dot{v}^i \\ g^i + R^b_i(\Theta) a^b \\ S(\Theta)(\omega - \delta b) \\ 0_{3 \times 1} \end{bmatrix}, $$

(2.3.15)

and where

$$ S(\Theta) \triangleq \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{bmatrix}, $$

(2.3.16)
\[ \mathbf{g}^i = \begin{bmatrix} 0 & 0 & g \end{bmatrix}^\top \] is the gravity vector in the inertial frame, and \( \mathbf{R}_b^i(\Theta) \) is the rotation matrix that transforms body axes to inertial axes.

The Jacobian of \( f(\mathbf{x}, \mathbf{u}) \) is given by

\[
\frac{\partial f}{\partial \mathbf{x}}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix}
0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\
o_{3 \times 3} & 0_{3 \times 3} & F_1 & 0_{3 \times 3} \\
o_{3 \times 3} & 0_{3 \times 3} & F_2 & S(\Theta) \\
o_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3}
\end{bmatrix}, \tag{2.3.17}
\]

where

\[
F_1 = a_x \begin{bmatrix}
0 & -s_\phi c_\psi & -c_\phi s_\psi \\
o & -s_\phi s_\psi & c_\phi c_\psi \\
o & -c_\theta & 0
\end{bmatrix} + \\
F_2 = \begin{bmatrix}
q c_\phi t_\theta - r s_\phi t_\theta & q s_\phi + r \frac{c_\phi}{c_\theta} & 0 \\
-q s_\phi - r c_\phi & 0 & 0 \\
q \frac{c_\phi}{c_\theta} - r \frac{s_\phi}{c_\theta} & q \frac{s_\phi s_\theta}{c_\theta} + r \frac{c_\phi s_\theta}{c_\theta} & 0
\end{bmatrix}, \tag{2.3.18}
\]

\[
c_\alpha \equiv \cos(\alpha), s_\alpha \equiv \sin(\alpha), t_\alpha \equiv \tan(\alpha), \text{ and}
\]

\[
\frac{\partial f}{\partial \mathbf{u}}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix}
0_{3 \times 3} & 0_{3 \times 3} \\
R_i^b(\Theta) & 0_{3 \times 3} \\
0_{3 \times 3} & S(\Theta) \\
0_{3 \times 3} & 0_{3 \times 3}
\end{bmatrix}. \tag{2.3.20}
\]
2.3.2 Measurement Model

Since the IMU is used in the prediction step, the remaining measurements are (1) the AGL distance $d_{AGL}$, (2) the heading angle $\psi$, and (3) the drift velocity given by Equation (2.2.15). Since the drift velocity is measured in the body frame of the vehicle, and since the radar measurements are independent of the roll and pitch angles of the vehicle (ignoring field of view constraints), the relationship between the drift velocity and the velocity is given by

$$
\begin{bmatrix}
v^d_x \\
v^d_y \\
\end{bmatrix} =
\begin{bmatrix}
\cos \psi & \sin \psi \\
-\sin \psi & \cos \psi \\
\end{bmatrix}
\begin{bmatrix}
v^i_x \\
v^i_y \\
\end{bmatrix}.
$$

(2.3.21)

The pitch angle is observable through the AGL measurement. To make the roll angle observable, the standard coordinated turn condition

$$
\dot{\psi} = \frac{g}{V_g} \tan \phi.
$$

(2.3.22)

is imposed. From Equation (2.3.15),

$$
\dot{\psi} = \frac{\sin \phi}{\cos \theta} q + \frac{\cos \phi}{\cos \theta} r.
$$

(2.3.23)

Therefore a pseudo-measurement for the coordinated turn condition, which should nominally be zero is

$$
y_{\text{turn}} = \frac{\sin \phi}{\cos \theta} q + \frac{\cos \phi}{\cos \theta} r - \frac{g}{\sqrt{(v^i_x)^2 + (v^i_y)^2}} \tan \phi.
$$

(2.3.24)

Defining the measurement vector to be

$$
\mathbf{y} = 
\begin{bmatrix}
y_d \\
y_{\psi} \\
y^d_x \\
y^d_y \\
y_{\text{turn}}
\end{bmatrix}^\top
$$

(2.3.25)

the measurement model is given by

$$
\mathbf{y}_k = h(\mathbf{x}_k) + \mathbf{\eta}_k,
$$

(2.3.26)
where

\[
h(x) = \begin{bmatrix} d \\ \psi \\ v_x^i \cos \psi + v_y^i \sin \psi \\ -v_x^i \sin \psi + v_y^i \cos \psi \\ \sin \phi/cos \theta q + \cos \phi/cos \theta r - g/\sqrt{(v_x^i)^2+(v_y^i)^2} \tan \phi \end{bmatrix},
\]

(2.3.27)

where we have assumed a flat earth model, i.e., \(d_{AGL} = d\). The Jacobian of \(h\) is given by

\[
\frac{\partial h}{\partial x} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{\psi} & s_{\psi} & 0 & 0 & 0 & H_5(x) & 0 & 0 \\ 0 & 0 & 0 & -s_{\psi} & c_{\psi} & 0 & 0 & 0 & H_6(x) & 0 & 0 \\ 0 & 0 & 0 & H_1(x) & H_2(x) & 0 & H_3(x) & H_4(x) & 0 & 0 & 0 \end{bmatrix},
\]

(2.3.28)

where

\[
H_1(x) = \frac{gv_x^i \tan \phi}{\left((v_x^i)^2 + (v_y^i)^2\right)^{3/2}}
\]

(2.3.29)

\[
H_2(x) = \frac{gv_y^i \tan \phi}{\left((v_x^i)^2 + (v_y^i)^2\right)^{3/2}}
\]

(2.3.30)

\[
H_3(x) = \frac{\cos \phi}{\cos \theta} q - \frac{\sin \phi}{\cos \theta} r - \frac{g}{\sqrt{(v_x^i)^2+(v_y^i)^2}} \sec^2 \phi
\]

(2.3.31)

\[
H_4(x) = \left(\frac{\sin \phi}{\cos \theta} q + \frac{\cos \phi}{\cos \theta} r\right) \tan \theta
\]

(2.3.32)

\[
H_5(x) = -v_x^i \sin \psi + v_y^i \cos \psi
\]

(2.3.33)

\[
H_6(x) = -v_x^i \cos \psi - v_y^i \sin \psi.
\]

(2.3.34)

### 2.4 Results

While GPS and IMUs are often coupled to provide accurate navigation, when GPS is lost, the IMU drift remains unbounded. The algorithm presented in this chapter seeks to reduce the
Figure 2.4.1: Average position error when the commanded to fly three flight paths: straight, 10 degree change in heading, and 40 degree change in heading. The average IMU and RO+IMU drift error is shown for all changes in heading.

IMU drift using range measurements to environmental point scatterers. Our results demonstrate the improved accuracy experienced when using radar odometry as opposed to an IMU alone.

2.4.1 Simulation Results

The simulation results were obtained by using the six-degree-of-freedom UAV model and autopilot described in [119]. Approximately 175 scattering centers were randomly placed on flat terrain and the range to the scatterers, corrupted by noise, were used to produce a simulated range-compressed image. The range-compressed image was then processed using the algorithms described in this chapter. The IMU and digital compass were simulated using equations and noise values described in [119] with the drift of all simulated sensors assumed to be zero at the beginning of the flight. The simulations include wind gusts and sensor noise. The initial conditions for each simulation started at the zero position with a fixed forward speed. For comparison purposes, the radar odometry plus IMU (RO+IMU) state estimation scheme described in this chapter, is compared to simply propagating the equations of motion using the IMU, without a radar odometry measurement update state (IMU-only).

The RO+IMU algorithm developed in this chapter assumes that between consecutive point scatterers there are no changes in heading. To test the performance of the algorithm when the
<table>
<thead>
<tr>
<th>Commanded Change in Heading</th>
<th>IMU Error</th>
<th>RO+IMU Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Standard deviation</td>
</tr>
<tr>
<td>0 degrees</td>
<td>1262</td>
<td>390</td>
</tr>
<tr>
<td>5 degrees</td>
<td>1438</td>
<td>445</td>
</tr>
<tr>
<td>10 degrees</td>
<td>828</td>
<td>373</td>
</tr>
<tr>
<td>15 degrees</td>
<td>1387</td>
<td>483</td>
</tr>
<tr>
<td>20 degrees</td>
<td>1375</td>
<td>549</td>
</tr>
<tr>
<td>25 degrees</td>
<td>1418</td>
<td>827</td>
</tr>
<tr>
<td>30 degrees</td>
<td>1439</td>
<td>754</td>
</tr>
<tr>
<td>35 degrees</td>
<td>1848</td>
<td>636</td>
</tr>
<tr>
<td>40 degrees</td>
<td>1906</td>
<td>648</td>
</tr>
</tbody>
</table>

relatively-constant heading flight assumption is violated, the simulation is repeated for the case when the commanded flight path changed by 10 degrees and 40 degrees respectively, over the 60 second simulation window. Figure 2.4.1 shows the results for the 10 degree heading change as indicated by the dashed gray line, while lines with the dash and dot show the results for the 40 degree heading change. Again, the RO+IMU estimates are significantly better than the IMU-only estimates, although the drift rates of both algorithms tend to degrade as the heading change increases.

To further assess the robustness of the algorithm, Monte-Carlo simulations were run where the position of the scatterers was randomly selected for each simulation. Nine additional heading change commands were each executed 144 times. The average position error as a function of heading change command is shown in Table 2.1, for both IMU-only and RO+IMU. The average and standard deviation position errors are shown for each heading change. The table shows that while the IMU-only average error remains relatively unaffected over changes in heading, RO+IMU average errors degrade as the heading change command increases. The table also shows that while the average error increases with changes in heading, the standard deviation of the RO solution remains relatively unaffected. Additionally, it is clear, that RO+IMU performs significantly better than IMU-only.
2.4.2 Flight Test Results

Flight tests were performed using a SAR radar mounted to a Cessna aircraft. Figure 2.4.2 shows a post processed image of the environment where white indicates a strong scatterer. The aircraft flew approximately 2.4 km over the 60 second time window. The terrain is mostly flat, rising approximately 50 m in the along-track direction. It is important to note that while it is assumed that the scatterers are on ground level, their precise elevation is dependent on where they lie in reference to the ground (i.e. on a building, in a tree, or on the ground).

The flight hardware that is used was designed for a different purpose and included a navigation-grade IMU with a gyro with 1-sigma standard deviation of 1 deg/hr and accelerometer standard deviation of 0.0003 g, significantly more precise, larger, and more costly than those used with typical small UAVs. To emulate a consumer-grade lower quality IMU that would typically be found on a small UAV, white noise was added to the IMU measurements resulting in a 1-sigma standard deviation of 0.013 deg/s for the gyros and 1-sigma standard deviation of 0.0025 g for the accelerometers. The RO+IMU and IMU-only schemes for the navigation-grade IMU and the consumer-grade IMU are implemented. Results for the navigation grade sensor are shown in Figure 2.4.3, and results for the commercial grade sensor are shown in Figure 2.4.4. As shown in Figure 2.4.3, with the navigation-grade sensor, RO+IMU outperformed the IMU with a final position error of approximately 135 m as compared to 188 m for the IMU alone.

Figure 2.4.4 shows the average position error using 12 simulations of an emulated commercial-grade IMU. In this case, RO+IMU performs significantly better than IMU-only. It is noteworthy
Figure 2.4.3: Flight results using a navigation-grade IMU.

Figure 2.4.4: Flight results using an emulated commercial-grade IMU, note the scale change as compared to Figure 2.4.3.
that the RO+IMU estimate using the low-quality commercial-grade IMU performs roughly equivalent to IMU-only using the navigation-grade IMU.

2.5 Conclusion

When the GPS signal is lost or denied, current small unmanned air systems are unable to accurately estimate their position. We have demonstrated that using the Hough transform, in conjunction with the radar odometry algorithm derived in this chapter, reduces the average drift error to less than 6%, as compared to the 43% drift when using a commercial-grade IMU to propagate the states. Average real flight results with an emulated commercial grade IMU reduces drift to approximately 6%, while drift rates using the navigation-grade IMU alone was approximately 8%. Therefore the RO+IMU solution gives roughly navigation-quality estimates without the need for an expensive and large navigation-grade IMU. Our results also indicate that using radar odometry for motion estimation is a feasible alternative to traditional visual odometry systems, while allowing for all-weather, any-time-of-day navigation.
Chapter 3

Radar Odometry with Recursive-RANSAC

The approach from Chapter 2 greedily identifies pairs of scatterers, which are used to triangulate both the point scatterers’ positions and the aircraft’s motion. While the approach decreases the GPS-denied navigation error, the piece-wise heading assumption limits the aircraft’s ability to navigate in dynamic environments. Additionally, by greedily selecting only pairs of scatterers, the measurements from all other scatterers are unnecessarily discarded, even further reducing the navigational accuracy.

In [120] we compare the feature identification of the Hough transform with the newly developed recursive random sample consensus (R-RANSAC) algorithm [121, 122]. R-RANSAC relies on the traditional random sample consensus (RANSAC) algorithm, which is a batch algorithm for estimating the parameters of a single signal in clutter and has found wide use within the computer vision community due to its simplicity and robustness. The premise of the standard RANSAC algorithm is to randomly select minimum subsets from a batch of data to form numerous hypotheses of an underlying signal. The hypothesis with the most supporting measurements, or consensus set, is identified as the most accurate hypothesis [113].

The R-RANSAC algorithm extends the standard RANSAC algorithm in two notable ways. First, it stores multiple hypotheses between time steps and recursively estimates their parameters using sequential measurements, e.g., using the recursive least-squares (RLS) filter; measurements whose residual is within a threshold to the stored hypotheses are identified as inliers and used to update existing hypotheses, while outliers are used to seed a new RANSAC hypothesis. The second extension is a natural consequence of storing multiple hypotheses between time steps, in that R-RANSAC is able to estimate the parameters of multiple signals simultaneously; hypotheses that have a consensus set above a threshold are identified as valid hypotheses. In [120], we found that using R-RANSAC drastically decreases the computational complexity compared to the
Hough transform used in [1, 44]. Additionally, the R-RANSAC algorithm is robust in cluttered environments [122], which is a known shortcoming of the GNN approach [123].

This chapter presents a novel, robust, scatterer-tracking and navigation algorithm, while demonstrating its results with real flight radar data. The approach presented in this chapter is novel in several ways. First, it uses the newly developed recursive-RANSAC (R-RANSAC) algorithm to perform data association, as is explored in [120]. Second, this chapter removes the piece-wise constant-heading assumption in [1], thus providing significant robustness to dynamic flight tracks. Third, rather than tracking scatterers using a Cartesian coordinate system, as in [20], this chapter uses a two-dimensional, polar coordinate system, which is more appropriate given the azimuth uncertainty [124]. Further, this chapter assumes the radar is not coherent, and as such the radar only measures the range to ground point-scatterers. Finally, the results in this section shows a reduced drift rate using a navigation grade IMU and real flight data.

The chapter is organized as follows. Section 3.1 describes the use of R-RANSAC for scatterer identification and tracking, while Section 3.2 describes the system’s state space model and the associated extended Kalman filter. Section 3.3 presents the simulated and real radar results.

### 3.1 Radar Odometry with R-RANSAC

The proposed tracking radar odometry tracking algorithm is outlined in Figure 3.1.1. A range-compressed image is the input to the algorithm. The image is initially pre-filtered, as described in Section 2.2.2, and used to generate an above-ground level (AGL) estimate. R-RANSAC identifies, tracks, and measures the range to scatterers in the pre-filtered image. An extended Kalman filter (EKF) uses the IMU to propagate the system state. The EKF uses the range measurement, the AGL estimate, GPS (when available), and the magnetometer to update the aircraft’s and scatterers’ states, correcting drift inherent in the IMU.

#### 3.1.1 AGL Measurement

The AGL measurement uses the radar as an altimeter. As can be seen in the range-compressed image shown in Figure 3.1.2, the upper portion of the image is relatively dark, indicating no radar return. As the range increases, nadir is measured as the range at which there is a large increase in return power as the radar signal reflects off of nadir (see [1] for more detail).
3.1.2 Resolving Strong Point Scatterers

Performing motion estimation with radar requires accurate detecting, associating, and tracking of point scatterers. When navigating an environment with very little radar return power, such as over a large body of water, strong point scatters are easily detected and tracked. In contrast, when navigating unknown environments containing clutter, noise sources, and a large number of strong and weak point-scatterers, the detection and tracking becomes significantly more difficult. The difficulty is amplified when using arbitrary point scatterers for navigation, as the difference between a “good” and “bad” point scatterer may be determined by the scatterer’s radar reflectivity, its motion, or even the reflectivity of interfering neighboring scatterers. Additionally, any error in detection, tracking, or association directly affects the motion estimation errors.

3.1.2.1 Scatter Detection

Range-compressed imagery has significant amounts of noise, as can be seen in Figure 3.1.2. Typically, a thresholding scheme is used to identify scatterers from noise. A thresholding scheme is used, however given the clutter and noise in the range-compressed image, some initial preprocessing must be performed. First, a weighted local pixel average is removed from the image. Second, the image is smoothed and then decimated, thus reducing the processing requirements of the system. Finally, a threshold is applied, resulting in a binary image indicating possible scatterers (see [1] for a more thorough description).
Figure 3.1.2: Range-compressed image from a real flight. The bright hyperbolas in the image are due to the range progression of strong point scatterers. The dark portion at the top of the image indicates no objects reflecting at that range. At nadir the return power drastically increases as ground scatterers reflect the radar signal back to the aircraft.

3.1.2.2 Data Association and Filtering

The process of tracking point-scatterers between time steps is an instance of the general multiple target tracking (MTT) problem, where both the number and individual state vectors of multiple signals are recursively estimated. Developing a robust MTT algorithm has been an active field of research since at least the 1960s. Since that time, numerous MTT tracking algorithms have been developed, including the global nearest neighbor [125], the joint probabilistic data association (JPDA) [126], the multiple hypothesis tracking (MHT) [127, 128], and the probability hypothesis density (PHD) [129, 130] filters.

Nearest neighbor algorithms, such as the GNN filter, associate the statistically nearest measurements to existing tracks. While computationally efficient, GNN has no mechanism to manage the initiation and deletion of new trackable objects. In their work, Kauffman et al. use a binary M-out-of-N (M/N) detector to identify persistent scatters and the GNN algorithm to filter new measurements to estimate the scatterer positions [17]. Unfortunately, the GNN state estimates can diverge in cluttered environments [123].
For this reason, our previous work used the Hough transform [44] to identify scatterers in the range-compressed image. While functional, the Hough transform is a batch algorithm and is very processing intensive. Additionally, the Hough transform is a voting algorithm with a finite set of possible coefficients; in [44], further processing is necessary to identify and track strong scatterers.

The other MTT algorithms mentioned above all suffer from at least one critical drawback: the JPDA filter does not have a track management algorithm, the MHT is computational complex and difficult to implement, and the PHD filter has poor track continuity in highly cluttered environments. Instead, the newly developed recursive-RANSAC (R-RANSAC) algorithm is used, which has been shown to build on RANSAC’s robustness, while handling multiple coefficient sets and sequential measurements [121, 122]. A review of the R-RANSAC algorithm is provided in the following section.

3.1.2.3 The Recursive-RANSAC (R-RANSAC) Algorithm

The R-RANSAC algorithm was originally developed in [121] to estimate the parameters of static signals. In [120], we apply R-RANSAC to estimate the location of point scatterers relative to an aircraft using range-compressed images. We show R-RANSAC to be as accurate as the Hough transform, while decreasing the processing time significantly. In this section, we review our original implementation of R-RANSAC to track point-scatterers to provide context for later improvements in the current iteration of this work.

At each time step, or chirp in a radar application, a set of measurements is received denoting the ranges to point-scatters or clutter. Rather than performing the standard RANSAC algorithm after each new chirp, R-RANSAC stores multiple scatterer hypotheses and identifies each new measurement as an outlier or inlier to existing hypotheses. Each outlier is used to seed the standard RANSAC algorithm to generate a new hypothesis from past data to describe the measurement. Inliers are used to update the parameters and consensus set of existing hypotheses [120].

The R-RANSAC algorithm is a modular framework that can incorporate various underlying data fusion techniques [122]. For example, after identifying the set of outliers, a measurement weighting can be assigned to each inlier measurement for use during the update step. In [120], an all neighbors approach is used, where all inlier measurements are averaged together into a single
psuedo-measurement. Alternatively, other measurement weighting techniques, such as nearest neighbor or probabilistic data association (PDA) [131], may be used.

3.1.2.4 Tracking

Our previous algorithms used the Hough transform and R-RANSAC exclusively for data association. Tracking the range progression of the point scatters involved an all-neighbors approach to detect deviations from the hyperbolic track. In this chapter the R-RANSAC algorithm is used for both data association and feature tracking. Rather than rely on ideal, hyperbolic range progression curves, we model the range progressions using polynomials. To support dynamic flight paths, a short time window is selected so that within the small window, the range curve resembles a second-order polynomial.

While the R-RANSAC algorithm is used for both data association and feature tracking, our navigational requirements demand that the aircraft and features co-exist in the state space model. Accordingly, the data association and tracking performed by R-RANSAC are used to provide the EKF with feature measurements. Additionally, R-RANSAC’s tracking is used to flag the EKF to create new scatters and prune stagnant scatterers from the state space model.

3.1.3 Implementing R-RANSAC for Data Association and Tracking

When accurate association and tracking is properly performed, as shown in Figure 3.1.3, accurate navigation is possible. Any error that exists is properly identified as the radar’s range measurement error or motion error of the aircraft.

In contrast, when association and tracking errors do occur, the error is improperly construed to be a range measurement error, or an aircraft motion error. Consider a few possible error scenarios:

- Clutter may be improperly identified as multiple point scatterers, as demonstrated in Figure 3.1.4a. The figure shows a point scatterer and clutter. While the lower scatterer is properly identified and tracked, the upper clutter is mistaken for multiple point scatterers. The tracked range progression of each of the clutter features is widely diverging.
Figure 3.1.3: Accurate association and tracking for real range-compressed image. The real range-compressed image is shown in gray, while the identified and tracked features are shown in black.

- Range progression drift can result in improper feature association, as shown in Figure 3.1.4b. The figure contains two distinct features, though the algorithm erroneously identifies sections of each scatterer as part of the same track. Not only does the track erroneously drift, but a new track is created for the new measurements of the original scatterer.

- Scatterer proximity (in range) may result in improper range measurements, as shown in Figure 3.1.4c. The figure shows three lower scatterer and a single offset scatterer. As the scatterers have similar ranges, the range return from the lower three scatterers are improperly identified as a measurement of the upper right track. As a result, proper identification of the bottom track is delayed.

- The tracking of features may prematurely end, as shown in the bottom track in Figure 3.1.4c. In the figure, the feature proximity resulted in a delayed identification and improper track characterization. The errors culminated in the feature track being prematurely killed.

Proper tracking involves carefully balancing of a variety of requirements. While short time windows allow for dynamic flight paths, they also increase the probability of unknowingly switching from one tracked scatterer to another. A high threshold allows for only features with a large number of measurements to be tracked, but also delays the identification of tracks, thus
resulting in a decreased number of measurements. These are particularly difficult as the aircraft’s antenna beamwidth limits the length of time that a scatterer is measurable.

To account for these requirements and limitations, a variety of modifications to the basic R-RANSAC framework are incorporated:

- When a new outlier is identified, rather than use RANSAC over all data points, it is used over a smaller window in range and time, surrounding the outlier under consideration. If a feature has sufficient neighboring outliers, the entire R-RANSAC window is used to identify all inliers.

Figure 3.1.4: Errors associated with incorrectly distinguishing clutter and features (3.1.4a), track drift (3.1.4b), and feature identification (3.1.4c). All show a real range-compressed image gray, while the identified and tracked features are shown in black.
• If a large period of time or a large distance in range separates consecutive inliers, the old samples are discarded and are considered as outliers.

• A constraint on the possible coefficient set is used to remove tracks that would be impossible given the aircraft’s dynamics.

• RANSAC is intermittently instantiated on new outliers, thus decreasing the processing requirements.

• Two inlier thresholds are used in the R-RANSAC algorithm. One threshold is used to identify initial tracks, while the second is a higher threshold that identifies tracks with sufficiently large inlier set to justify tracking in the EKF. While the first set of tracks are discarded whenever a better track is found, the second set is only discarded when it is stagnant or has fallen below its higher threshold.

• Many features have very fine range return, while others smear across multiple range bins. Accordingly, rather than use the total number of inliers to determine support, the number of time samples with an inlier is used to identify good tracks.

• A probabilistic data association scheme, with a Mahalanobis distance, is used to weight new measurements

3.1.4 Tracking Results

It is difficult to systematically assess the accuracy of the tracking solution using real data containing unknown point scatterers. While not ideal, using simulated data allows for accuracy of the tracking scheme to be assessed. A Monte Carlo simulation is performed using strong point scatterers. Figure 3.1.5a shows the simulated average tracking error for each identified feature. Only identified and tracked errors are measured. As can be seen, the average error is .35 meters, barely more than the 0.3 meter pixel resolution. Figure 3.1.5b indicates the number of range-compressed chirps where the point scatter was in the imagery, but not identified as a track. The average of 30 chirps is expected, given the clutter in real imagery and the delay required to identify new features.
Figure 3.1.5: Simulated error between truth and tracked measurements (3.1.5a) and the number of simulated chirps where the feature was not tracked (3.1.5b).

3.2 EKF and the Applied State-Space Model

This section describes our extended Kalman filter (EKF) shown in Figure 3.1.1. While an indirect Kalman filter approach could have allowed computational savings, the more simplistic and direct approach is used. Analysis indicates that either approach would result in comparable results [132–135]. The measurements are the IMU, a digital compass, the AGL estimate, and the range to each feature, as computed using the method described in the previous section. The output of the EKF are the aircraft position, velocity, and attitude estimates.

We assume the system model

\[
\begin{align*}
\dot{x} &= f(x, u) + \xi \\
y_k &= h(x_k) + v_k,
\end{align*}
\]

where \(\xi\) is the zero mean Gaussian system noise with covariance \(Q\), \(v_k\) is the zero mean Gaussian measurement noise with covariance \(R\), while \(y_k\) indicates the discrete system measurements at time \(k\).

In this chapter a continuous dynamics, discrete measurement, extended Kalman filter [118] is used,
Prediction Step:

\[
\dot{x} = f(\hat{x}, u) \quad (3.2.3)
\]
\[
\dot{P} = \frac{\partial f}{\partial x} P + P \frac{\partial f^T}{\partial x} + Q + \frac{\partial f}{\partial u} M \frac{\partial f^T}{\partial u}, \quad (3.2.4)
\]

Measurement Step:

\[
L = P^{-} \frac{\partial h}{\partial x} \left( R + \frac{\partial h}{\partial x} P^{-} \frac{\partial h^T}{\partial x} \right)^{-1} \quad (3.2.5)
\]
\[
P^{+} = \left( I - \frac{\partial h}{\partial x} \right) P^{-} \quad (3.2.6)
\]
\[
\hat{x}^{+} = \hat{x}^{-} = L \left( y_k - h(\hat{x}^{-}) \right), \quad (3.2.7)
\]

where \(\hat{x}^{-}\) and \(P^{-}\) denote \(\hat{x}\) and \(P\) after the prediction step, but before the measurement update, and \(\hat{x}^{+}\) and \(P^{+}\) denote \(\hat{x}\) and \(P\) after the measurement update. This section is organized as follows. First, the system state, consisting of both the aircraft and scatterer states is described. Second, the state prediction model is discussed. Third, the update model for each sensor is derived. The last section discusses new scatterer initialization.

3.2.1 System State

The system state is

\[
x = \left[ x^T_a \ x^T_s \right]^T, \quad (3.2.8)
\]

where \(x_a\) is the aircraft state and \(x_s\) is the states of the \(n\) tracked scatterers, given as

\[
x_s = \left[ x^T_1 \ \ldots \ x^T_n \right]^T. \quad (3.2.9)
\]
3.2.1.1 Aircraft State

The aircraft state is represented as

\[ x_a = \left[ \begin{array}{c} p_a^T \ p_a^T \ \Theta_a^T \end{array} \right]^T, \quad (3.2.10) \]

with the position of the aircraft defined is

\[ p_a = \left[ \begin{array}{c} n_a \ e_a \ h_a \end{array} \right]^T, \quad (3.2.11) \]

where \( n_a \) is the north position, \( e_a \) is the east position, and \( h_a \) is the altitude, and where the attitude is given by

\[ \Theta_a = \left[ \begin{array}{c} \phi_a \ \theta_a \ \psi_a \end{array} \right]^T, \quad (3.2.12) \]

where \( \phi_a \), \( \theta_a \), and \( \psi_a \) represent the roll, pitch, and yaw respectively [119].

3.2.1.2 Point Scatterer State

The ground point scatterer state is represented as

\[ x_i = \left[ \begin{array}{c} r_i^g \\ \psi_i \end{array} \right] = \left[ \begin{array}{c} \sqrt{(n_i - n_a)^2 + (e_i - e_a)^2} \\ \arctan \left( \frac{e_i - e_a}{n_i - n_a} \right) \end{array} \right], \quad (3.2.13) \]

where \( n_i \) and \( e_i \) are the north and east coordinates of the scatterer, \( r_i^g \) is the ground range to the scatterer, and \( \psi_i \) is the azimuth angle represented in the aircraft’s body frame, \( \psi_i \), shown in Figure 3.2.1. Assuming a flat earth, the elevation angle is represented as

\[ \phi_i = \arcsin \left( \frac{h_i - h_a}{r_i} \right). \quad (3.2.14) \]

The state model allows for the inertial position of the scatterer to be calculated as

\[ p_i = \left[ \begin{array}{c} n_i \\ e_i \\ h_i \end{array} \right] = \left[ \begin{array}{c} n_a \\ e_a \end{array} \right] + \left[ \begin{array}{c} r_i^g \cos \psi_i \\ r_i^g \sin \psi_i \end{array} \right]. \quad (3.2.15) \]
3.2.2 EKF State Prediction Model

A gyro and accelerometer are used as the input for the prediction model

\[
\mathbf{u} = \begin{bmatrix} \mathbf{a}^b \transpose \ 
\omega \transpose \end{bmatrix} \transpose, \tag{3.2.16}
\]

where \( \mathbf{a}^b \) are the accelerometer measurements, as measured in the body frame, and \( \omega \) are the angular body rates measured by the gyros.

The prediction model is expressed as

\[
\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} f_0(\mathbf{x}, \mathbf{u}) \\
\vdots \\
f_N(\mathbf{x}, \mathbf{u}) \end{bmatrix}, \tag{3.2.17}
\]
for $N$ scatterers. The aircraft dynamics are defined as

$$
fa(x,u) = \begin{bmatrix}
v \\
g^i + R^b_i(\phi, \theta, \psi) a^b \\
S(\Theta) \omega
\end{bmatrix},
$$

(3.2.18)

where $g^i = \begin{bmatrix} 0 & 0 & g \end{bmatrix}^T$ is the gravity vector in the inertial frame,

$$
S(\Theta) = \begin{bmatrix}
1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\
0 & \cos \phi & -\sin \phi \\
0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta}
\end{bmatrix},
$$

(3.2.19)

and $R^b_i(\phi, \theta, \psi)$ is the transformation from the body frame to the inertial frame.

The scatterer state propagation model is

$$
\dot{x}_i = \begin{bmatrix} \dot{r}_i^g \\
\dot{\psi}_i
\end{bmatrix}
$$

(3.2.20)

$$
= \begin{bmatrix}
-\dot{n}_a \cos \psi_i - \dot{e}_a \sin \psi_i \\
\frac{1}{\dot{r}_i}(\dot{n}_a \sin \psi_i - \dot{e}_a \cos \psi_i)
\end{bmatrix} \triangleq f_i(x_i, u_i),
$$

where

$$
u_i = \begin{bmatrix} n_a & e_a \end{bmatrix}^T.
$$

(3.2.21)

The Jacobian of the aircraft dynamics with respect to the aircraft state are given by

$$
\frac{\partial f_a}{\partial x_a}(x,u) = \begin{bmatrix}
0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} \\
0_{3 \times 3} & 0_{3 \times 3} & \frac{\partial R^b_i(\Theta)a^b}{\partial \Theta} \\
0_{3 \times 3} & 0_{3 \times 3} & \frac{\partial S(\Theta) \omega}{\partial \Theta}
\end{bmatrix}.
$$

(3.2.22)
where

\[
\frac{\partial R_i^b(\Theta)}{\partial \Theta} a^b = a_x \begin{bmatrix} 0 & -s_\theta c_\psi & -c_\theta s_\psi \\ 0 & -s_\theta s_\psi & c_\theta c_\psi \\ 0 & -c_\theta & 0 \end{bmatrix} + \\
\begin{bmatrix} c_\phi s_\theta c_\psi + s_\phi s_\psi & s_\phi c_\theta c_\psi & -s_\phi s_\theta s_\psi - c_\phi c_\psi \\ c_\phi s_\theta s_\psi - s_\phi c_\psi & s_\phi c_\theta s_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi \\ -r g_i & -s_\theta s_\phi & 0 \\
\end{bmatrix},
\]

(3.2.23)

and

\[
\frac{\partial S(\Theta)}{\partial \Theta} \omega = \begin{bmatrix} q c_\phi t_\theta - r s_\phi t_\theta & q s_\phi c_\psi + r c_\psi c_\phi & 0 \\ -q s_\phi - r c_\phi & 0 & 0 \\ q c_\phi - r s_\phi c_\psi & q s_\phi - r c_\phi c_\psi & 0 \end{bmatrix},
\]

(3.2.24)

with \(c_\alpha \triangleq \cos(\alpha), s_\alpha \triangleq \sin(\alpha),\) and \(t_\alpha \triangleq \tan(\alpha).\) Similarly, the Jacobians with respect to the input are given by

\[
\frac{\partial f_i(x, u)}{\partial u} = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} \\ R_i^b(\Omega) & 0_{3 \times 3} \\ 0_{3 \times 3} & R_i^b(\Theta) \end{bmatrix},
\]

(3.2.25)

and \(\frac{\partial f_i(x, u)}{\partial u} = 0_{2 \times 1}.\)

The Jacobian of the scatterer states with respect to the aircraft state is

\[
\frac{\partial f_i(x, u)}{\partial x_a} = \begin{bmatrix} 0_{1 \times 3} & -\cos \psi_i & \frac{\sin \psi_i}{r_i^f} & 0_{1 \times 4} \\ 0_{1 \times 3} & -\sin \psi_i & -\frac{\cos \psi_i}{r_i^f} & 0_{1 \times 4} \end{bmatrix},
\]

(3.2.26)
while the Jacobian of the scatterer states with respect to the scatterer states is

$$\frac{\partial}{\partial x} f_i(x, u) = \begin{bmatrix} 0 & -n_a \sin \psi_i - e_a \cos \psi_i \\ \frac{1}{(r_i^2)} (-n_a \sin \psi_i + e_a \cos \psi_i) & \frac{n_a \cos \psi_i + e_a \sin \psi_i}{r_i^2} \end{bmatrix}.$$ (3.2.27)

When $i \neq j$, $\frac{\partial f_i}{\partial x_j}(x, u) = \mathbf{0}_{2 \times 2n}$.

### 3.2.3 Update Sensor Dynamics

#### 3.2.3.1 GPS

GPS measures

$$y_{\text{GPS}} = \begin{bmatrix} n_a & e_a & h_a & V_g & \chi \end{bmatrix}^\top,$$ (3.2.28)

where the ground velocity is defined as

$$V_g = \sqrt{n_a^2 + e_a^2},$$ (3.2.29)

and the heading is defined as

$$\chi = \tan \left( \frac{n_a}{e_a} \right).$$ (3.2.30)

#### 3.2.3.2 Magnetometer

Magnetometers measure the earth’s magnetic field in three dimensions, while a digital compass is used, which measures $\psi$, resulting in

$$y_{\text{mag}} = \psi.$$ (3.2.31)

#### 3.2.3.3 Coordinated Turn Pseudo-Measurement

The standard coordinated turn condition

$$\psi = \frac{g}{V} \tan \psi.$$ (3.2.32)
is imposed to make the roll angle observable. From Equation 3.2.18, the turn condition becomes

$$
\dot{\psi} = \frac{\sin \phi}{\cos \theta} q + \frac{\cos \phi}{\cos \theta} r.
$$

(3.2.33)

Accordingly, the coordinated turn pseudo-measurement (which should nominally be zero) is

$$
y_{\text{turn}} = \frac{\sin \phi}{\cos \theta} q + \frac{\cos \phi}{\cos \theta} r - \frac{g}{\sqrt{n_a^2 + e_a^2}} \tan \phi.
$$

(3.2.34)

### 3.2.3.4 AGL

Using nadir, the AGL measurement is defined as

$$
y_{\text{AGL}} = -h.
$$

(3.2.35)

### 3.2.3.5 Scatterer Range Measurement

The range to each scatterer is modeled as

$$
r_i = \sqrt{(n_i - n_a)^2 + (e_i - e_a)^2 + h_a^2}
$$

(3.2.36)

$$
= \sqrt{(r_i^y)^2 + h_a^2},
$$

(3.2.37)

with the resulting range measurements given by

$$
y_r = \left[ r_1 \ldots r_N \right]^T.
$$

(3.2.38)

### 3.2.3.6 Relative Range Pseudo-Measurement

While the relative positioning of each scatter is arbitrary, the relative range is measured when scatterers are both measured during the same chirp. This may be represented as

$$
r_{ij} = r_i - r_j,
$$

(3.2.39)
with the resulting relative range measurements given by

\[ y_\Delta = \begin{bmatrix} r_{12} & \ldots & r_{(N-1)N} \end{bmatrix}^T. \]  

(3.2.40)

### 3.2.3.7 Measurement Model

The resulting measurement model is given by

\[ y_k = h(x_k) + \eta_k, \]  

(3.2.41)

where

\[ h(x) = \begin{bmatrix} h_{\text{GPS}} & h_{\text{mag}} & h_{\text{turn}} & h_{\text{AGL}} & h_t & h_\Delta \end{bmatrix}^T \]  

(3.2.42)

and \( \eta_k \) is the measurement noise.

The corresponding Jacobian is

\[
\frac{\partial h}{\partial x} = \begin{bmatrix}
\frac{\partial h_{\text{GPS}}}{\partial x_a} & \frac{\partial h_{\text{GPS}}}{\partial x_s} \\
\frac{\partial h_{\text{mag}}}{\partial x_a} & \frac{\partial h_{\text{mag}}}{\partial x_s} \\
\frac{\partial h_{\text{turn}}}{\partial x_a} & \frac{\partial h_{\text{turn}}}{\partial x_s} \\
\frac{\partial h_{\text{AGL}}}{\partial x_a} & \frac{\partial h_{\text{AGL}}}{\partial x_s} \\
\frac{\partial h_t}{\partial x_a} & \frac{\partial h_t}{\partial x_s} \\
\frac{\partial h_\Delta}{\partial x_a} & \frac{\partial h_\Delta}{\partial x_s}
\end{bmatrix}.
\]  

(3.2.43)

The Jacobian, with respect to the aircraft state \( x_a \), for the non-scatterer state terms are

\[
\frac{\partial h_{\text{GPS}}}{\partial x_a} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{\dot{\eta}_g}{V_g} & \frac{\ddot{\eta}_g}{V_g^2} & 0 & 0 & 0 \\
0 & 0 & 0 & -\frac{1}{V_g^2} & -\frac{\dot{\eta}_g}{V_g^2} & 0 & 0 & 0
\end{bmatrix},
\]  

(3.2.44)
\[ \frac{\partial h_{\text{mag}}}{\partial x_a} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \] (3.2.45)
\[ \frac{\partial h_{\text{turn}}}{\partial x_a} = \begin{bmatrix} 0 & 0 & 0 & H_1(x) & H_2(x) & 0 & H_3(x) & H_4(x) & 0 \end{bmatrix}, \] (3.2.46)
\[ \frac{\partial h_{\text{AGL}}}{\partial x_a} = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \] (3.2.47)

where

\[ H_1(x) = \frac{gn_a \tan \phi}{V_g^3}, \] (3.2.48)
\[ H_2(x) = \frac{g \dot{e}_a \tan \phi}{V_g^3}, \] (3.2.49)
\[ H_3(x) = \frac{\cos \phi}{\cos \theta} q - \frac{\sin \phi}{\cos \theta} r - \frac{g \sec^2 \phi}{V_g}, \] (3.2.50)
\[ H_4(x) = \left( \frac{\sin \phi}{\cos \theta} q + \frac{\cos \phi}{\cos \theta} r \right) \tan \theta. \] (3.2.51)

As the non-scatterer measurements \((h_{\text{GPS}}, h_{\text{turn}}, h_{\text{mag}}, \text{and } h_{\text{AGL}})\) are measured irrespective of the scatterer positions, their Jacobians with respect to the scatterer states, are all zero.

Calculating the partial derivative of the range measurement \(h_{r,i}\), for scatterer \(i\), with respect to the aircraft state is

\[ \frac{\partial}{\partial x_a} h_{r,i}(x) = \begin{bmatrix} 0 & 0 & h_a & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \] (3.2.52)

while calculating the partial derivative of \(h_{r,i}\) with respect to the scatterer state is

\[ \frac{\partial}{\partial x_i} h_i(x) = \begin{bmatrix} r_i^s & 0 \end{bmatrix}, \] (3.2.53)

where \(\frac{\partial}{\partial x_j} h_i(x) = \begin{bmatrix} 0 & 0 \end{bmatrix}\) when \(i \neq j\).

The partial derivative of the relative scatterer range measurement \(h_\Delta\) with respect to the state is
and

$$\frac{\partial}{\partial \mathbf{x}_i} h_{ij}(\mathbf{x}) = \begin{bmatrix} \frac{r_i^2}{r_i} & 0 \\ \frac{r_j^2}{r_j} & 0 \end{bmatrix}, \quad (3.2.55)$$

and

$$\frac{\partial}{\partial \mathbf{x}_j} h_{ij}(\mathbf{x}) = \begin{bmatrix} -\frac{r_i^2}{r_i} & 0 \end{bmatrix}, \quad (3.2.56)$$

with

$$\frac{\partial}{\partial \mathbf{x}_m} h_{ij}(\mathbf{x}) = \mathbf{0}_{1 \times 2}, \quad (3.2.57)$$

when $m \neq i$ and $m \neq j$.

### 3.2.3.8 Range Rate

While the R-RANSAC algorithm performs a polynomial fit to the range progression, it also estimates the range rate $\dot{r}_i$ to each scatterer. The range rate may be calculated as

$$\dot{r}_i = \dot{n}_a \cos \psi_i \cos \phi_i + \dot{e}_a \sin \psi_i \cos \phi_i + \dot{h}_a \sin \phi_i. \quad (3.2.58)$$

By defining

$$\begin{bmatrix} \dot{n}_a \\ \dot{e}_a \end{bmatrix} = \begin{bmatrix} V_g \cos \psi_a \\ V_g \sin \psi_a \end{bmatrix}, \quad (3.2.59)$$
and

\[ V_g = \sqrt{h_a^2 + e_a^2}, \quad (3.2.60) \]

the range rate equation may be written as

\[ \frac{\dot{r}_i - \dot{h}_a \sin \phi_i}{\cos \phi_i} = V_g \cos (\psi_i - \psi_a). \quad (3.2.61) \]

Using Equation 3.2.14 and the flat-earth assumption, the azimuth angle may be calculated as

\[ \psi_i = \cos \left( \frac{\dot{r}_i \dot{r}_i + \dot{h}_a \dot{h}_a}{\sqrt{\dot{r}_i^2 - \dot{h}_a^2} \sqrt{n_a^2 + e_a^2}} \right) + \psi_a, \quad (3.2.62) \]

allowing for the azimuth angle to be estimated from the scatterer state and range rate. Note that the result may be infinite, but this only occurs when \( V_g = 0 \) (ie. the aircraft has no ground velocity) or when \( \phi_i = 0 \) (ie. the aircraft is at the same elevation as the scatterer).

### 3.2.4 New Scatterer State Initialization

For SAR systems, the typical antenna beamwidth and aircraft speeds often results in point scatterers being illuminated for only a few seconds (as shown in Figure 3.1.2). During this small time window, it is necessary to identify, track, and estimate the position of the scatterer with sufficient accuracy to estimate small variations in the aircraft’s position. Accordingly, an increased accuracy in each scatterer’s initial state estimate greatly improves the estimates of the aircraft’s position. For this reason, a range-rate pseudo-measurement is used to provide an initial azimuth angle measurement.

When a new scatterer \( i \) is first identified, at time \( t_k \), it’s state is initialized as

\[ \mathbf{x}_i [t_k^+] = \begin{bmatrix} r_i^g [t_k^+] \quad \psi_i [t_k^+] \end{bmatrix}^\top, \quad (3.2.63) \]

where \( r_i^g [t_k^+] \) is the ground range estimate calculated

\[ r_i^g [t_k^+] = \sqrt{r_i^2 [t_k] - h_a^2 [t_k]}, \quad (3.2.64) \]
and $\psi_f^+_{t_k}$ is the azimuth angle pseudo-measurement calculated

$$\psi_f^+_{t_k} = \text{acos} \left( \frac{\dot{r}_i [t_k] r_i [t_k] + h_a [t_k^-] h_a [t_k^-]}{\sqrt{r_i^2 [t_k] - h_a^2 [t_k^-] \sqrt{h_a^2 [t_k^-] + e_a^2 [t_k^-]}}} \right) + \psi_a [t_k^-], \quad (3.2.65)$$

where $r_i [t_k]$ is the R-RANSAC measurement, $\dot{r}_i [t_k]$ is the R-RANSAC pseudo-measurement, and $h_a [t_k^-], \dot{h}_a [t_k^-], \dot{e}_a [t_k^-], h_a [t_k^-], \psi_a [t_k^-]$ are current aircraft state estimates.

Additionally, the new scatterer EKF covariance terms are initialized as

$$\text{cov} \left[ x_i^* x_i^{*T} \right] = \begin{bmatrix} \sigma_r & 0 \\ 0 & \sigma_\phi \end{bmatrix}, \quad (3.2.66)$$

$$\text{cov} \left[ x_a^* x_a^{*T} \right] = \textbf{0}_{9 \times 2} \quad (3.2.67)$$

where the range and elevation angle standard deviation are $\sigma_r^2$ and $\sigma_\phi^2$, and where $i \neq j$. The scatterer covariance terms, when $i \neq j$, are set to

$$\text{cov} \left[ x_i^* x_j^{*T} \right] = \textbf{0}_{2 \times 2} \quad (3.2.68)$$

### 3.3 Results

As mentioned in Section 3.1.2.4, a variety of association and tracking errors can and do occur. Accordingly, a simulation was run to demonstrate the accuracy of our approach where no data association errors occur. Figure 3.3.1 shows the average, best, and worst results from the 10 Monte Carlo simulations (from Section 3.1.4) when integrated with an EKF and a navigation grade IMU. During the simulation the aircraft navigated for 60 seconds, moving approximately 2.7 km. The unknown point scatterers were positioned on a periodic grid. The IMU drift is approximately 260 meters (or 9.6% drift rate), while the worst-case drift error of the EKF was approximately 2 meters (0.076% drift), with a worst-case drift rate of 2.2 meters (or .076% drift), and a best-case drift of 1.96 meters (or 0.073% drift).

The real flight tests were performed using a SAR mounted to a Cessna aircraft. A SAR image formed from the flight data is shown in Figure 3.3.2. The elevation rises approximately 50 meters during the course of the flight. As the algorithm assumes a flat earth model, an elevation-
compensated GPS was used as truth data for the results. From the start to the end of the flight track there is also a 7 to 29 meter elevation drop in the cross-track direction. In addition to the violated flat terrain assumption, many point scatterers from features found in trees and on buildings may be positioned significantly higher than ground level. As the test data was provided after the data collect, all results are post-processed, though the computational complexity of the algorithm allows for real time implementation on modest hardware.

During the flight the readings from a navigation-grade IMU were recorded. The IMU’s gyro has a 1-sigma standard deviation of 1 deg/hr, while its accelerometer standard deviation is 0.0003 g, more precise than those found on small UAVs. The test involves initially using the elevation-compensated GPS, during which time the algorithm estimates the point scatterer locations. After 16 seconds of flight (during which the aircraft traveled approximately 560 meters), the GPS signal was “lost” and the test continued for 60 seconds (around 2.4 km). A simulated consumer-grade magnetometer, with a standard deviation of 0.3 degrees, is used as a bearing measurement.

Figure 3.3.3 shows the results of the radar odometry position estimation error after GPS signal is lost. As can be seen, the data association errors, due to the extremely cluttered environment, degraded the results during the first twenty seconds of the radar collect. As the flight continued
Figure 3.3.2: Processed image formed from real flight data.

Figure 3.3.3: Radar odometry real flight error. The solid black line indicates the drift error with the navigation-grade IMU alone, while the blue line indicates the average of 10 Monte Carlo results. The best and worst results are shown in red and green respectively.
however, the proper association and tracking resulted in improved results, though a single flight track improperly deviated for the time between 48 and 54 seconds. Throughout the flight many clutter and noise sources are improperly identified and tracked, resulting in a wide variation between each of the Monte Carlo results. Compared to the unaided IMU, the radar odometry results performs well. The unaided IMU drifts 209 meters (or approximately 8.7% drift rate), while the average radar odometry drift rate is 92 meters (3.8% drift rate). While the worst-case drift was 165 meters (6.9%), the best drift was 49 meters (2.0%). All radar odometry results were better than the navigation grade IMU. With a commercial grade IMU, even better improvements are expected.

3.4 Conclusion

When GPS is lost or denied, accurate navigation of current UAVs is subject to the IMU’s drift rates. We have demonstrated that in real, cluttered environments, R-RANSAC is able to accurately identify and track arbitrary point-scatterers. Additionally, we have shown that the radar odometry algorithm developed in this chapter is able to improve the real flight drift error to an average 3.8%, as compared to 8.7% when using an unaided navigation-grade IMU. The results indicate that radar odometry is a feasible and robust solution for navigating in unknown, GPS-denied environments.
Chapter 4

Radar Odometry and Range-Only Observability

Chapter 3 discussed a radar odometry approach that uses the Recursive-RANSAC (R-RANSAC) algorithm to perform both data association and tracking [2, 120]. The R-RANSAC algorithm reports significant computational improvements over the Hough transform and demonstrates similar feature identification accuracy [120]. While real and simulated results present a decreased drift rate during GPS-denied navigation, aspects of the results warrant further analysis. Specifically, the real results demonstrate a wide disparity between the best-case and worst-case drifts (at one point the error difference is 198 m). This is particularly troublesome as it is for a single data set. Of further concern is the length of test, which spans only a 60 second window.

In this chapter we explore the validity of the radar odometry approach. First, the observability of the system is considered. Second, using the resulting analysis, a framework is presented to allow for observable, GPS-denied radar odometry navigation in an unknown environment. Third, we present a more realistic simulation model, which is used to explore the radar odometry accuracy over longer flights. Additionally, this chapter presents enhancements to the scatterer identification and tracking that reduces the disparity between the best-case and worst-case drift rates. A more robust technique for above ground level (AGL) estimation is presented, in contrast with the previous thresholding technique. Finally, the approach is validated with real flight data.

This chapter is organized as follows. Section 4.1 describes the observability analysis with range-only measurements. Section 4.2 describes the enhancements to the radar odometry algorithm, while Section 4.3 describes the implemented extended Kalman filter (EKF). The simulated and real results are presented in Section 4.4.
4.1 Observability Analysis for Navigation with Range-Only Measurements

While various approaches exist for motion estimation with a range-only sensor, the observability of these systems (such as with a single synthetic aperture radar) has yet to be considered. For bearing-only sensors, the observability of navigation using unknown features has been thoroughly explored [136–141], with [141] concluding that at least two known features must be tracked for observability. Observability analysis for range-only sensors is considerably less thorough. Gadre et. al. considers the observability of a navigating system using range measurements from a single, known location [142], while others consider target tracking using range-only sensors [143] or multi-agent navigation using range-only sensors [64]. Even with range and bearing measurements of a simultaneous localization and mapping (SLAM) system, observability analysis shows that navigation with unknown feature locations is rank deficient, no matter the number of features [144, 145].

4.1.1 Nonlinear Observability Criteria

For the observability analysis in this work, the nonlinear observability criteria presented by Hermann and Krener [146] is used, which is summarized as follows:

Consider the system model

\[ \dot{X} = F(X, u) \]  \hspace{1cm} (4.1.1)
\[ Y = h^G(X) = \begin{bmatrix} h^1_1(X) & \ldots & h^G_1(X) \end{bmatrix}, \]  \hspace{1cm} (4.1.2)

where \( X \in \mathbb{R}^A \) is the system state, \( Y : \in \mathbb{R}^G \) is the measurement output of the system, \( u \in \mathbb{R}^B \) is the input vector, and \( F : \mathbb{R}^A \times \Lambda \rightarrow \mathbb{R}^A \). For the analysis, consider the special case where \( F \) may be divided into a summation of each component of the control vector

\[ X = F(X, u) = \sum_{b=1}^{B} f_b u_b, \]  \hspace{1cm} (4.1.3)
where \( u_b \triangleq \begin{bmatrix} u_1 & \ldots & u_B \end{bmatrix}^\top \). [146] demonstrates that the system is \textit{locally weakly observable} if the combined observability matrices consisting of

\[
O_g = \begin{bmatrix}
\nabla L^0_h(X) \\
\nabla L^1_h(X) \\
\vdots \\
\nabla L^p_h(X)
\end{bmatrix},
\]

are of rank \( A \), for \( 1 \leq g \leq G \), and for any \( p \), where \( \nabla \) is the gradient operator. The elements of \( O_g \) consist of \( L^D_h(X) \), the \( D^{th} \) repeated Lie derivatives of the \( g^{th} \) component of \( \nabla h \) with respect to \( F(X, u) \). Specifically, a zeroth order Lie derivative is defined as

\[
L^0_h = h,
\]

while the \( D^{th} \)-order Lie derivative is expressed

\[
L^D_h = \nabla L^{D-1}_h \cdot F,
\]

for \( D > 0 \), where \( \cdot \) is the vector inner product.

### 4.1.2 State-Space Model

Radar odometry seeks to estimate the aircraft’s motion using radar range measurements during GPS-denied navigation. Accordingly, the observability analysis considers a simplified system consisting of the aircraft’s north, east, and down coordinates and the aircraft’s heading, represented as

\[
X_a = \begin{bmatrix}
n_a & e_a & d_a & \psi_a
\end{bmatrix}^\top.
\]

The state of each scatterer is also represented in the Cartesian coordinate frame,

\[
X_i = \begin{bmatrix}
n_i & e_i
\end{bmatrix}^\top,
\]
resulting in the $2N + 4$ dimensional state-space model

$$X = \begin{bmatrix} X_a \\ X_1 \\ \vdots \\ X_N \end{bmatrix}, \quad (4.1.9)$$

where $N$ scatterers are represented.

By defining the three dimensional control input vector

$$u = \begin{bmatrix} v_g \\ v_d \\ \omega \end{bmatrix},$$

consisting of $v_g$, the linear ground velocity, $v_d$, the linear elevation velocity, and $\omega$, the aircraft’s heading, the process model is defined as

$$\dot{X} = F(X, u) = \begin{bmatrix} v_g \cos \psi \\ v_g \sin \psi \\ v_d \\ \omega \\ \mathbf{0}_{1\times2N} \end{bmatrix}, \quad (4.1.10)$$

where

$$\frac{\partial F(X, u)}{\partial X_a} = \begin{bmatrix} 0 & 0 & 0 & -v_g \sin \psi \\ 0 & 0 & 0 & v_g \cos \psi \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad (4.1.11)$$

$$\frac{\partial F(X, u)}{\partial X_i} = \mathbf{0}_{2N\times3}, \quad (4.1.12)$$

$$\frac{\partial F(X, u)}{\partial X_i} = \mathbf{0}_{3+2N\times2N}. \quad (4.1.13)$$
As required by Equation 4.1.4, $F$ may be divided into three components

$$\dot{X} = F(X, u) = f_{v_b}v_b + f_{vd}v_d + f_\omega \omega. \quad (4.1.14)$$

where

$$f_{v_b} = \begin{bmatrix} c_\psi & s_\psi & 0 & 0 & \mathbf{0}_{2n \times 1} \end{bmatrix}^\top, \quad (4.1.15)$$

$$f_{vd} = \begin{bmatrix} 0 & 0 & 1 & 0 & \mathbf{0}_{2n \times 1} \end{bmatrix}^\top, \quad (4.1.16)$$

and

$$f_\omega = \begin{bmatrix} 0 & 0 & 0 & 1 & \mathbf{0}_{2n \times 1} \end{bmatrix}^\top. \quad (4.1.17)$$

For the system, the measurement model consists of the AGL measurement $h_{AGL}$ and the range measurements $h_1$ through $h_N$ for $N$ point scatterers.

### 4.1.3 Lie Derivatives for the AGL Measurement, $h_{AGL}$

First consider the AGL sensor, its derivatives and gradients. The zeroth-order Lie derivative of $h_{AGL}$ is defined

$$L^0 h_{AGL} = d_a, \quad (4.1.18)$$

which has a gradient

$$\nabla L^0 h_{AGL} = \begin{bmatrix} 0 & 0 & 1 & 0 & \mathbf{0}_{1 \times 2n} \end{bmatrix}. \quad (4.1.19)$$

The higher order Lie derivatives are all 0.

### 4.1.4 Lie Derivatives for a Range Measurement, $h_i$

Next consider the Lie derivatives for the range measurements between between the aircraft and a single scatterer, $i$. The radar measures the range to a scatterer, though the derivation is
simplified by considering the squared range as

\[ h_i = r_i^2 = (n_i - n_a)^2 + (e_i - e_a)^2 + (d_i - d_a)^2. \]  

(4.1.20)

Given the flat earth assumption, the equation is simplified as

\[ h_i = (n_i - n_a)^2 + (e_i - e_a)^2 + d_a^2. \]  

(4.1.21)

Hereafter the Lie derivatives of \( h_i \) and its gradients are computed.

The zeroth-order Lie derivative of \( h_i \) is calculated as

\[ L^0 h_i = (n_a - n_i)^2 + (e_a - e_i)^2 + d_a^2 \]  

with its corresponding gradient

\[ \nabla L^0 h_i = \begin{bmatrix} n_a - n_i \\ e_a - e_i \\ d_a \\ 0 \\ 0_{1 \times 2(i-1)} \\ n_i - n_a \\ e_i - e_a \\ 0_{1 \times 2(n-i)} \end{bmatrix}^T. \]  

(4.1.22)

The non-zero first-order Lie derivatives are calculated

\[ L^1_{f_{vg}} h_i = \nabla L^0 h_i \cdot f_{vg} = (n_a - n_i) \cos \psi - (e_a - e_i) \sin \psi, \]  

(4.1.23)

\[ L^1_{f_{vd}} h_i = \nabla L^0 h_i \cdot f_{vd} = d_a, \]  

(4.1.24)
which result in the unique gradient

\[
\nabla L^1_{fg} h_i = \begin{bmatrix}
\cos \psi \\
\sin \psi \\
0 \\
(n_i - n_a) \sin \psi + (e_a - e_i) \cos \psi \\
0_{1 \times 2(i-1)} \\
-\cos \psi \\
-\sin \psi \\
0_{1 \times 2(n-i)}
\end{bmatrix}^T .
\tag{4.1.26}
\]

The only non-constant second-order Lie derivative is calculated

\[
L^2_{fg, f\omega} h_i = \nabla L^1_{fg} h_i \cdot f\omega = (n_i - n_a) \sin \psi + (e_a - e_i) \cos \psi,
\tag{4.1.27}
\]

with its corresponding gradient

\[
\nabla L^2_{fg, f\omega} h_i = \begin{bmatrix}
-\sin \psi \\
\cos \psi \\
0 \\
(n_i - n_a) \cos \psi + (e_i - e_a) \sin \psi \\
0_{1 \times 2(i-1)} \\
s \psi \\
-\cos \psi \\
0_{1 \times 2(n-i)}
\end{bmatrix}^T .
\tag{4.1.28}
\]

The only non-zero third-order Lie derivative is

\[
L^3_{fg, f\omega, f\omega} h_i = \nabla L^2_{fg, f\omega} h_i \cdot f\omega = (n_i - n_a) \cos \psi + (e_i - e_a) \sin \psi,
\tag{4.1.29}
\]
which has a gradient

\[
\nabla L_{f_{r,g} f_{a} f_{\omega}}^3 h_i = \begin{bmatrix}
-\cos \psi \\
-\sin \psi \\
0 \\
(n_a - n_i) \sin \psi + (e_i - e_a) \cos \psi \\
\mathbf{0}_{1 \times 2(i-1)} \\
\cos \psi \\
\sin \psi \\
\mathbf{0}_{1 \times 2(n-i)}
\end{bmatrix}
\]

\[= -\nabla L_{f_{r,g}}^1 h_i. \quad (4.1.30)\]

which is a multiple of a previous gradient. All higher-order Lie derivatives are zero or are multiples of lower-order Lie derivatives.

### 4.1.4.1 Observability Analysis for a Single Scatterer

Given the AGL estimate \( h_{AGL} \) and the a single range measurement, \( h_i \), the observability matrix may be represented as

\[
\mathcal{O}_i (h_i, h_{AGL}) = \begin{bmatrix}
\nabla L^0 h_{AGL} \\
\nabla L^0 h_i \\
\nabla L^1_{f_{r,g}} h_i \\
\nabla L^2_{f_{r,g} f_{a}} h_i
\end{bmatrix}, \quad (4.1.32)
\]
which when transformed into its row reduced echelon form (RREF), is

\[
\mathcal{O}_i(h_i, h_{AGL}) = 
\begin{bmatrix}
0 & 0 & 1 & 0 & \mathbf{0}_{1 \times 2(N-i)} & 0 & \mathbf{0}_{1 \times 2(N-i)} \\
1 & 0 & 0 & (e_a - e_i) & \mathbf{0}_{1 \times 2(N-i)} & -1 & \mathbf{0}_{1 \times 2(N-i)} \\
0 & 1 & 0 & (n_i - n_a) & \mathbf{0}_{1 \times 2(N-i)} & 0 & -1 & \mathbf{0}_{1 \times 2(N-i)}
\end{bmatrix}.
\] (4.1.33)

For a single scatterer, \(N = 1\), the system has 6 states, with 3 linearly independent columns, 3 states deficient of being observable. It is worth noting that the third, altitude, column is linearly independent from all others, while the northern and eastern position of the aircraft is proportional to the corresponding scatterer states.

### 4.1.5 Lie Derivatives for \(N\) Range Measurements

Now consider a system containing \(N\) scatterers, where the aircraft’s range-squared measurement to each scatterer is represented

\[
\mathbf{h}^N = 
\begin{bmatrix}
h_1 \\
\vdots \\
h_N
\end{bmatrix}.
\] (4.1.34)

The resulting observability matrix may be represented as a combination of each of the scatterer matrices observability matrices, \(1 \leq n \leq N\),

\[
\mathcal{O}_N(\mathbf{h}^N, h_{AGL}) = 
\begin{bmatrix}
\mathcal{O}_1 \\
\mathcal{O}_2 \\
\vdots \\
\mathcal{O}_N
\end{bmatrix},
\] (4.1.35)
which may be represented in its RREF form as

\[ \mathcal{O}_N(h^N, h_{AGL}) = \begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\
1 & 0 & 0 & (e_a - e_1) & -1 & 0 & 0 & \cdots & 0 & 0 \\
0 & 1 & 0 & (n_1 - n_a) & 0 & -1 & 0 & \cdots & 0 & 0 \\
1 & 0 & 0 & (e_a - e_2) & 0 & 0 & -1 & \cdots & 0 & 0 \\
0 & 1 & 0 & (n_2 - n_a) & 0 & 0 & 0 & -1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
1 & 0 & 0 & (e_a - e_N) & 0 & 0 & 0 & \cdots & -1 & 0 \\
0 & 1 & 0 & (n_N - n_a) & 0 & 0 & 0 & \cdots & 0 & -1 
\end{bmatrix} \] (4.1.36)

No matter the number of scatterers, the elevation remains linearly independent (as shown by the third column). As the positions of scatterers are only represented relative to the aircraft’s position, each scatterers’ 2D position remains represented relative to the aircraft’s north and eastern coordinates. While the matrix has \(2N + 4\) states, there are only \(2N + 1\) linearly independent columns, three states deficient of being full rank.

### 4.1.5.1 Observability with an Absolute Bearing Measurement

One approach to resolve the aircraft’s heading state involves adding an absolute bearing measurement, represented as \(h_\psi = \psi\). The zeroth-order Lie derivative of \(h_\psi\) is

\[ L^0 h_\psi = \psi, \] (4.1.37)

with its corresponding gradient,

\[ \nabla L^0 h_\psi = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0_{1 \times 2n} \end{bmatrix}. \] (4.1.38)

Analysis indicates that higher order Lie derivatives are zero.
Given the range measurements, the AGL measurement, and the bearing measurement, the resulting observability matrix from Equation 4.1.36 may be appended to become

\[
O_N^* (h^N, h_{AGL}, h_\psi) =
\begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\
1 & 0 & 0 & 0 & -1 & 0 & 0 & \cdots & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & -1 & 0 & \cdots & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & -1 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & -1 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & -1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\
\end{bmatrix}, \quad (4.1.39)
\]

where the new row is shown in italic. The new matrix is rank \(2N+2\), leaving 2 states unobservable.

Two approaches may be used to resolve the two remaining unobservable states.

**Observability with an Additional Single Known Scatterer Position**

Given a scatterer \(j\), in a known location, the measurement model may represented as \(h'_j = \begin{bmatrix} n_j & e_j \end{bmatrix}^\top\), which has a zeroth-order Lie derivative,

\[
L_0^0 h'_j = n_j, \quad (4.1.40)
\]

\[
L_0^0 h'^e_j = e_j, \quad (4.1.41)
\]

with the resulting gradient

\[
\nabla L_0^0 dh'_j (X) =
\begin{bmatrix}
0 & 0 & 0 & 0_1 \times 2(j-1) & 1 & 0 & 0_1 \times 2(n-j) \\
0 & 0 & 0 & 0_1 \times 2(j-1) & 0 & 1 & 0_1 \times 2(n-j) \\
\end{bmatrix}, \quad (4.1.42)
\]
If $\mathcal{O}_N(h^N, h_{AGL}, h_\psi)$ were modified so that the first scatterer was in a known location, the resulting observability matrix would become

$$\mathcal{O}_{1,N}(h^n, h_{AGL}, h_\psi, h'_1) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}, \quad (4.1.43)$$

which shows the two new rows in italic. The new matrix is Rank $2N + 4$, a linearly independent, fully observable matrix.

**Observability with an Additional Known Aircraft Position** An alternative approach to achieving observability involves knowing the north and east coordinates of the aircraft, such as could be measured with GPS. Knowing the position of the aircraft, as represented by the measurement model $h_{GPS} = \left[ n_a \ e_a \ d_a \right]^\top$. The GPS model has the zeroth-order Lie derivatives

$$L^0 h_{GPS}(X) = \left[ n_a \ e_a \ d_a \right]^\top,$$  

with gradient

$$\nabla L^0 dh_{GPS}(X) = \begin{bmatrix} 1 & 0 & 0 & 0_{1 \times 2N} \\ 0 & 1 & 0 & 0_{1 \times 2N} \\ 0 & 0 & 1 & 0_{1 \times 2N} \end{bmatrix}.$$

75
Using the GPS in conjunction with $O_N(h^N, h_{AGL}, h_\psi)$, results in the observability matrix

$$
O_{GPS,N}(h^n, h_{AGL}, h_\psi, h_{GPS}) = 
\begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & \cdots & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & -1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\
\end{bmatrix}
$$

(4.1.46)

with the new rows represented in italics. $O_{GPS,N}$ is full rank.

### 4.1.6 Applying Observability Analysis Findings

Given the observability analysis, the system is observable when the following measurements are available:

- The range to $N \geq 1$ scatterers,
- The aircraft AGL,
- Aircraft bearing,

in addition to at least one of the following measurements

- Aircraft north and east position,
- The absolute position of a single scatterer.

Rather than require the agent to navigate in the vicinity of a known scatterer, consider an operation where an observable system is maintained by initially navigating with GPS. Using GPS, the agent estimates the position of any tracked point scatterers, resulting in a fully observable system.
When GPS is lost, the agent maintains its previously estimated scatterer positions, maintaining the systems observability. As the ranges to newly identified scatterers are measured, the previously estimated scatterer states are used to estimate the new scatterers, continuing to maintain the system’s observability. Using this approach, the radar odometry algorithm described in this chapter is observable as long as at least one previously estimated scatterer remains in the sensors field-of-view.

4.2 R-RANSAC Based Radar Odometry

This section describes the enhanced, observable, radar odometry system presented in this chapter. Figure 4.2.1 shows a block diagram of the approach. The radar-generated range-compressed, magnitude image is initially pre-filtered, exactly as performed in Section 2.2.2. The other inputs to the algorithm include GPS (prior to GPS-denial), an IMU, and a magnetometer. An extended Kalman filter (EKF) uses the IMU measurements to propagate the system state and the AGL, range, and magnetometer measurements to update the state estimate. As in Chapter 3, R-RANSAC is used to identify and track ground scatterers.

The algorithmic improvements developed in this paper are two-fold. First, a new track filtering technique is used to improve track selection, discarding tracks containing clutter and tracks from weak point scatterers. Second, a new R-RANSAC-based AGL estimating technique is used to provide an improved AGL estimate.

This section is organized as follows. Section 4.2.1 briefly discusses the scatterer identification, association, and tracking, while Section 4.2.2 describes the enhanced track filtering. Section 4.2.3 discusses AGL estimation with R-RANSAC.

4.2.1 Resolving Point Scatterers

Resolving strong point scatterers is relatively simple in the absence of noise and clutter, though when clutter is present, such as in real flight, identifying distinct point scatterers is non-trivial. The classical approach to target tracking divides the algorithm into three distinct segments: detection, association, and tracking.
4.2.1.1 Detection of Strong Point Scatterers

A variety of algorithms and techniques exist to remove noise and clutter from radar imagery. These algorithms and techniques typically culminate in the use of a threshold to identify sufficiently strong scatterers [112]. The detection scheme is identical to that described in [44] and involves initially removing a weighted local pixel average, followed by an image smoothing technique, culminating in a decimation step. A threshold is then applied to the resulting pre-filtered image. Any pixel with return power higher than the final threshold is considered a strong point scatterer. While all detection techniques result in some false alarms, the use of association and tracking steps provide additional filtering and robustness.

4.2.1.2 Associating Strong Point Scatterers

Associating measurements between sensor samples is an instance of the multiple target tracking (MTT) problem, and has been an area of research since the 1960s. Some of the most common approaches include global nearest neighbor (GNN) [125], joint probabilistic data association (JPDA) [126], multiple hypothesis tracking (MHT) [127, 128], and the probability hypothesis density (PHD) [129, 130]. While useful, each of the these approaches has drawbacks. GNN is very poor in noisy environments. JPDA has no track management, while MHT is computationally expensive, and PHD has poor continuity in noisy environments.
The recently developed Recursive-RANSAC (R-RANSAC) algorithm resolves many of the difficulties with more traditional MTT algorithms, while also providing the modularity to add custom data association or tracking schemes. Specifically, R-RANSAC seeks to identify parameters with a large number of supporting inliers in the data. When new measurements are acquired, such as after a new chirp, R-RANSAC assesses each detected scatterer. If the detection is an inlier to a current parameter model, the parameter model is updated with the additional inlier. When a detection is not an inlier to any current parameter model, R-RANSAC initiates a RANSAC run to determine if a new parameter model should be initiated. For a more thorough description of R-RANSAC, see [122].

4.2.1.3 Tracking Point Scatterers

In addition to providing data association, R-RANSAC is used to track the scatterers (separate from the EKF-generated system state). As each new chirp arrives, R-RANSAC updates each track with the new inliers, spawning and killing tracks as needed. Using the PDA weighting algorithm, R-RANSAC provides a measurement to the EKF for each weighted detection. For synchronization purposes, R-RANSAC additionally informs the EKF whether to birth or kill tracks based on whether a maximum and minimum inlier ratio is achieved. The tracking implemented in this chapter mimics that in Section [2].

4.2.2 Scatterer Track Filtering

While the R-RANSAC-based radar odometry approach improves the GPS-denied navigational accuracy (see Section 3.3), the real results demonstrate a large disparity between the worst and best Monte-Carlo results, suggesting inadequacies in the implemented tracking algorithm. Accurately discerning between scatterers in the presence of clutter involves detecting persistent scatterers, which requires consideration over a wide window of time. At the same time, better navigation estimates are achieve when maximizing the number measurements for each scatterer. The balancing of these two attributes is very difficult as increasing the threshold to identify a scatterer decreases the remaining number of measurements that can be used to estimate the scatterers position.
For SAR systems, the narrow window of time where scatterers are visible is particularly difficult as the antennas often have a relatively narrow antenna beamwidth (the antenna used in this dissertation has roughly 20-30 degrees beamwidth in azimuth). Additionally, small UAVs typically fly at low elevations, even further decreasing the window of time over which point scatterers are in the antennas beamwidth. Figure 4.2.2a shows the scatterers tracked during a 10 second segment, prior to any additional filtering. As can be seen by the tracks, a majority of the scatterers are only detectable for a few seconds, and some that are only present for a few measurements.

This chapter enhances the scatterer tracking in Chapter 3 using two specific techniques. The first maximizes the window over which scatterers are present, while the second removes tracks that are likely clutter. Without the post-tracking filter, the tracker alone must delicately balance the desire to insert only good data into the system state while also maximizing the number of scatterer measurements. Adding the additional filtering stage allows for the tracker to identify more tracks, with a smaller number of inlier support, as the track filter can provide more specific filtering to remove the unnecessary tracks. Further, as the decreased urgency to expel data, running RANSAC on new outlier detections can be performed less frequently.

### 4.2.2.1 Increasing Scatterer Measurements

Figure 3.1.5b shows the simulated number of chirps required before a range progression curve was identified as a track [2]. While some of that error may be due to measurement error alone, by the time each scatterer is tracked, there are a significant number of previous measurements that have remained unused by the EKF. Accordingly, in this work all data is buffered for a fixed window of time (roughly 4 seconds) to allow for the first inliers from each track to be inserted at the time of first measurement. Figure 4.2.2b shows a scatterer track segment from Figure 4.2.2a with the augmented data. It is worth noting that increasing the number of measurements for identified track, increases the number of samples for both good and bad tracks, further increasing the need for additional track filtering.
Figure 4.2.2: (4.2.2a) shows all point scatterers track measurements after being labeled as a track using the R-RANSAC algorithm. (4.2.2b) shows the tracks, while adding the inlier measurements that occurred prior to track labeling.

4.2.2.2 Identifying and Removing Clutter

Clutter often presents itself in two different forms. Some clutter is present for very short periods of time. To mitigate this form of clutter, tracks containing a small number of measurements are simply discarded.

Other clutter is comprised of distributed targets, which is identifiable as it results in inconsistent range response across varying angles. The range progression of this clutter is much more likely to change arbitrarily from chirp to chirp, as shown by the blue track visible between 38 and
Figure 4.2.3: The results of the track filtering, plotting all measurements and tracks used by the EKF

41 seconds and between ranges 560 and 570 in Figure 4.2.2b. When track measurements vary significantly from chirp to chirp, they are identified as clutter and discarded. Figure 4.2.3 shows the resulting scatterer tracks. As can be seen, when compared to Figure 4.2.2a, the strong scatterers are present for a longer time window and shorter and noisier tracks have been removed.

4.2.3 AGL estimation with R-RANSAC

When using a radar aboard an aircraft, nadir is defined as the point at which the ground is closest to the aircraft and is identifiable as the range at which the return power sharply increases, as is visible by the sharp increase in return power in Figure 4.2.4a. While nominally true, Figure 4.2.4b shows that when point scatterers are located above the ground, such as in the presence of buildings or trees, radar return can occur at ranges closer than nadir. When this occurs, the thresholding scheme developed in Section 2.2.1 is unable to accurately identify these local, temporary changes in range, resulting in erroneous AGL measurements.

Similar to the R-RANSAC tracking of scatterer range progression discussed in Chapter 3, in this chapter R-RANSAC is used to track the range progression of the nadir return. A threshold is initially used to identify a window of ranges where nadir is likely to exist. Within that window, R-RANSAC is used to identify changes in range that corresponds to a change in the aircraft’s AGL. While the framework is identical to scatterer tracking a few modifications are implemented:
The best track is the track with the most number of inliers. This applies to resolving similar tracks and selecting which track to use as the AGL estimate.

A very small number of possible tracks are used.

Only linear tracks are considered.

When a track change results in a large, instantaneous change in AGL measurement, the transition is slightly smoothed.
A comparison of the thresholded AGL and the R-RANSAC AGL estimates is shown in Figure 4.2.5. The previously used threshold error is both higher and significantly less consistent, while the R-RANSAC AGL estimation error is both lower and more consistent, both of which improve the accuracy of GPS-denied navigation.

4.3 EKF and the Applied State-Space Model

A direct-form discrete measurement is used with a continuous dynamics EKF [118], with the system model

\[
\dot{x} = f(x, u + \mu) + \xi \tag{4.3.1}
\]

\[
y_k = h(x_k) + v_k, \tag{4.3.2}
\]

where \(y_k\) are discrete measurements at time \(k\), \(\mu\) is zero mean Gaussian system noise with covariance \(M\), \(\xi\) is zero mean Gaussian process noise with covariance \(Q\), and \(v_k\) is zero mean Gaussian measurement noise with covariance \(R\). The implemented EKF is defined as
Prediction Step:

\[
\dot{\hat{x}} = f(\hat{x}, u),
\]
\[
\dot{P} = \frac{\partial f}{\partial x} P + P \frac{\partial f^T}{\partial x} + Q + \frac{\partial f}{\partial u} M \frac{\partial f^T}{\partial u}.
\]

Measurement Step:

\[
L = P \frac{\partial h}{\partial x} \left( R + P \frac{\partial h}{\partial x} P \frac{\partial h^T}{\partial x} \right)^{-1},
\]
\[
P = \left( I - \frac{\partial h}{\partial x} \right) P^r,
\]
\[
\hat{x} = \hat{x}^r = L (y_k - h(\hat{x})).
\]

4.3.1 System State

The system state is represented as a combination of the aircraft state, \(x_a\), and the state of each scatterer, \(x_i\), resulting in

\[
x = \begin{bmatrix}
x_a \\
x_1 \\
\vdots \\
x_N
\end{bmatrix},
\]

where \(N\) scatterers are tracked.

4.3.1.1 Aircraft State

The aircraft state is

\[
x_a = \begin{bmatrix}
p_a^T \\
p_a^T \\
\Theta_a^T
\end{bmatrix}^T,
\]

with the aircraft position,

\[
p_a = \begin{bmatrix}
n_a \\
e_a \\
d_a
\end{bmatrix}^T.
\]
represented in Cartesian north, \( n_a \), east, \( e_a \), and down, \( d_a \) coordinates, and attitude

\[
\Theta_a = \begin{bmatrix} \phi_a & \theta_a & \psi_a \end{bmatrix}^\top,
\]

where \( \phi_a \), \( \theta_a \), \( \psi_a \) are the roll, pitch, and yaw, measured relative to the vehicle-2, vehicle-1, and vehicle frame respectively.

### 4.3.1.2 Point Scatterer State

Given the azimuth uncertainty [124], each tracked point scatter is represented using polar coordinates

\[
x_i = \begin{bmatrix} \sqrt{(n_i - n_a)^2 + (e_i - e_a)^2} \\ \arctan \left( \frac{e_i - e_a}{n_i - n_a} \right) \end{bmatrix} = \begin{bmatrix} r_i^g \\ \psi_i \end{bmatrix},
\]

where \( r_i^g \) is the ground-range from the aircraft to the scatterer and \( \psi_i \) is the body angle from the aircraft to the scatter, as shown in Figure 4.3.1. The Cartesian coordinates of the scatterer may be represented as

\[
p_i = \begin{bmatrix} n_i \\ e_i \\ 0 \end{bmatrix} = p_a + \begin{bmatrix} r_i^g \cos \psi_i \\ r_i^g \sin \psi_i \\ -h_a \end{bmatrix}.
\]

### 4.3.2 Prediction Model

The IMU is used for the prediction model input

\[
u = \begin{bmatrix} (a^b)^\top \\ \omega^\top \end{bmatrix}^\top,
\]

where \( a^b \) is the accelerometer measurements and \( \omega \) are the gyros measurements of the angular body rates.
The prediction model may be represented as

\[
\dot{x} = F(x, u) = \begin{bmatrix}
F_a(x, u) \\
F_1(x, u) \\
\cdots \\
F_N(x, u)
\end{bmatrix},
\tag{4.3.15}
\]

where \(F_a\) is the aircraft’s prediction model and \(F_n\) is the \(n^{th}\) scatterer’s prediction model.

**4.3.2.1 Aircraft Dynamics**

The aircraft dynamics are defined as

\[
F_a(x, u) = \begin{bmatrix}
v \\
g^i + R^b_i(\phi, \theta, \psi)a^b \\
S(\Theta)\omega
\end{bmatrix},
\tag{4.3.16}
\]
where \( \mathbf{g}^i \) is the gravity vector in the inertial frame, \( R^b_i (\phi, \theta, \psi) \) is the transformation from the body frame to the inertial frame, and \( S(\Theta) \) is

\[
S(\Theta) = \begin{bmatrix}
1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\
0 & \cos \phi & -\sin \phi \\
0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta}
\end{bmatrix}.
\] (4.3.17)

The Jacobian of the aircraft dynamics are

\[
\frac{\partial \mathbf{F}_a}{\partial \mathbf{x}_a}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix}
0^{3x3} & \mathbf{I}^{3x3} & 0^{3x3} \\
0^{3x3} & 0^{3x3} & \frac{\partial R^b_i(\Theta)a^b}{\partial \Theta} \\
0^{3x3} & 0^{3x3} & \frac{\partial S(\Theta)\mathbf{o}}{\partial \Theta}
\end{bmatrix},
\] (4.3.18)

where

\[
\frac{\partial R^b_i(\Theta)a^b}{\partial \Theta} = a_x \begin{bmatrix}
0 & -s_\theta c_\psi & -c_\theta s_\psi \\
0 & -s_\theta s_\psi & c_\theta c_\psi \\
0 & -c_\theta & 0
\end{bmatrix} + a_y \begin{bmatrix}
c_\phi s_\theta c_\psi + s_\phi s_\psi & s_\phi c_\theta c_\psi & -s_\phi s_\theta s_\psi - c_\phi c_\psi \\
c_\phi s_\theta s_\psi - s_\phi c_\psi & s_\phi c_\theta s_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi \\
-c_\phi c_\theta & -s_\phi s_\theta & 0
\end{bmatrix} + a_z \begin{bmatrix}
-s_\phi s_\theta c_\psi + c_\phi s_\psi & c_\phi c_\theta c_\psi & -c_\phi s_\theta s_\psi + s_\phi c_\psi \\
-s_\phi s_\theta s_\psi - c_\phi c_\psi & c_\phi c_\theta s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\
-s_\phi c_\theta & -c_\phi s_\theta & 0
\end{bmatrix}
\] (4.3.19)

and

\[
\frac{\partial S(\Theta)\mathbf{o}}{\partial \Theta} = \begin{bmatrix}
q c_\phi t_\theta - r s_\phi t_\theta & q \frac{s_\phi}{c_\theta} + r \frac{c_\phi}{c_\theta} & 0 \\
-q s_\phi - r c_\phi & 0 & 0 \\
q \frac{c_\phi}{c_\theta} - r \frac{s_\phi}{c_\theta} & q \frac{s_\phi}{c_\theta} + r \frac{c_\phi}{c_\theta} & 0
\end{bmatrix},
\] (4.3.20)

with \( c_\alpha \triangleq \cos(\alpha) \), \( s_\alpha \triangleq \sin(\alpha) \), and \( t_\alpha \triangleq \tan(\alpha) \).
4.3.2.2 Scatterer State Dynamics

The scatterer state propagation model is

\[ F_i(x, u_i) = \dot{x}_i = \begin{bmatrix} -\dot{n}_a \cos \psi_i - \dot{e}_a \sin \psi_i \\ \frac{1}{r_i}(\dot{n}_a \sin \psi_i - \dot{e}_a \cos \psi_i) \end{bmatrix}, \] (4.3.21)

where \( u_i = \begin{bmatrix} n_a \\ e_a \end{bmatrix}^T \).

The Jacobian of the scatterer states with respect to the aircraft state is

\[ \frac{\partial}{\partial x_a} F_i(x, u) = \begin{bmatrix} 0_3 \times 3 & -\cos \psi_i & \frac{1}{r_i} \sin \psi_i & 0_1 \times 4 \\ 0_3 \times 3 & -\sin \psi_i & -\frac{1}{r_i} \cos \psi_i & 0_1 \times 4 \end{bmatrix}. \] (4.3.22)

With respect to the scatterer state \( x_j \), the scatterer state \( x_i \) Jacobian is

\[ \frac{\partial}{\partial x_j} F_i(x, u) = \begin{cases} \begin{bmatrix} 0 & \dot{n}_a \sin \psi_i - \dot{e}_a \cos \psi_i \\ (r_i^g)^{-2}(-\dot{n}_a \sin \psi_i + \dot{e}_a \cos \psi_i) & (r_i^g)^{-1}(\dot{n}_a \cos \psi_i + \dot{e}_a \sin \psi_i) \end{bmatrix} & i = j \\ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} & i \neq j \end{cases} \] (4.3.23)

4.3.2.3 IMU Dynamics

The Jacobian of the aircraft dynamics and the scatter states with respect to the input is

\[ \frac{\partial}{\partial u} F_a(x, u) = \begin{bmatrix} 0_3 \times 3 & 0_3 \times 3 \\ R_i^b(\Omega) & 0_3 \times 3 \\ 0_3 \times 3 & R_i^b(\Theta) \end{bmatrix}. \] (4.3.24)
and
\[
\frac{\partial}{\partial u} F_i(x, u) = 0_{2 \times 1},
\]
respectively.

### 4.3.3 Update Sensor Dynamics

The update sensor measurement model is given by
\[
y_k = H(x_k) + \eta_k,
\]
where
\[
H(x) = \begin{bmatrix}
h_{GPS} \\
h_{mag} \\
h_{turn} \\
h_{AGL} \\
h_f \\
h_{\Delta}
\end{bmatrix}.
\]
(4.3.27)

The corresponding update measurement Jacobian with respect to the system state is
\[
\frac{\partial H}{\partial x} = \begin{bmatrix}
\frac{\partial h_{GPS}}{\partial x_a} & \frac{\partial h_{GPS}}{\partial x_s} \\
\frac{\partial h_{mag}}{\partial x_a} & \frac{\partial h_{mag}}{\partial x_s} \\
\frac{\partial h_{turn}}{\partial x_a} & \frac{\partial h_{turn}}{\partial x_s} \\
\frac{\partial h_{AGL}}{\partial x_a} & \frac{\partial h_{AGL}}{\partial x_s} \\
\frac{\partial h_f}{\partial x_a} & \frac{\partial h_f}{\partial x_s} \\
\frac{\partial h_\Delta}{\partial x_a} & \frac{\partial h_\Delta}{\partial x_s}
\end{bmatrix}.
\]
(4.3.28)

#### 4.3.3.1 GPS

GPS measures
\[
y_{GPS} = \begin{bmatrix}
n_a \\
e_a \\
-d_a \\
V_g \\
x_a
\end{bmatrix}^\top,
\]
(4.3.29)
where $n_a$, $e_a$, and $d_a$ are the Cartesian north, east, and up positions,

$$V_g = \sqrt{n_a^2 + e_a^2} \quad (4.3.30)$$

is the aircraft’s ground velocity and

$$\chi_a = \text{atan}\left(\frac{\dot{e}_a}{\dot{n}_a}\right) \quad (4.3.31)$$

is the aircraft heading.

The Jacobian of the GPS measurement, with respect to the system state is

$$\frac{\partial h_{GPS}}{\partial x_a} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{\dot{n}_a}{V_g} & \frac{\dot{e}_a}{V_g} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -\frac{\dot{e}_a}{V_g} & \frac{\dot{n}_a}{V_g} & 0 & 0 & 0 & 0 \\
\end{bmatrix}, \quad (4.3.32)$$

$$\frac{\partial h_{GPS}}{\partial x_i} = 0_{5\times2}. \quad (4.3.33)$$

### 4.3.3.2 Magnetometer

The digital compass measures the bearing,

$$y_{\text{mag}} = \psi, \quad (4.3.34)$$

while its Jacobian with respect to the system state is

$$\frac{\partial h_{\text{mag}}}{\partial x_a} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.3.35)$$

$$\frac{\partial h_{\text{mag}}}{\partial x_i} = \begin{bmatrix} 0 & 0 \end{bmatrix}. \quad (4.3.36)$$
4.3.3.3 Coordinated Turn

To make roll observable, a standard coordinated pseudo-measurement is used,

\[ y_{\text{turn}} = \frac{\sin \phi}{\cos \theta} q + \frac{\cos \phi}{\cos \theta} r - \frac{g}{\sqrt{n_a^2 + \dot{e}_a^2}} \tan \phi. \]  

(4.3.37)

The Jacobian of the coordinated turn with respect to the system state is

\[ \frac{\partial h_{\text{turn}}}{\partial x_a} = \begin{bmatrix} 0 & 0 & 0 & H_1(x) & H_2(x) & 0 & H_3(x) & H_4(x) & 0 \end{bmatrix}, \]  

(4.3.38)

\[ \frac{\partial h_{\text{turn}}}{\partial x_i} = \begin{bmatrix} 0 & 0 \end{bmatrix}, \]  

(4.3.39)

where

\[ H_1(x) = \frac{g n_a \tan \phi}{V_g^3}, \]  

(4.3.40)

\[ H_2(x) = \frac{g \dot{e}_a \tan \phi}{V_g^3}, \]  

(4.3.41)

\[ H_3(x) = \frac{\cos \phi}{\cos \theta} q - \frac{\sin \phi}{\cos \theta} r - \frac{g \sec^2 \phi}{V_g}, \]  

(4.3.42)

\[ H_4(x) = \left( \frac{\sin \phi}{\cos \theta} q + \frac{\cos \phi}{\cos \theta} r \right) \tan \theta. \]  

(4.3.43)

4.3.3.4 AGL Measurement

The AGL measurement (as described in Section 4.2.3) measures the aircraft’s elevation above the ground, and given our flat earth assumption is

\[ y_{\text{AGL}} = -h_a. \]  

(4.3.44)

The Jacobians of the AGL with respect to the aircraft state and scatterer states is

\[ \frac{\partial h_{\text{AGL}}}{\partial x_a} = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \]  

(4.3.45)

\[ \frac{\partial h_{\text{AGL}}}{\partial x_i} = \begin{bmatrix} 0 & 0 \end{bmatrix}. \]  

(4.3.46)
4.3.3.5 Scatterer Range Measurements

The scatterer range measurements consist of

\[ y_N = \begin{bmatrix} r_1 & \ldots & r_N \end{bmatrix}^\top, \]  

where each scatterer range measurement is

\[ r_i = \sqrt{(r_i^g)^2 + d_a^2}. \]

The Jacobian of the range measurement of the scatterer relative to the aircraft is

\[ \frac{\partial}{\partial x_a} h_i(x) = \frac{1}{r_i} \begin{bmatrix} 0 & 0 & d_a & 0_{1 \times 6} \end{bmatrix}^\top, \]

while the Jacobian relative state \( x_i \) with respect to the scatterer state \( x_j \) is

\[ \frac{\partial}{\partial x_j} h_i(x) = \begin{cases} \frac{1}{r_i} \begin{bmatrix} r_i^g & 0 \end{bmatrix}^\top & i = j \\ \begin{bmatrix} 0 & 0 \end{bmatrix}^\top & i \neq j \end{cases}. \]

4.3.3.6 Relative Range Measurement

While the relative positioning of each scatter is random and unknown, the relative range is measured when scatterers are both measured during the same chirp. This may be represented

\[ r_{ij} = r_i - r_j, \]

with the resulting relative range measurements

\[ y_\Delta = \begin{bmatrix} r_{12} & \ldots & r_{(N-1)N} \end{bmatrix}^\top. \]
The Jacobian of the relative range with respect to the aircraft state is

\[
\frac{\partial}{\partial x_a} h_{ij}(x) = \begin{bmatrix} 0_{1 \times 2} & \left( r_i^{-1} - r_j^{-1} \right) h_a & 0_{1 \times 6} \end{bmatrix}^\top,
\]

while the Jacobian of the relative range \( h_{ij} \) with respect to the scatterer states is

\[
\frac{\partial}{\partial x_k} h_{ij}(x) = \begin{cases} 
\begin{bmatrix} r_i^{-1} r_i^g - r_j^{-1} r_j^g & 0 \end{bmatrix}^\top & k = i \neq j \\
- r_i^{-1} r_i^g + r_j^{-1} r_j^g & 0 & k = j \neq i \\
0 & 0 & m \neq i \neq j \text{ and } m \neq j 
\end{cases}.
\]

### 4.3.3.7 Range Rate

The R-RANSAC algorithm performs a polynomial fit to the range progression, and as such is able to estimate the range rate \( \dot{r} \) to each scatterer. This estimate is only used to calculate the initial azimuth angle for the point scatterer, in attempt to decrease the time necessary to estimate the scatterer’s position.

The range rate may be calculated as

\[
\dot{r}_i = \dot{n}_a \cos \psi_i \cos \phi_i + \dot{e}_a \sin \psi_i \cos \phi_i + \dot{h}_a \sin \phi_i,
\]

where, assuming a flat earth model, the scatterer elevation angle may be represented as

\[
\phi_i(t) = \arcsin \left( -\frac{h_a}{r_i} \right).
\]

Using the ground velocity from Equation 4.3.30, and by defining

\[
\begin{bmatrix} \dot{n}_a \\ \dot{e}_a \end{bmatrix} = \begin{bmatrix} V_g \cos \psi_a \\ V_g \sin \psi_a \end{bmatrix},
\]

94
the range rate equation may be modified to calculate an azimuth estimate

\[ \psi_i = \acos \left( \frac{\dot{r}_i - h_a \sin \phi_i}{\cos \phi_i \sqrt{n^2_a + \dot{e}^2_a}} \right) + \psi_a. \]  \hspace{1cm} (4.3.58)

**New Scatterer State Initialization**  When a new scatterer \( i \) is first identified, its state is initialized as

\[ \mathbf{x}_i = \begin{bmatrix} r^g_i \\ \psi_i \end{bmatrix}^T, \]  \hspace{1cm} (4.3.59)

where the azimuth angle is initialized, using Equation 4.3.58 as

\[ \psi_i = \acos \left( \frac{\dot{r}_i + h_a (t^-) h_a (t^-)}{r^g_i \sqrt{n^2_a (t^-) + \dot{e}^2_a (t^-)}} \right) + \psi_a (t^-), \]  \hspace{1cm} (4.3.60)

where \( t^- \) indicates the current state estimate.

Additionally, the new scatterer EKF covariance terms must be updated. The covariance is set to

\[ \text{cov} \left[ \mathbf{x}_i \mathbf{x}_i^T \right] = \begin{bmatrix} \sigma_r & 0 \\ 0 & \sigma_\psi \end{bmatrix}, \]  \hspace{1cm} (4.3.61)

with the range and azimuth angle are \( \sigma_r^2 \) and \( \sigma_\psi^2 \) respectively. The cross correlation terms \( \text{cov} \left[ \mathbf{x}_i \mathbf{x}_j^T \right] \) and \( \text{cov} \left[ \mathbf{x}_i \mathbf{x}_j^T \right] \), when \( i \neq j \), are initialized as \( \mathbf{0}_{2 \times 9} \) and \( \mathbf{0}_{2 \times 2} \) respectively. while the cross correlation term \( \text{cov} \left[ \mathbf{x}_a \mathbf{x}_j^T \right] = \mathbf{0}_{2 \times 9} \).

### 4.4 Results

It is very difficult to emulate the return from a real radar operating in a cluttered environments. This is particularly true when non-ideal point scatterers are used. Unfortunately, access to real flight data is limited, and as such some testing must be performed with simulated data.

#### 4.4.1 Simulated Results

For the simulations presented in this chapter, significant care is taken to ensure that the results behave similarly to that of real flight. For each point-scatter configuration, each point scatter is assigned a random two dimensional location, in addition to a random return power, and a random
Figure 4.4.1: Average drift error for simulated flight as the aircraft underwent varying commanded banking angles (each with 36 Monte Carlo runs).

range “width”, defining the return power and the range “width” over which the return is present. While previous simulations modeled the dynamic nature of real flight, to more accurately match the real flight results, the simulations in this section have increased the random aircraft motion (such as would be present with increased wind gusts). The simulated flights all use a simulated navigation-grade IMU (such as is used in the real flight).

The approach presented in this chapter makes no motion assumptions. Figure 4.4.1 compares the drift error as the aircraft navigates across varying banking angles, with each banking angle representing the average error over 36 Monte Carlo simulations, consisting of 6 distinct point-scatterer configurations and 6 different R-RANSAC random seeds. For straight flight, the average drift rate is roughly 1.7%, while the drift rate for 5, 10, and 15 degrees turns (each across the 60 second flight, as the aircraft navigated roughly 3300 meters) is roughly 1.3%, 1.0%, and 1.3% respectively.

The previous results have only explored flights consisting of 60 seconds of GPS-denied navigation. Figure 4.4.2 shows the results of a 10 minute simulated GPS-denied flight, over which the aircraft flies roughly 33 km. During the, the simulated navigation-grade IMU drifted roughly 29.8 km (a 90% drift rate), while the radar odometry system drifted 1.66 km, a 5.0% drift rate. While this represents only a single simulation, it suggests that the drift rate of the radar odometry approach may drift significantly less over longer flights.
Figure 4.4.2: Simulated GPS-denied navigational error over a prolonged 10 minutes simulated flight. The blue, solid line is the IMU drift and the green, dotted line is the drift error of the radar odometry algorithm.

4.4.2 Real Results

Flight tests were performed using a SAR mounted to a Cessna aircraft. For observability, a simulated GPS was used for 16 seconds, after which the navigation-grade IMU was used in conjunction with a simulated magnetometer for 60 seconds, during which the aircraft traveled approximately 2400 meters. 10 Monte Carlo simulations were run, with Figure 4.4.3 comparing the IMU drift to the best-case, worst-case, and average radar odometry results. If the IMU alone had been used, the aircraft position estimate would have been 209 meters off, a drift rate of roughly 8.7%. Conversely, at the end of the 60 seconds, the drift with the enhanced radar odometry algorithm was on average 55 meters (or a 2.3% drift rate), with a worst-case drift of 95 meters (3.9% drift rate), and a best-case drift of 24 meters (1.0% drift rate).

A comparison of the real and simulated results is shown in Figure 4.4.4. The Figure shows that the real IMU drifts less than the simulated IMU in contrast to the similar real and simulated radar odometry results. As only a single real flight was available, the data collection method is slightly different for the real and simulated flights. The real radar odometry results are the average across ten R-RANSAC runs, from a single flight. In contrast, the simulated radar odometry results are the average across six R-RANSAC runs, over each of six simulated scatterer configurations.
Figure 4.4.3: Position estimation error from real flight with 10 Monte Carlo runs.

Figure 4.4.4: Comparison of real flight results (shown as a solid line) and simulated flight (shown as a dashed line). The IMU results are shown in bold, while the radar odometry results with IMU are shown with a narrow line.

(resulting in 36 total simulations). In the Figure, the average simulated results are much more smooth, though this may be due to averaging across a larger collection set.

The results compare favorably with the simulated results in this chapter and the real results from previous chapters. The greedy RO algorithm’s drift error 133 meters [1], while the RO with R-RANSAC results had an average drift rate of 92 meters. Additionally, previous R-RANSAC results demonstrated a large disparity (180 meters) between the best and worst-case simulations.
Table 4.1: Comparison of previous radar odometry approaches using the same data set. The unaided IMU, greedy RO algorithm [1], the RO with R-RANSAC algorithm [2], and the enhanced results presented in this chapter (all results are from real flight data). The greedy RO algorithm did not require a Monte Carlo run for their real results, as the Hough transform exhaustively considers all data, whereas both the RO with R-RANSAC and these results each ran 10 Monte Carlo runs.

<table>
<thead>
<tr>
<th></th>
<th>IMU Alone</th>
<th>Greedy RO</th>
<th>RO with R-RANSAC</th>
<th>Enhanced RO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best run final position error</td>
<td>209 m</td>
<td>135 m</td>
<td>49 m</td>
<td>24 m</td>
</tr>
<tr>
<td>Average final position error</td>
<td>209 m</td>
<td>135 m</td>
<td>92 m</td>
<td>55 m</td>
</tr>
<tr>
<td>Worst run final position error</td>
<td>209 m</td>
<td>135 m</td>
<td>165 m</td>
<td>94 m</td>
</tr>
<tr>
<td>Disparity at flight conclusion</td>
<td>N/A</td>
<td>N/A</td>
<td>117 m</td>
<td>70 m</td>
</tr>
<tr>
<td>Average disparity over flight</td>
<td>N/A</td>
<td>N/A</td>
<td>54 m</td>
<td>30 m</td>
</tr>
<tr>
<td>Worst disparity over flight</td>
<td>N/A</td>
<td>N/A</td>
<td>198 m</td>
<td>72 m</td>
</tr>
</tbody>
</table>

[2]. A comparison between the three sets may be seen in Table 4.1. As shown in the table, these enhanced results were significantly better than previous approaches.

While data was only only available from this single real flight, the results suggest that radar odometry is a possible solution for accurate position estimation in the presence of intermittent GPS. Other techniques, such as Visual Odometry, exist and demonstrate promising results, but in poor visibility, or at high altitudes, the radar odometry approach in this paper may be a valuable alternative.

4.4.3 Conclusion

In this chapter the validity of the radar odometry approach to GPS-denied navigation is considered. Observability analysis is performed and an approach is proposed that guarantees system observability. Further, a track filtering stage is added to the improve clutter suppression. A more robust, AGL estimation technique using R-RANSAC is presented. Using simulated flight, the robustness of the algorithm is demonstrated as the aircraft performs various banking maneuvers in addition to a longer flight time. Real flight results further demonstrate the accuracy of the solution resulting in an average drift rate of 2.3% when navigating a real, cluttered environment.
Chapter 5

Conclusion

5.1 Summary

For true autonomous navigation, GPS-denied navigation is essential. Vision-based navigation solutions have been used for accurate navigation, though at long stand-off ranges or in inclimate weather their capability is limited. In this we use radar to measure the aircraft’s odometry during GPS-denied navigation.

In Chapter 2, we present a greedy approach to radar odometry, which selects pairs of scatterers and uses their relative range progression to estimate the motion of the aircraft. We also use the radar range return to detect the altitude of the aircraft. The Hough transform performs data association, with GNN performing the tracking. The greedy approach relies on a piece-wise constant heading assumption and as such, the simulated results indicate that as the aircraft increases its banking angle, the results deteriorate. Though limited due to its assumptions, the approach reduces the real flight drift from 8.7% with the IMU alone to roughly 5.6% with greedy radar odometry.

Chapter 3 discusses an alternate approach, using R-RANSAC to detect and track scatterers. Rather than rely on pairs of scatterers, this approach uses the range progression of all identified point scatterers to estimate the motion of the aircraft. The simulated results in this section decrease the drift rate to less than 1%, though the real results improved the GPS-denied drift rate to an average 3.8%.

Chapter 4 explores the validity of the radar odometry approach. It performs observability analysis of the radar odometry algorithm, proposing a modality that guarantees observability. A more robust AGL estimation technique is presented, using R-RANSAC to track and minimize deviations caused by scatterers located above the ground. This chapter also adds significant robustness and consistency by discarding previously tracked clutter. Simulations are used to explore the accuracy of a long flight and to test the accuracy over a variety of banking angles. The real
results presented demonstrate an average drift rate of 55 meters over the 2400 meter flight, an approximately 2.3% drift rate as compared to the IMU-only results which drifted 209 meters, a 8.7% drift rate. Table 4.1 provides a more complete comparison of the results from each Chapter.

The real results in this dissertation demonstrate that radar odometry can improve the navigational accuracy of a GPS-denied system. While effective in the data available, the results in this paper do not prove its accuracy for all environments. Additional work is also necessary to validate the robustness of the solution in variety of environments.

5.2 Future Work

The are a number of viable avenues for future research with radar odometry and GPS-denied navigation. Listed here are some of the possible extensions.

5.2.1 Remove system constraints

The previously mentioned assumptions are designed for low SWaP systems, though removing some of the assumptions may provide better accuracy with a marginal increase in SWaP. Specifically we assume:

1. The radar is not coherent. While this assumption provides robustness, allowing for incoherent systems, allowing for a coherent processing techniques, such as the use doppler, should increase the system’s accuracy as the range rate would be a radar measurement.

2. The range-compressed images are magnitude only. With high precision radar, complex imagery is typically used for more accurate measurements of individual strong scatterers. Specifically utilizing each scatterer’s phase progression provides insight into the presence of noise, while also providing sub-pixel range accuracy.

3. Nadir is present in the range-compressed image. We use nadir to measure the aircraft’s AGL, which is used for motion detection. An alternate approach is to ignore the aircraft’s AGL entirely and use a second receive aperture to directly measure the elevation angle to each scatterer. For this approach more processing would be necessary, but the processing could be performed on a DSP or FPGA for real-time operation.
4. Only a single receive aperture is used in any of the simulations or flights. Using two or three antennas (particularly with a coherent radar) allows for the azimuth and/or elevation angle to each scatterer to be measured directly.

Removing the above assumptions should increase the system's accuracy and robustness with a possible increase in SWaP, though to many platforms the increases in size or power may be negligible.

5.2.2 Explore the Use of Traditional Radar Algorithms

A variety of traditional radar processing algorithms exist that may be leveraged for increased accuracy or improved robustness. Two approaches are of particular note.

5.2.2.1 Ground Motion Tracking Indicator (GMTI) Algorithms

GMTI algorithms use a coherent radar with multiple receive antennas to estimate the motion of ground targets. While the intent of these algorithms is very different from radar odometry, there is value in leveraging this mature field. One of the first steps in GMTI algorithms is to use a space time adaptive programming (STAP) technique to characterize the range progression of stationary point scatterers (an excellent GMTI reference is [73]). Once identified, GMTI seeks to remove the stationary scatterers from the measured data. Radar odometry attempts the inverse of GMTI and STAP. Rather than removing the stationary scatterers, RO seeks to use them to estimate the aircraft’s motion. Using STAP techniques to both identify scatterers and to estimate the aircraft’s motion may allow for improved navigation.

5.2.2.2 SAR Image Formation Algorithms

SAR image formation algorithms rely on a coherent radar to form very precise ground images. For proper compression, these techniques also rely on highly accurate aircraft position estimation, while also utilizing a digital elevation map (DEM) to properly estimate their range to scatterers. Their use of GPS and DEMs, and their required processing requirements, has put them outside the scope of the focus of this work, though their motion compensation techniques [93, 147, 148] could be leveraged to assist in GPS-denied navigation.
5.2.3 Mapping

SLAM algorithms typically combine odometric approaches with mapping techniques to allow for better navigation. While many SLAM approaches have used radar beacons, terrain aided navigation, and a narrow-beam rotating antenna for SLAM (see Section 1.2.6 for more detail) the use of a single, fixed radar antenna with SLAM has yet to be explored.

The difficulty with a single aperture radar for localization and mapping lies in the ability to create accurate feature descriptors. Some work has been performed to use individual point scatterers for feature description [149, 150], while other approaches use contour feature descriptors [151]. The use of either approach relies on image formation, and as such, work is necessary to relate the return from individual chirps to the image-based features.

5.2.4 Real Flight Testing Using Known and Unknown Scatterers

Our reliance on unknown features for GPS-denied navigation has been relatively difficult. While the use of environmental point scatterers for navigation is the end goal, significant value could be gained by performing testing using known, geo-located corner reflectors and/or point scatterers. Such testing could provide a baseline to properly consider the accuracy of our approach.

In addition to calibrated testing, there is a need to test radar odometry in a variety of environments. Specifically, changes in moisture, terrain, and features must be tested prior to considering radar odometry a GPS-denied solution.

5.2.5 Estimate Pitch and Roll from the Radar Return

Further analysis of the range progression hyperbola could provide some attitude measurements. Specifically the beam pattern of the antenna and the presence of scatterers has been shown to provide rough attitude estimates of aircraft [152]. Building off of these techniques, a radar-based navigation solution could even further mitigate the gyro drift.

5.3 Discussion

When considering autonomous navigation, there are a wide variety of solutions that have been and are being explored. The accuracy and reliability of these systems is increasing at a
dramatic pace. However, modern systems, particularly airborne systems, are currently unable to autonomously navigate in GPS-denied, unknown environments in poor weather, or even at night. Many systems can operate well in controlled or constrained environments but, in the presence of rain, at night, or in changing environments, their limitations are shown. We believe that radar may change that. While radar has limitations, it is able to operate in environments where other sensors cannot.

Radars have been around for decades, but the cost, size, availability, and complexity of these systems have rendered them relatively unused by all but the radar community. Throughout our research, there have been multiple discussions regarding sensor selection where researchers have readily admitted that radar would better meet their needs, but due to cost and availability, it was an intractable solution. Further limiting in the use of radar is its complexity. While vision sensors are very intuitive to users, the operation and use of radar is not.

Fortunately, over the last decade, the SWaP of radar systems has dramatically decreased. Commercially available software-defined radios are now available for average consumers. While the quality and accuracy of these radios is still limited, their presence suggests that higher-quality inexpensive solutions may soon follow.

For radar odometry to become a GPS-denied solution it must be a cheaper more readily available solution. Additionally, it must be tested across a wide variety of environments. While changes in foliage, moisture, and scenery do not impede the radar’s operation, the approach must be more thoroughly validated in hardware.

In this dissertation we explore the use of a low-SWaP radar for navigation. We show that when GPS is lost, radar odometry has improved the navigational accuracy of an IMU alone. While a variety of other approaches and sensors may provide similar results, we believe that radar’s innate ability to operate in inclimate weather render it a strong candidate for future autonomous navigation solutions.
Bibliography


106


113


115