1 Abstract

CONTACT is a three-year graduate project being conducted at the University of Colorado Boulder (CU Boulder) to educate students and advance algorithms/methods for small satellite timing systems. The primary objective of CONTACT is to design, build, and operate a testbed to facilitate development of low size, weight, and power (SWaP) approaches for small satellite timing systems. A secondary objective is to develop and conduct a flight experiment onboard the MAXWELL mission to characterize the on-orbit performance of a chip scale atomic clock (CSAC). CSAC-class clocks are expected to be a key element of low SWaP timing systems.

The testbed supports evaluation of concepts for timing systems that include the following functionality:

- Combine/ensemble multiple low SWaP clocks to create a stable, robust, onboard time scale for timing of events and observations
- Detect and isolate/correct clock frequency and phase faults
- Create a steered clock realization of the time scale to enable onboard generation of communication or navigation signals

The testbed is built around a software defined radio (SDR) architecture which performs clock measurements, clock characterization, clock ensembling, fault detection/isolation, and steered signal generation.

The opportunity to conduct a flight experiment to characterize CSAC performance on-orbit has been made possible by the UNP-9 MAXWELL mission; which is being developed concurrently as a graduate project at CU Boulder. The CONTACT team has primary responsibility for defining and preparing for the CSAC experiment, working closely with the MAXWELL team to develop software for commanding and recording telemetry data from the onboard GPS receiver that will measure the CSAC clock performance, and testing the functionality of both the CSAC and GPS receiver in the desired operational configuration in space.

This semester report provides a general overview of the CONTACT project and details the work completed in Spring 2021. Highlights of recent progress include:

- Individually tested three CSACs using GPS receiver with roof antenna drop to compare the performance of each clock
- Conducted thermal testing experiments using CSAC as external reference to GPS receiver to better understand the relationship between the CSAC performance and temperature variations

- Conducted electromagnetic field testing using CSAC as external reference to GPS receiver to characterize CSAC performance in an increased magnetic field

- Made CSAC phase measurements using Rubidium as reference signal input to assess the use of the N310 as a measurement system

- Used N310 phase measurements of three CSACs to generate phase difference measurement inputs to a Kalman filter

- Conducted open-loop and closed-loop steering experiments to determine the capabilities of the Ettus N310 in making fine frequency adjustments
## Contents

1 Abstract ................................................................. 1

2 Introduction ............................................................ 8

3 Project Overview ......................................................... 9

   3.1 Project Description .................................................. 9

   3.2 Project Objective .................................................... 10

       3.2.1 Semester Objectives ......................................... 10

   3.3 Resources ........................................................... 11

       3.3.1 Roles and Responsibilities .................................. 11

       3.3.2 Other Resources ............................................. 12

   3.4 Schedule ............................................................ 13

   3.5 Budget ............................................................... 15

   3.6 Requirements ....................................................... 15

   3.7 ConOps ............................................................... 19

   3.8 Testbed Architecture ................................................ 20

       3.8.1 Measurement Subsystem ..................................... 20

       3.8.2 Characterization Subsystem .................................. 21

       3.8.3 Filter/Ensemble Subsystem .................................. 21

       3.8.4 Signal Generation Subsystem ................................. 21

4 Ettus N310 and GNU Radio ................................................. 22

5 Measurement Subsystem .................................................. 26

   5.1 Measurement Process ............................................... 26

   5.2 Measurement Experiments ......................................... 27

   5.3 Measurement System Characterization ............................... 27

6 Filter and Clock Ensemble Subsystem ................................... 30

   6.1 Clock Model ........................................................ 31

   6.2 Measurement Model ............................................... 31

   6.3 Filter Results ...................................................... 32

   6.4 Future Work ....................................................... 33

7 Signal Generation Subsystem ............................................. 33

   7.1 Signal Generation Architecture ................................... 33
<table>
<thead>
<tr>
<th>Contents</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1.1 Original Conceptual Design</td>
<td>33</td>
</tr>
<tr>
<td>7.1.2 Implementation using Ettus N310 and GNU Radio</td>
<td>36</td>
</tr>
<tr>
<td>7.2 Open-loop Steering Experiment</td>
<td>42</td>
</tr>
<tr>
<td>7.3 Closed-loop Steering Experiment</td>
<td>46</td>
</tr>
<tr>
<td>7.4 Future Work</td>
<td>56</td>
</tr>
<tr>
<td>8 CSAC Flight Experiment on MAXWELL CubeSat</td>
<td>57</td>
</tr>
<tr>
<td>8.1 Live Sky Testing</td>
<td>59</td>
</tr>
<tr>
<td>8.2 RF/GNSS Simulator Testing</td>
<td>60</td>
</tr>
<tr>
<td>8.2.1 Locked-Clock Testing</td>
<td>60</td>
</tr>
<tr>
<td>8.2.2 Flight Mode Testing</td>
<td>62</td>
</tr>
<tr>
<td>8.3 Thermal Testing</td>
<td>64</td>
</tr>
<tr>
<td>8.3.1 Constant Temperature Hot Test</td>
<td>64</td>
</tr>
<tr>
<td>8.3.2 Constant Temperature Cold Test</td>
<td>65</td>
</tr>
<tr>
<td>8.3.3 Orbital Temperature Profile Test</td>
<td>65</td>
</tr>
<tr>
<td>8.4 Electromagnetic Field Testing</td>
<td>66</td>
</tr>
<tr>
<td>8.5 Future Work</td>
<td>67</td>
</tr>
<tr>
<td>9 Acknowledgements</td>
<td>68</td>
</tr>
<tr>
<td>10 Appendix</td>
<td>71</td>
</tr>
<tr>
<td>10.1 CONTACT Team</td>
<td>71</td>
</tr>
<tr>
<td>10.2 Closed-Loop Steering Exp. Flowgraph</td>
<td>72</td>
</tr>
<tr>
<td>10.3 Clock Phase Subtraction</td>
<td>73</td>
</tr>
<tr>
<td>10.4 Gantt Chart</td>
<td>74</td>
</tr>
</tbody>
</table>
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Spring 2021 Organization Chart</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>Gantt Chart Spring 2021</td>
<td>13</td>
</tr>
<tr>
<td>3</td>
<td>Concept of Operations</td>
<td>19</td>
</tr>
<tr>
<td>4</td>
<td>Functional Block Diagram</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>Ettus N310 SDR[7]</td>
<td>22</td>
</tr>
<tr>
<td>6</td>
<td>Process for measuring clock phase</td>
<td>26</td>
</tr>
<tr>
<td>7</td>
<td>Measured clock bias (seconds) over time (hours) showing the common bias among all three clocks</td>
<td>27</td>
</tr>
<tr>
<td>8</td>
<td>Measured clock bias (seconds) with respect to a rubidium reference over time (hours)</td>
<td>27</td>
</tr>
<tr>
<td>9</td>
<td>Clock bias residuals (picoseconds) from differencing the phase growth of identical signals on the same daughterboard</td>
<td>28</td>
</tr>
<tr>
<td>10</td>
<td>Clock bias residuals (nanoseconds) from differencing the phase growth of identical signals on different daughterboards</td>
<td>28</td>
</tr>
<tr>
<td>11</td>
<td>CSAC &amp; Rb Time Domain Behavior</td>
<td>29</td>
</tr>
<tr>
<td>12</td>
<td>Clock bias after subtracting the Rb signal from each CSAC</td>
<td>30</td>
</tr>
<tr>
<td>13</td>
<td>Detrended clock bias (nanoseconds) of the CSAC signals measured against the same Rb signal</td>
<td>30</td>
</tr>
<tr>
<td>14</td>
<td>CSAC Stability Relative to a Rb Reference on the Same / Different Daughterboard</td>
<td>30</td>
</tr>
<tr>
<td>15</td>
<td>Overlapping Allan Deviation for CSACs Measurements &amp; CSAC State Estimates</td>
<td>33</td>
</tr>
<tr>
<td>16</td>
<td>Functional block diagram for the nominal signal generation subsystem design.</td>
<td>34</td>
</tr>
<tr>
<td>17</td>
<td>Block diagram for the high-rate steering system implementation.</td>
<td>35</td>
</tr>
<tr>
<td>18</td>
<td>Block diagram for the baseband steering system implementation.</td>
<td>35</td>
</tr>
<tr>
<td>19</td>
<td>Overlapping Allan Deviation result comparison between steering methods.</td>
<td>36</td>
</tr>
<tr>
<td>20</td>
<td>Message Passing Interface</td>
<td>39</td>
</tr>
<tr>
<td>21</td>
<td>Steering Interval - Generation and Reception of Steering Commands</td>
<td>40</td>
</tr>
<tr>
<td>22</td>
<td>Steering Interval Distribution - Generation and Reception of Steering Commands</td>
<td>41</td>
</tr>
<tr>
<td>23</td>
<td>GNU Radio Flowgraph - ZOH Control Input Blocks</td>
<td>42</td>
</tr>
<tr>
<td>24</td>
<td>Block Diagram Representation of Open-loop steering</td>
<td>43</td>
</tr>
<tr>
<td>25</td>
<td>Power Spectrum of Unsteered Signal</td>
<td>44</td>
</tr>
<tr>
<td>26</td>
<td>Power Spectrum of Steered Signal</td>
<td>44</td>
</tr>
<tr>
<td>27</td>
<td>Detrended Bias of Unsteered Signal</td>
<td>44</td>
</tr>
<tr>
<td>28</td>
<td>Detrended Bias of Steered Signal</td>
<td>44</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>29</td>
<td>Overlapping Allan Deviations for Unsteered and Steered signals</td>
<td>45</td>
</tr>
<tr>
<td>30</td>
<td>Power Spectral of Beat Frequency</td>
<td>46</td>
</tr>
<tr>
<td>31</td>
<td>Closed-Loop Steering Experiment Block Diagram</td>
<td>47</td>
</tr>
<tr>
<td>32</td>
<td>CSAC Detrended Bias and Fractional Frequency (w.r.t Rb)</td>
<td>49</td>
</tr>
<tr>
<td>33</td>
<td>Steered Signal Detrended Bias and Fractional Frequency (Rb Reference)</td>
<td>50</td>
</tr>
<tr>
<td>34</td>
<td>Steered Signal v CSAC - Fractional Frequency (Rb Reference) - N310</td>
<td>51</td>
</tr>
<tr>
<td>35</td>
<td>Steered Signal Output During Underrun</td>
<td>51</td>
</tr>
<tr>
<td>36</td>
<td>Steered Signal Detrended Bias and Fractional Frequency (Rb Reference) - N200</td>
<td>52</td>
</tr>
<tr>
<td>37</td>
<td>CSAC and Steered Signal Between Underruns (Rb Reference) - N310</td>
<td>53</td>
</tr>
<tr>
<td>38</td>
<td>CSAC and Steered Signal Between Underruns - N310</td>
<td>54</td>
</tr>
<tr>
<td>39</td>
<td>CSAC and Steered Signal ADEVs - N310</td>
<td>55</td>
</tr>
<tr>
<td>40</td>
<td>Block diagram of the flight configuration and live sky testing setup for the</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>CSAC experiment</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>Concept of operations for the CSAC flight experiment</td>
<td>58</td>
</tr>
<tr>
<td>42</td>
<td>Allan deviation using receiver estimated clock bias from CSAC and Rb live</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>sky tests</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>Allan deviation using receiver estimated clock bias comparing Chip live sky</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>tests</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>Locked-clock testing setup using CSAC as external reference to GPS receiver</td>
<td>61</td>
</tr>
<tr>
<td></td>
<td>and Spirent simulator</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>Locked-clock testing using the CSAC, Rb, and Spirent simulator as external</td>
<td>62</td>
</tr>
<tr>
<td></td>
<td>reference to GPS receiver and Spirent simulator</td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>Flight mode testing setup using CSAC as external reference to GPS receiver</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>and Rb as external reference to Spirent simulator</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>Allan deviation using receiver estimated clock bias from locked-clock, flight</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>mode, and live sky tests</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>Allan deviation using receiver estimated clock bias of Ralphie in live sky</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>and constant temperature hot tests</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>Detrended clock bias of Ralphie comparing live sky and constant temperature</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>hot tests</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>Time series of bias, bias-rate, detrended bias, and temperature for CSAC</td>
<td>66</td>
</tr>
<tr>
<td></td>
<td>orbital temperature profile test</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>Closed-Loop Steering Exp. GRC (1/3)</td>
<td>72</td>
</tr>
<tr>
<td>52</td>
<td>Closed-Loop Steering Exp. GRC (2/3)</td>
<td>72</td>
</tr>
<tr>
<td>53</td>
<td>Closed-Loop Steering Exp. GRC (3/3)</td>
<td>73</td>
</tr>
<tr>
<td>54</td>
<td>Gantt Chart (1/2) Spring 2021</td>
<td>74</td>
</tr>
</tbody>
</table>
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Roles and Responsibilities</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>Resource List</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>Budget</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>GNU Radio Blocks</td>
<td>24</td>
</tr>
<tr>
<td>5</td>
<td>CSAC experiment log messages</td>
<td>59</td>
</tr>
<tr>
<td>6</td>
<td>Past and Present CONTACT Team Members</td>
<td>71</td>
</tr>
</tbody>
</table>

Gantt Chart (2/2) Spring 2021 | 74
2 Introduction

In August 2019, the DoD publicly released an unclassified version of its “Strategy for the Department of Defense Position, Navigation, and Timing (PNT) Enterprise: Ensuring a U.S. Military PNT Advantage.” Emphasizing the foundational role of GPS PNT in Joint Force operations, the Strategy warns that “space-based PNT services provided by GPS will be targeted and will not always be available in contested military operating areas. . . complementary PNT capabilities must be applied.” [1]

While GPS is best known as a navigation system, one of its critical functionalities in providing global PNT capability is disseminating precise time, time intervals, and frequency. Communications systems use both time and frequency to maintain carrier frequencies and data-bit phase timing, and military networks have especially stringent timing requirements for the synchronization of data encryption and decryption equipment [2]. Since the 1950s, the gold standard for timekeeping has been ground-based atomic clocks; which are generally not suitable for spaceflight or ground vehicles due to their size, weight, power requirements, and environmental sensitivity [3]. With that said, atomic clocks used for GNSS applications provide a consistent, accurate, and inexpensive external frequency source to both military and civil users around the globe. As a part of the “complementary PNT capabilities” aimed at mitigating the inherent vulnerability of DoD dependence on GPS, the Air Force Research Lab (AFRL) is interested in developing accurate onboard timing technologies that are less reliant on frequent GPS time corrections. The AFRL Space Vehicles Directorate is sponsoring a graduate student project under Dr. Penina Axelrad at the University of Colorado Boulder in alignment with the AFRL mission to develop and transition emerging technologies for military space applications. The Colorado Nanosat Atomic Clock Testbed (CONTACT) project objective is to develop a testbed capable of ensembling three or more low-SWaP (size, weight, and power) atomic clocks and producing an accurate and robust time signal for small satellite applications. Advanced PNT satellites could use this technology to achieve a highly accurate time scale when GPS is unavailable. Satellites in GPS-denied environments with strict position accuracy requirements could also use an improved onboard time reference to support precise orbit determination [4].

The following sections of this report discuss the project scope and give a description of the overall system. From there, the subsequent sections cover each of the subsystems in detail, as well as overall system hardware implementation and the MAXWELL CSAC Flight Experiment. Each section details development completed this semester and next steps.
3 Project Overview

3.1 Project Description

The continued development of accurate, robust onboard timekeeping technologies is foundational to the improvement of non-GPS PNT capabilities for a multiplicity of military and civil applications. Complementary PNT enabled by low-SWaP (Size, Weight, and Power) onboard timing systems can increase the resilience of communication and navigation systems in degraded or denied GPS environments. The limited performance inherent in low-SWaP clocks can be mitigated by ensembling several independent devices and generating a steered output frequency signal based on the weighted averages of the inputs. A testbed can be used to evaluate the performance of various clock configurations and signal steering techniques as well as to measure the effects of environmental factors and clock errors. The objective of this report is to describe the design and implementation of this clock ensembling testbed on a Software Defined Radio (SDR) hardware platform during the Spring 2021 semester. Results from this work could be used in the development of low-SWaP onboard timing systems capable of meeting Precise Time and Frequency (PT&F) requirements in the absence of GPS.

The testbed will be a standalone, modular, fully integrated system, that takes a reference and three low-SWaP atomic clocks under test as inputs and generates a steered output signal along with clock characterization reports. The system is currently implemented on a 4-input SDR and programmed with the GNU Radio open-source software development toolkit. The testbed uses well-defined SDR oscillator metrology techniques to measure high-frequency signals while maintaining low onboard storage requirements. The testbed is designed to take phase and frequency measurements of signals in the 10 MHz – 200 MHz range to ensemble clocks of dissimilar accuracy and stability. The system must also have the capability to identify clock errors such as jumps in phase and frequency and evaluate the robustness and resilience of the steered output signal. The measured time series data (phase and frequency) are processed to produce Allan Deviations (ADEVs) and plots for the characterization of performance. A filtering subsystem estimates the bias and drift of each clock under test. The filter uses a standard clock model and the calculated variance values for clock state propagation. The filter outputs deviation and covariance time histories for comparison of actual errors with three-sigma error statistics. The clock estimates are also used to produce a composite clock timescale in a weighted-average ensembling algorithm. The outputs of the ensemble are phase and frequency tuning corrections, which are passed into the signal generation subsystem along with the corresponding reference signal to be steered. The steered digital output is passed through a PLL to generate an analog output signal that has
been steered and smoothed. This output signal can be measured for controller performance analysis and signal characterization under variable testing parameters.

3.2 Project Objective

The Colorado Nanosat Atomic Clock Testbed (CONTACT) three-year project objective is to design, build, and operate a clock ensemble testbed for low cost, high accuracy timing applications. The function of the testbed is to measure multiple Chip-Scale Atomic Clock (CSAC) inputs, combine them using filtering and ensembling algorithms, and generate a steered output signal more accurate and robust than any one of the individual clocks. The testbed system will be modular, standalone, and support customization including multiple clock inputs, ensembling methods, and fault management capabilities.

3.2.1 Semester Objectives

The CONTACT project was initiated in Spring 2019 and has wrapped up its fifth semester in Spring 2021. This report highlights the developments made during Spring 2021. Following the road-map from previous semesters, further testing was done using the Ettus N310 and the characterization of the signal produced on N310 using SDR software, GNU radio companion. The MAXWELL CSAC Flight Experiment, which was initiated in Spring 2020, continued work on ground testing hardware in cooperation with the MAXWELL team.

During Spring 2021, the main objective for the testbed was to complete verification and validation of each component in the signal processing chain in GNU radio. Most of the work focused on signal generation subsystems. The steering steering capabilities of the numerically controlled oscillator (NCO) were tested in both open-loop and closed-loop configurations. In addition, hardware tests were conducted to characterize the use of the N310 as a measurement system and its ability to make minor frequency adjustments.

The MAXWELL CubeSat launch date has been pushed to late 2021 or early 2022, giving us more time to conduct environmental ground testing. This semester we focused on live sky testing, thermal testing, and studying the influence of the torque rods on the CSAC due to the proximity of torque rods to CSAC. These activities will be discussed in detail in later sections.
3.3 Resources

3.3.1 Roles and Responsibilities

This semester report is part of a structured reporting plan which also includes internal monthly individual progress reports and regular reviews with AFRL. This reporting plan is intended to mitigate the effects of team discontinuity between academic years or between semesters as well as facilitate stakeholder involvement in the project development. Shown below is the CONTACT team organization chart for Spring 2021 as well as a description of roles and responsibilities for select stakeholders. Project POC: luciana.schement@colorado.edu

Figure 1: Spring 2021 Organization Chart
### 3.3 Resources

#### Table 1: Roles and Responsibilities

<table>
<thead>
<tr>
<th>Members</th>
<th>Role/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dr. Joanna Hinks</td>
<td>Sponsor POC - provides requirements, use cases, and guidance on project development. Reviews progress at scheduled intervals.</td>
</tr>
<tr>
<td>Dr. Penina Axelrad</td>
<td>Faculty advisor – provides subject area expertise and direction to other useful university resources and personnel</td>
</tr>
<tr>
<td>Luciana Schement</td>
<td>Project Management, MAXWELL CSAC experiment lead</td>
</tr>
<tr>
<td>Quinn LaBarge</td>
<td>Systems Engineer, SDR software subsystem development</td>
</tr>
<tr>
<td>Caroline Dixon</td>
<td>MAXWELL CSAC experiment testing and integration</td>
</tr>
<tr>
<td>Harkuver Preet Singh Sidhu</td>
<td>Signal generation, SDR hardware and software testing</td>
</tr>
<tr>
<td>Laura Davies</td>
<td>Filtering subsystem, RINEX data processing</td>
</tr>
<tr>
<td>Christopher Flood</td>
<td>Measurement subsystem, software subsystem interfacing</td>
</tr>
</tbody>
</table>

#### 3.3.2 Other Resources

#### Table 2: Resource List

<table>
<thead>
<tr>
<th>Resource</th>
<th>Point of Contact</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aerospace Instrumentation Shop</td>
<td>Trudy Schwartz</td>
<td>Signal measuring/analysis equipment</td>
</tr>
<tr>
<td>MATLAB/Simulink</td>
<td>OIT</td>
<td>Implementation of Kalman filter and signal generator</td>
</tr>
<tr>
<td>COMPASS Lab</td>
<td>Dr. Penina Axelrad</td>
<td>GPS receiver and antenna, electronics workstation, and RF simulator</td>
</tr>
<tr>
<td>Microsemi</td>
<td>Richard Foster/Peter Cash</td>
<td>Contacts at the company that produces the CSACs</td>
</tr>
<tr>
<td>RF Simulator</td>
<td>Dr. Dennis Akos</td>
<td>RF simulation setup</td>
</tr>
<tr>
<td>MAXWELL Graduate Project</td>
<td>Anastasia Muszynski</td>
<td>MAXWELL contact for implementation and integration of the CSAC experiment</td>
</tr>
</tbody>
</table>
3.4 Schedule

Figure 2 below shows the Gantt chart for Spring 2021. A full-page version can be found in Appendix I.

<table>
<thead>
<tr>
<th>TASK</th>
<th>ASIGNED TO</th>
<th>PROGRESS</th>
<th>START</th>
<th>END</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Measurement Subsystem</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Characterize CSACs using Updated Measurement Sys</td>
<td>CF</td>
<td>90%</td>
<td>1/5/21</td>
<td>2/1/21</td>
</tr>
<tr>
<td>Characterize USRP2 Tx Signal Stability using N310 Rx</td>
<td>CF</td>
<td>50%</td>
<td>1/20/21</td>
<td>2/3/21</td>
</tr>
<tr>
<td>Measure all 3 CSAC Signals on Custom Breakout Boards</td>
<td>CF</td>
<td>0%</td>
<td>2/8/21</td>
<td>2/17/21</td>
</tr>
<tr>
<td><strong>Filtering and Ensembling Subsystem</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document Fault Management Plan</td>
<td>QL/LD/HS</td>
<td>0%</td>
<td>2/1/21</td>
<td>4/5/21</td>
</tr>
<tr>
<td>KF Output Rate Testing</td>
<td>LD/HS</td>
<td>0%</td>
<td>2/8/21</td>
<td>2/17/21</td>
</tr>
<tr>
<td>Verify the KF Steering Model Parameters (KF Tuning)</td>
<td>LD/HS</td>
<td>0%</td>
<td>2/8/21</td>
<td>3/8/21</td>
</tr>
<tr>
<td><strong>Signal Generation Subsystem</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integrate Steering Commands with Signal Generation Sys</td>
<td>CF/QL</td>
<td>70%</td>
<td>1/7/21</td>
<td>2/3/21</td>
</tr>
<tr>
<td>USRP2 Steered Signal Transmission Test</td>
<td>CF/QL</td>
<td>40%</td>
<td>1/19/21</td>
<td>2/4/21</td>
</tr>
<tr>
<td>Real-time Steered Signal Generation Testing (Closing the Loop)</td>
<td>CF/QL</td>
<td>0%</td>
<td>2/22/21</td>
<td>4/12/21</td>
</tr>
<tr>
<td>Characterize Steered Signal Stability</td>
<td>CF/QL</td>
<td>0%</td>
<td>3/22/21</td>
<td>4/5/21</td>
</tr>
<tr>
<td><strong>Fault Injection and Detection Subsystem</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create Custom Phase Jump Block</td>
<td>QL</td>
<td>0%</td>
<td>2/8/21</td>
<td>2/17/21</td>
</tr>
<tr>
<td>Integrate Fault Injection Sys with Measurement Sys</td>
<td>QL</td>
<td>0%</td>
<td>2/15/21</td>
<td>2/22/21</td>
</tr>
<tr>
<td>Transfer Fault Detection Algorithm to KF Clock Model</td>
<td>QL/LD/HS</td>
<td>0%</td>
<td>2/25/21</td>
<td>3/5/21</td>
</tr>
<tr>
<td>Real-time Fault Injection/Detection Testing</td>
<td>QL</td>
<td>0%</td>
<td>3/10/21</td>
<td>3/22/21</td>
</tr>
<tr>
<td>Document Fault Attribution Plan</td>
<td>QL</td>
<td>0%</td>
<td>2/1/21</td>
<td>4/5/21</td>
</tr>
<tr>
<td><strong>MAXWELL CSAC Flight Experiment</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Locked Clock Testing</td>
<td>LS/CD</td>
<td>100%</td>
<td>1/11/21</td>
<td>1/28/21</td>
</tr>
<tr>
<td>Live sky test with CSACs</td>
<td>LS/CD</td>
<td>100%</td>
<td>1/11/21</td>
<td>2/5/21</td>
</tr>
<tr>
<td>Thermal Testing</td>
<td>LS</td>
<td>100%</td>
<td>1/15/21</td>
<td>2/8/21</td>
</tr>
<tr>
<td>CSAC UART Driver F5W</td>
<td>CD</td>
<td>5%</td>
<td>1/25/21</td>
<td>9/3/21</td>
</tr>
<tr>
<td>GPS/CSAC Functional/Acceptance</td>
<td>LS</td>
<td>100%</td>
<td>2/4/21</td>
<td>2/18/21</td>
</tr>
<tr>
<td>Simulator-GPS-Beaglebone-SD CardTest</td>
<td>LS/CD</td>
<td>0%</td>
<td>2/18/21</td>
<td>3/14/21</td>
</tr>
<tr>
<td><strong>MAXWELL Antenna Testing</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maxwell Antenna Testing</td>
<td>LS</td>
<td>100%</td>
<td>3/14/21</td>
<td>4/3/21</td>
</tr>
<tr>
<td>Thermal Profile Testing</td>
<td>CD</td>
<td>20%</td>
<td>4/1/21</td>
<td>9/1/21</td>
</tr>
<tr>
<td>Measure Torque Rod Electromagnetic Field Strength</td>
<td>CD</td>
<td>10%</td>
<td>4/29/21</td>
<td>9/1/21</td>
</tr>
<tr>
<td>Test CSAC in increased EM field</td>
<td>CD</td>
<td>10%</td>
<td>4/29/21</td>
<td>9/1/21</td>
</tr>
</tbody>
</table>

The project team has completed documentation deliverables and schedule-driven reviews concluding with an end-of-semester review on April 22, 2021. Efforts during the Spring 2021 semester have included the following accomplishments:

- Individually tested three CSACs using GPS receiver with roof antenna drop to compare
the performance of each clock

- Conducted thermal testing experiments using CSAC as external reference to GPS receiver to better understand the relationship between the CSAC performance and temperature variations

- Conducted electromagnetic field testing using CSAC as external reference to GPS receiver to characterize CSAC performance in an increased magnetic field

- Measured CSAC phase using Rubidium as reference signal input to characterize the use of the N310 as a measurement system

- Phase measurements of three CSACs were used to generate phase difference measurement inputs to a Kalman filter

- Conducted open-loop and closed-loop steering experiments to determine the capabilities of Ettus N310 to make fine frequency adjustments

Moving forward into Fall 2021, development efforts will include:

- Determine the ability of N200 to characterize steered signals

- Use observed measurement noise and process noise as the filtering parameters

- Observe and compare the stability of steered output signal using the SDR on which the signal is generated (N310) and with an independent SDR (N200)

- Generate steered signal using auxiliary hardware (e.g., OCXO)

- Conduct tests where input signals are intentionally degraded (phase, frequency errors)

- Conduct thermal profile testing with all three CSACs to characterize each clock’s performance

- Measure the strength of the electromagnetic fields produced by the torque rods

Each of these subsystems shown in the Gantt chart, is discussed in detail in later sections.
3.5 Budget

The budget spreadsheet shown in Table 3 shows an accounting for the first 2.5 years of project funds which extends through April 2021. Included in this spreadsheet are all previous expenses and all planned major expenses for CY21.

Table 3: Budget

<table>
<thead>
<tr>
<th>Source/Expense Item</th>
<th>Project Funds Available</th>
<th>Expected Expenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFRL Grant (less CU overhead)</td>
<td>$48,000</td>
<td></td>
</tr>
<tr>
<td>Previous Expenditures</td>
<td></td>
<td>$31,216</td>
</tr>
<tr>
<td>Spring 2021 Expenditures</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Department Overhead</td>
<td></td>
<td>$0</td>
</tr>
<tr>
<td>Clocks - LN-CSAC</td>
<td></td>
<td>$6,607</td>
</tr>
<tr>
<td>Electronics - TX Boards, magnetometers</td>
<td></td>
<td>$256</td>
</tr>
<tr>
<td>Other - PTTI Conference</td>
<td></td>
<td>$599</td>
</tr>
<tr>
<td>Subtotal</td>
<td>$48,000</td>
<td>$38,678</td>
</tr>
<tr>
<td>Final Balance</td>
<td>$9,322</td>
<td></td>
</tr>
</tbody>
</table>

3.6 Requirements

1. The measurement subsystem shall be capable of measuring the phase of selected clocks against a reference clock

   1.1. The measurements shall be taken at a rate of no less than once per minute
       - Rationale: To enable optimal filtering and signal steering
       - V&V: Timestamps attached to measurements to verify the sampling rate

   1.2. The measurement system shall store up to 24 hours of continuous measurements
       - Rationale: 24 hours is the upper bound for potential continuous runtime of the testbed
       - V&V: Collect measurements for a full day and inspect the timestamp data to verify the test duration

   1.3. The measurements shall be output to the filtering subsystem within TBD minutes of being generated
3.6 Requirements

- Rationale: Real-time steering performance depends on prompt delivery of measurements to the filtering subsystem
- V&V: Timestamp data can be used to determine the lag time between subsystems

1.4. The measurement system error shall be less than the specified performance error of the highest quality clock
- Rationale: Necessary for measuring the stability of the member clocks and steered output signal
- V&V: Analysis of measurements to identify potential biases and measurement noise variances

1.5. The measurement subsystem shall be capable of simultaneously measuring at least three 10MHz oscillators
- Rationale: The filtering subsystem for the clock ensemble relies on measurements from three or more member clocks
- V&V: Demonstrate simultaneous measurements from multiple input channels

1.6. The measurement system shall be capable of utilizing an external reference clock
- Rationale: A stable reference clock is required to assess the stability of measured input clocks
- V&V: Design analysis of testbed architecture and measurement subsystem

2. The testbed shall be capable of characterizing oscillators

2.1. The characterization system shall accept clock phase measurements as inputs
- Rationale: Require a standard input for ease of use with outputs of different subsystems (modularity of the testbed components)
- V&V: Design and analysis of characterization subsystem architecture

2.2. The characterization system shall generate Allan deviations
- Rationale: Allan deviations represent clock frequency stability across different time intervals and are normalized by an oscillator’s nominal frequency for simplified comparison to other clocks
- V&V: Verify Allan deviations with Stable32

2.3. The characterization system shall calculate and output the clock variances to the filtering system
3.6 Requirements

- Rationale: Tuning the filters requires estimates of the process noise and measurement noise covariances
- V&V: Calculate variances for clocks with known process noise and measurement covariances (e.g. simulated clock signals)

3. The testbed shall estimate clock biases and clock drifts using one or more filtering approaches

3.1. The filtering system shall produce state estimates of each member clock

- Rationale: The ensemble filtering subsystem accounts for process noise and measurement noise
- V&V: The state estimates will be run through the characterization subsystem and the estimation error covariance matrix will be computed at each sample interval

3.2. The filtering subsystem shall consist of modular functions

- Rationale: Ability to switch between filtering algorithms
- V&V: Demonstrate multiple filter modules

3.3. The filtering system shall be capable of processing output data from measurement system

- Rationale: Construct the necessary interface between the measurement and filtering subsystems
- V&V: Demonstrate the ability of filtering subsystem to accept and process data input from the measurement subsystem

4. The testbed shall be capable of ensembling the clock state estimates output by the filtering subsystem

4.1. The input to the ensembling algorithm shall be the clock state estimates output by the filtering subsystem

- Rationale: Standard inputs/outputs of each subsystem to enable modularity of testbed components
- V&V: Demonstrate the ability of the ensembling subsystem to accept and process data input from the filtering subsystem

4.2. The ensembling algorithm shall output corrections based on the clock estimate to the signal generation system at TBD rate
3.6 Requirements

- Rationale: The corrections are applied within the signal generation subsystem to steer the output signal.
- V&V: Demonstrate the ability of the ensembling subsystem to pass clock corrections to the signal generation subsystem.

5. The testbed shall produce a steered output signal with better frequency stability than the member clocks of the ensemble.

5.1. The signal generation system shall steer a nominal 10 MHz signal.
- Rationale: The steered signal is at the same nominal frequency as the member clocks.
- V&V: Measure the frequency of the output signal with an oscilloscope and the testbed.

5.2. The signal generation system shall generate a steered sinusoidal output signal.
- Rationale: The overall objective of the clock ensemble testbed.
- V&V: Evaluation of output signal via feedback to measurement subsystem and characterization subsystem.

5.3. The signal generation system shall feed back the steered output signal to the measurement system.
- Rationale: Required for closed-loop control and characterization of the steered output signal.
- V&V: Store measurements of the steered signal for post-processing and stability analysis.

5.4. The signal generation system shall be capable of processing the clock ensemble state that is output from the ensembling algorithm at a TBD rate.
- Rationale: Generate a sufficiently accurate and robust output signal based on real-time, filtered/ensembled measurements.
- V&V: Analysis of signal generation system design (i.e. optimal steering interval) and time tagging of the control inputs (i.e. steering commands).

6. The system shall be capable of identifying clock faults and managing the effects of faults on the clock ensemble and steered output signal stability.

6.1. The system shall be capable of injecting faults in simulation through the modification of measurement data.
3.7 ConOps

The Concept of Operations for this project in Figure 3 shows the nominal system use case.

![Concept of Operations Diagram](image)

Figure 3: Concept of Operations

Moving from left to right along the ConOps diagram, the nominal use case includes:

1. Clock Selection – Identify 3 or more low-SWaP atomic clocks for test.

2. Measurements – The clock signals are input to the testbed, where phase and frequency measurements are taken with respect to a reference clock.

3. Fault Injection – The user can evaluate system performance and resilience under non-ideal conditions by inducing or simulating clock errors for each of the member clocks.

- Rationale: Ability to inject faults with known magnitudes and directions will facilitate testing of the real-time fault detection and fault management algorithms
- V&V: Demonstrate fault injection with real-time signals measured by the testbed (i.e. faults should appear in the measurements)

6.2. The system shall be capable of inducing faults by varying environmental conditions

- Rationale: Explore environmental effects on system performance (e.g. temperature variations or EM field variations)
- V&V: Thermal chamber testing and Hemholtz cage testing

3.7 ConOps

The Concept of Operations for this project in Figure 3 shows the nominal system use case.
4. Filter & Ensemble – Measured clock signals are filtered to provide state estimates for each of the member clocks with respect to the implicit ensemble mean (IEM). The state estimates and measurements of the steered signal are used to estimate the offset between the steered signal and the IEM. Steering commands are generated based on this estimated offset and sent to the signal generation subsystem.

5. Steered signal output – Steering commands output by the filtering/ensembling algorithm are applied to an oscillator to drive the output signal toward the IEM.

6. Characterization - Allan deviation plots and phase error plots are created to assess the stability performance of the steered signal over various time scales in comparison to the member clocks.

3.8 Testbed Architecture

The functional block diagram in Figure 4 shows the boundary of the standalone testbed system. The full, integrated testbed is currently being implemented using the Ettus N310 software defined radio (SDR). The digital signal processing chain is implemented using GNU Radio which runs on a PC connected to the SDR. The testbed accepts clock signal inputs and generates a steered output signal along with characterization reports. Each subsystem block within the testbed is described below.

![Functional Block Diagram](image)

Figure 4: Functional Block Diagram

3.8.1 Measurement Subsystem

The measurement subsystem accepts input signals from multiple clocks and makes phase and frequency measurements for comparison between the selected clocks. Ideally, the mea-
measurement subsystem will be capable of accepting multiple types of clocks for simultaneous input. The clocks under test are sampled and the measurement data is formatted and sent to the filtering subsystem.

Included in the measurement subsystem are the fault injection processes. The user has the ability to add both frequency and phase jumps directly onto the measurements of the member clocks. These clock faults are added in simulation to avoid exposing the member clocks to adverse environmental conditions necessary to induce a fault. In addition, the precise time, magnitude, and direction of the clock faults are controlled by the user. This information is used to verify the functionality of the fault detection algorithm.

3.8.2 Characterization Subsystem

The characterization subsystem processes clock time series data to produce Allan Deviation (ADEV) plots. The ADEV plots are used to determine the types of noise affecting the stability of the clock signals over different time intervals. The values are sent to the filter and ensemble subsystem to update the process noise and measurement noise covariance matrices. The ADEVs will also be compared to the frequency stability specifications provided by the manufacturer of the member clocks.

3.8.3 Filter/Ensemble Subsystem

The filtering subsystem produces estimates for the bias and drift of each member clock [5]. The filter uses a standard clock model and the variance values from the characterization subsystem for the filter clock state propagation. The differenced clock measurements (i.e. with respect to one of the member clocks) from the measurement subsystem are used as input to the filter observation update. The filter outputs clock state estimates relative to the implicit ensemble mean.

A separate filter is used to estimate the frequency error between the steered output signal and the implicit ensemble mean. This filter generates an optimal frequency adjustment for the steered output signal.

3.8.4 Signal Generation Subsystem

The signal generation subsystem serves the role of producing the realization of the implicit ensemble mean. This is accomplished by steering the frequency of one of the clocks under test towards the IEM [6]. In order to perform this required task, a numerically controlled oscillator is periodically adjusted based on optimal frequency corrections computed by the
filtering/ensembling system. The output of the NCO is sampled by the DAC of the Ettus N310 to transmit the steered signal.

4 Ettus N310 and GNU Radio

In Spring 2020, the Ettus N310 SDR was selected for the clock ensemble testbed [7]. Figure 5 shows the N310, which features four transmit and four receive ports. The N310 was chosen, in part, due to the number of available RX channels. At least three channels are required to measure and ensemble the CSACs under test, while the fourth channel can be used to measure the steered output signal and close the control loop. The CONTACT team purchased 3 CSACs and one Low-Noise CSAC (LNCSAC) in previous semesters. To help differentiate the three CSACs, they were nicknamed Ralphie, Chip, and SpaceBuff. The SDR also supports a 1Gb Ethernet connection with a host PC, on which the digital signal processing chain is executed using GNU Radio.

This section outlines the processes of sampling the member clocks and transmitting the steered output signal using the N310 and GNU Radio. The digital representation of these timing signals, in GNU radio, is discussed along with a method of simulating clock faults upstream of the measurement subsystem. In addition, this section provides an overview of program development in GNU Radio and introduces challenges associated with implementing the real-time signal processing chain using GNU Radio. These challenges are discussed in greater detail in the Signal Generation section.

Figure 5: Ettus N310 SDR[7]

To begin, the maximum signal input power for the Ettus N310 is -15dBm. Therefore, input signals (e.g. CSAC with 3.3Vpp) must be attenuated to levels below the maximum allowable input power. In addition, the Ettus N310 is designed to support signals ranging from 10MHz to 6GHz. The ensemble member clocks (i.e. the Rb frequency standard and CSACs)
output signals at 10MHz. Sampling clocks at the low end of the N310’s RF capabilities requires an additional upconversion stage within the SDR. To be specific, the four receive channels of the N310 are split across two daughterboards. These daughterboards employ AD9371 transceivers, which support frequencies above 300MHz. The N310 upconverts the 10MHz clock signals into the operational range of the transceivers prior to being sampled by the ADC.

The upconverted input signals are then downconverted by the N310 to a near-baseband frequency for input to the GNU Radio signal processing chain. It is important to note that the near-baseband signals reduce the execution time and data storage requirements associated with the testbed digital signal processing chain by reducing the required sampling rate for the input signals. The downconversion process is commanded by a UHD tune request in GNU Radio, where the user specifies a center frequency (in Hz) and a local oscillator (LO) offset (in Hz). The center frequency represents the difference between the input clock frequency (i.e. 10MHz) and the desired near-baseband frequency. For example, a center frequency of 9999900Hz is used to shift a 10MHz clock input down to 100Hz. Shifting the input signal down by the center frequency is achieved in two stages. First, the input signal is mixed with the LO, which is offset from the center frequency by the user-specified frequency. Again, the analog mixing process does not fully shift the input signal to the desired near-baseband frequency. Assuming the same 10MHz clock signal, a 9999900Hz center frequency, and an LO offset of 250kHz, the analog mixer will shift the input signal down to 250100Hz (i.e. 100Hz + 250kHz). The digital down-converter then shifts the signal by the LO offset frequency, producing a signal at the desired near-baseband frequency. In the previous example, the digital down-converter shifts the output of the analog mixer by 250kHz to produce the desired 100Hz signal.

The same tune requests are also used to transmit signals using the SDR. In that case, near-baseband signals are upconverted to 10MHz RF signals using the same center frequencies and LO offsets.

Within GNU Radio, each downconverted clock signal is captured as a combination of two sinusoids, referred to as the in-phase and quadrature (I & Q) components of the input signal, and stored as complex variables within GNU Radio. The complex signal can be represented using Euler’s formula (Eq. 1). The signal phase, $\theta(t)$, is the argument of the complex exponential term. In Eq. 2, $Im(z(t))$ is the imaginary component of the complex number, while $Re(z(t))$ is the real component.

$$z(t) = e^{i\theta(t)} = \cos \theta(t) + i \sin \theta(t)$$

23
\[ \theta(t) = \tan^{-1}\left( \frac{\text{Im}(z(t))}{\text{Re}(z(t))} \right) \]

The complex representation facilitates insertion of simulated clock faults, including frequency jumps and phase jumps. To insert frequency jumps, a complex single-sideband (SSB) mixer multiplies the downconverted clock signal with another simulated signal at a relatively low frequency. The result is a clock signal at the difference frequency of the two complex signals. Phase jumps are inserted by adding a constant angular offset to the phase of the complex signal.

As previously discussed, the input signals are downconverted by the Ettus N310 to reduce the computational load on the host computer. The downconverted member clock signals are input to the testbed signal processing chain, shown in Figure 4, that is implemented in GNU Radio. The GNU Radio Companion (GRC) application allows the user to connect pre-built and custom blocks to form a flowgraph. There are several useful standard pre-built blocks, a few of which are listed below in Table 4.

<table>
<thead>
<tr>
<th>Block Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UHD: USRP Source/Sink</td>
<td>Enables the user to receive/transmit signals from/using the SDR (includes the UHD tune request)</td>
</tr>
<tr>
<td>File Source/Sink</td>
<td>Read or write data streams from/to a binary file to enable pre/post-processing</td>
</tr>
<tr>
<td>Complex to Arg/Arg to Complex</td>
<td>Convert between I &amp; Q data and signal phase</td>
</tr>
<tr>
<td>Add/Subtract/Multiply/Divide</td>
<td>Perform standard element-wise operations on a stream input data</td>
</tr>
<tr>
<td>Rational Resampler</td>
<td>Resample a signal through interpolation/decimation of an input data stream</td>
</tr>
<tr>
<td>Signal Source</td>
<td>Generates a digital signal during program execution</td>
</tr>
<tr>
<td>QT GUI Sink</td>
<td>Displays the frequency spectrum and time-domain of signals during program execution</td>
</tr>
</tbody>
</table>

Custom blocks, written in Python, perform operations specific to the clock ensemble testbed. Custom blocks are used to inject clock errors in simulation. Custom Python blocks are also used to measure and unwrap the input signal phases. The Kalman filters which comprise the filtering subsystem are implemented using custom blocks. The steering commands generated by the filtering/ensembling algorithm are applied to a numerically
controlled oscillator (NCO), which is another example of a custom Python block within the clock ensemble testbed flowgraph.

The testbed flowgraph begins with streams of I & Q data for each member clock input to the Ettus N310. The dynamic GNU Radio scheduler attempts to optimize the program performance by breaking the streams of data into large chunks. The overhead associated with moving data is significant. The program runs more efficiently (i.e. with higher throughput) when each block in the flowgraph operates on large chunks of data at a time, minimizing data movement. So, GNU Radio sets up the program such that each block includes an input buffer that fills up with the chunks of data. However, this presents a few challenges for real-time systems[8].

First, the GNU Radio scheduler interferes with real time execution of feedback loops within the flowgraph. Feedback loops generally require processing each sample in real time. However, transferring one sample at a time is not feasible with the GNU radio framework. In some instances, loops can be contained within a single custom Python block. In other circumstances, asynchronous message passing interfaces can be used to circumvent the loop issue.

Second, the input buffers affect the input and output rates of each block. A block will operate on the input data once the input buffer is filled. As a result, a block may be waiting for the input buffer to fill up and will not output data at the desired rate. For example, suppose the filtering/ensembling subsystem is consistently waiting on measurements to fill its input buffers. Then, steering commands may not be output to the signal generation subsystem at the desired steering interval. Thus, it is important to be able to measure the input and output rates of the testbed subsystems.

With that said, the signal generation system does not necessarily need to apply steering commands at a high rate. A main objective of the clock ensemble testbed is to generate a steered output signal with stability performance better than the individual member clocks. If the short-term stability of the generated signal is observed to be better than the member clocks without steering, then steering commands can be applied at longer time intervals to meet the system requirements. These issues associated with the implementation of the real-time system are discussed in further detail in the Signal Generation section.
5 Measurement Subsystem

5.1 Measurement Process

The process for making phase measurements using the SDR is outlined in Figure 6. The clocks are shown on the left side of the diagram producing 10MHz signals. After the SDR performs the frequency mixing process, the signal that is sent to the host PC is at a lower frequency \( f \approx 73\text{Hz} \) with a sampling rate of 120kHz. During the frequency mixing process the clock onboard the Ettus N310 contributes a common error that is present in all three signals. The argument of the complex signal is computed and results in a phase from \([-\pi, \pi]\). This wrapped phase is unwrapped, producing an approximately linear phase growth. The expected phase growth is computed and subtracted from the measured phase growth, which yields phase information that represents the stability of the original input clock. However, the measured phase information of the CSACs is corrupted by a relatively low stability clock that is on the SDR. The trend in the measured phase deviation for each clock over time will be dominated by the clock onboard the SDR, as shown in Figure 7.

The expected phase growth for each input signal is computed based on the beat frequency of the signal and elapsed time, both of which are computed by the host PC using Python scripts. Since the CSACs are reasonably stable clocks, the phase deviations over time are much smaller than the expected phase growth that is subtracted from the measured phase growth. When the measured phase growth is unwrapped prior to subtracting the expected phase growth, the magnitudes of the phase values become very large relative to phase deviations we are trying to measure. A method of phase subtraction is performed on the complex signal streams prior to computing the argument of the signal to avoid the accumulation of error that results in subtracting two large phase values. This process is explained in Appendix 11.3.

![Figure 6: Process for measuring clock phase](image-url)
5.2 Measurement Experiments

The error contribution from the SDR, common to all signals shown in Figure 7, is eliminated when the signals are measured against a reference that has the same error contribution. A timing signal from a Rb frequency reference is applied to the fourth channel of the SDR and goes through the same signal processing chain outlined in Figure 6. The measured clock bias from the rubidium input is subtracted from all of the clock biases in Figure 7 and results in the measured CSAC clock biases shown in Figure 8. The results in Figure 8 are more consistent with the CSAC stability specifications outlined by the manufacturer. With the common error contribution from the SDR removed, the behavior of each CSAC is clearly visible.

5.3 Measurement System Characterization

There are two daughterboards on the Ettus N310 that handle the front end RF processing required to shift the 10MHz signals up to the minimum 300MHz frequency required for the AD9371. These daughterboards each contribute error to the incoming signals. With an ensemble of clocks, signals will be measured against another clock that may be on the same or different daughterboard. The error contribution from signals going into the same / across daughterboards is analyzed by connecting the same signal to input ports on the SDR and differencing the computed phase growth from the two signals. The difference in measured
phase deviations for identical signals reveals the error contribution from the processing chain. Figure 9 shows clock bias difference residuals on the scale of picoseconds for signals on the same daughterboard and Figure 10 shows clock bias difference residuals on the scale of nanoseconds for signals on different daughterboards.

![Figure 9: Clock bias residuals (picoseconds) from differencing the phase growth of identical signals on the same daughterboard](image)

![Figure 10: Clock bias residuals (nanoseconds) from differencing the phase growth of identical signals on different daughterboards](image)

Another test was run to quantify the measurement noise when using unique signals. Two CSACs and a Rb reference were put into the SDR. One of the CSAC signals was on the same daughterboard as the Rb and the other CSAC signal was on a different daughterboard. The time domain behavior of the individual signals is shown in Figure 11. The primary trend in Figure 11 is due to the common clock contribution from the measurement device. When the Rb signal is subtracted from each CSAC signal, as shown in Figure 12, the common error drops out and the remaining bias is the due to the frequency error of the clock under test.

The clock offsets and the noise of the measurement device are present in Figure 12, where signals measured across daughterboards are noisier than signals measured on the same daughterboard. The magnitude of the noise contribution is shown in Figure 13 where the same daughterboard noise is at the picosecond level and the cross-daughterboard noise is on the nanosecond level, corroborating the results in Figure 9 and 10. The device measurement noise is important to understand as it will affect the observed stability of the clocks under test. Figure 14 shows the overlapping Allan deviation for a longer test using the same CSACs. In the short term, the behavior of a CSAC that is measured against a signal on a different daughterboard may be dominated by the nanosecond level noise contribution of the measurement device. The blue curve in Figure 14 shows this occurring - the computed short
term stability is larger than the specification and much larger than the short term stability of the other CSAC. Some of this may be due to stability differences between CSACs, but the dominant contributor in the short term is most likely the cross daughterboard measurement noise. The measurements made on the same daughterboard enable the behavior of the CSACs to be observed at all time scales, but the noise in the cross-daughterboard measurements obscures the short term behavior of the CSACs.

Figure 11: CSAC & Rb Time Domain Behavior
6 Filter and Clock Ensemble Subsystem

The bias and frequency of one clock can only be measured relative to another reference. In our laboratory, rather than measuring a clock directly against a highly stable reference, the phase of the clock under test and the phase of a reference are measured with respect to a less

---

Figure 12: Clock bias after subtracting the Rb signal from each CSAC

Figure 13: Detrended clock bias (nanoseconds) of the CSAC signals measured against the same Rb signal

Figure 14: CSAC Stability Relative to a Rb Reference on the Same / Different Daughterboard
stable clock, as detailed in [9]. The difference in the phase measurements serves as the basis for characterizing the clock under test. In a small system like the CONTACT testbed, we also employ this approach, where the sampling clock of the SDR serves as the common clock source, and differences between two of the clocks under test, or a lab reference are used to characterize and create the clock ensemble. We implement a Kalman Filter following Brown [10] and also as outlined in [11].

6.1 Clock Model

A clock model is required for estimating the states based on the phase difference measurements. The clock model used in the project is a two state model which evolves according to the following expression:

\[ x_{k+1} = \phi(\tau)x_k + w_k \]  

(3)

Where the state transition matrix is defined by:

\[ \phi(\tau) = \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix} \]  

(4)

And process noise:

\[ w_k \sim \mathcal{N}(0, Q(\tau)) \]  

(5)

The process noise covariance is treated as Gaussian, zero mean, and with covariance \( Q(\tau) \) described by white and random walk frequency noise parameters, \( q_1 \) and \( q_2 \). These noise parameters are specific to each clock and will change based on the clocks used in the ensemble.

\[ Q(\tau) = \begin{bmatrix} q_1\tau + \frac{q_2\tau^3}{3} & \frac{q_2\tau^2}{2} \\ \frac{q_2\tau^2}{2} & q_2\tau \end{bmatrix} \]  

(6)

6.2 Measurement Model

The input measurements to the system are phase difference measurements between the ensemble member clocks. The phase difference measurements are used to estimate the phase and frequency of each of the member clocks.

\[ z_k = Hx_k + v_k \]  

(7)

\[ H = \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \]  

(8)
6.3 Filter Results

A system composed of \( N \) clocks and \( N - 1 \) relative clock measurements will always be unobservable when trying to estimate \( N \) clock states since an error common to all of the clocks will not manifest in the relative measurements \([10]\). An assessment of the observability matrix will show that the system is unobservable for 3 clock states.

\[
rank(O) = rank \left( \begin{bmatrix} H & \cdots & H \phi \cdots H \phi^{m-1} \end{bmatrix} \right) = 4 \ \forall \ m \geq 2
\]  

Since there are 6 elements of the system state vector, the covariance matrix of the state estimate in the filter will continue to grow without bound. The covariance reduction method detailed in \([10]\) separates the observable and unobservable components of \( P \) and uses the observable component to prevent unbounded uncertainty growth.

\[
P = P - H^* (H^*^T P H^*)^{-1} H^*^T
\]

6.3 Filter Results

An experiment was run with three member clocks connected to the SDR measurement pipeline. Phase difference measurements are made and sent to a Kalman filter that is running in real time. This filter uses phase difference measurements to estimate the phase and frequency states of each of the member clocks. Figure 15 compares the ADEV for the estimated clock states and for the clock states measured against a Rb reference.

The frequency stability of the CSACs when measured against the Rb yield similar results as the frequency stability of the estimated CSAC clock states from the Kalman filter. The relative order of the ADEV curves, as well as the size of the ADEV are similar between the two methods, indicating that the Kalman filter is properly estimating the clock state.
6.4 Future Work

A Kalman filter that estimates clock states based on phase difference measurements has been implemented and verified. Some of the next steps for the filtering system are to adjust the measurement noise values in the filter to reflect the observed noise values and incorporate logic that would accommodate inputs from a fault detection system. When a member clock is determined to be faulty, the clock needs to be appropriately handled through an increase in the process noise, removal from the system, or an alternative fault management method.

7 Signal Generation Subsystem

7.1 Signal Generation Architecture

7.1.1 Original Conceptual Design

The following subsection details the original, conceptual design for the signal generation subsystem. It is included for comparison to the current implementation of the signal generation subsystem in GNU radio and to provide context for a few key changes that were made to the design over the course of this semester. The design proposed by former Signal Generation Lead for CONTACT, Henry Dixon, and other CONTACT members, leverages concepts used by certain frequency standards to generate timing signal outputs and apply frequency tuning.
7.1 Signal Generation Architecture

A summary of the original signal generation subsystem design, as described by Henry Dixon in the Fall Semester Report[14], is as follows.

One of the ensembled 10MHz CSAC signals is selected as the steering reference ($CSAC_i$), providing the stable frequency source. This signal is frequency multiplied by a factor of four, and is used to clock a direct digital frequency synthesis (DDS) routine. The DDS behaves as a microphase stepper and applies frequency corrections ($\Delta f_{steer}$) to the nominal 10MHz frequency source. Optimal $\Delta f_{steer}$ adjustments are computed by a state estimator and regulator that are updated by feedback measurements and state deviation estimates ($z_i, \hat{x}_i$), received from the measurement and ensembling subsystems. The synthesis routine produces a low resolution sinusoidal output containing much unwanted harmonic noise. The low resolution steered signal is smoothed by a PLL that produces a low phase noise output - locked to the digital output of the DDS. A functional block diagram for the signal generation subsystem is provided in Figure 16 below.

Before pursuing the baseband steering approach with the Ettus N310 and GNU radio, the baseband steering approach was validated with alternative hardware and software. The validation test compares the stability performance of a baseband steering approach to a high-rate steering method. Henry Dixon performed this test in Fall 2020. An overview of the experiment and the results, from the Fall Semester report and Henry’s thesis, are given below ([15] [14]).

A myRIO digital controller was used to execute the high-rate and baseband real-rime steering systems. The algorithms for both systems steered an externally generated (Koolertron CJDS66 function generator) 53Hz sine wave to 54Hz over a period of 200 seconds. More specifically, frequency adjustments of +0.05Hz are applied to the DDS every second. While the high rate system directly steers the 53Hz input to 54Hz, the baseband system steers a downconverted signal at 5Hz (i.e. with a 48Hz center frequency) to 6Hz using the
same +0.05Hz frequency adjustments. Then, the 6Hz signal is upconverted to 54Hz using the same center frequency. Data were collected at 400S/s with an Arduino Uno microcontroller. Figures 17 and 18 show high level functional block diagrams for each steering system.

Figure 17: Block diagram for the high-rate steering system implementation.

The downconversion chain in Figure 18 shifts the 53Hz signal down to a 5Hz complex signal using a combination of sine/cosine mixers and a lowpass filter. The upconversion chain shifts the steered, smoothed output of the PLL up by 48Hz using the complex transmitter method shown in [16].

Figure 19 shows the frequency stability of the output signals produced by each system for two trials.
Figure 19: Overlapping Allan Deviation result comparison between steering methods.

The long-term stability results show that initial values match up fairly well with the specified stability of the function generator clock (±1 ppm - clocks the input signal) and the Z-7010 FPGA clock (±10 ppm - clocks the output signal). At longer time intervals the steered outputs track their respective input signals. A slight difference in frequency stability is observable between the two methods. Steered outputs from the high-rate system appear to perform better than the baseband system outputs by approximately a factor of 2. While there is a difference present in these results, the baseband steering approach does not appear to significantly degrade short-term or long-term steering performance. In theory, the baseband steering approach can be adapted for the testbed hardware and software to produce stable output signals.

### 7.1.2 Implementation using Ettus N310 and GNU Radio

Now, with the baseband steering method validated, the steering algorithm was implemented in GNU Radio with a few key changes to the original conceptual design. As previously discussed, there are challenges associated with implementing real-time processes in GNU Radio. First, the dynamic GNU Radio scheduler restricts the use of loops within a flowgraph. Second, the input buffer sizes control the input and output rates of the testbed subsystems. 
To begin, the PLL is removed from the signal generation system. In the validation tests, the PLL produces a smoothed, lower phase noise signal which locks to the output of the DDS. This loop structure is difficult to implement in GNU Radio. The PLL requires the continuous input/output of a low phase noise oscillator. Incorporating this oscillator and obtaining the signal within GNU Radio requires low-level programming.

The need for the smoothing PLL is eliminated by replacing the low resolution DDS with a higher resolution numerically controlled oscillator (NCO). At a high level, the NCO operates in a similar manner to the DDS. A signal at a known nominal frequency is input to the NCO. This signal acts as the clock source for the NCO, triggering the NCO to update the phase of the output signal. The phase increment (in radians) applied by the NCO at each time step is a function of the desired output frequency $f_{\text{desired}}$ and the frequency of the input clock source $f_{\text{trigger}}$ (Eq. 12).

$$\Delta \theta_{\text{NCO}} = 2\pi f_{\text{desired}} / f_{\text{trigger}}$$  \hspace{1cm} (12)

In the previous section, the clock source is a pulse signal at four times the frequency of the reference clock. In this case, the clock source is replaced by a simulated signal that is sampled at a high rate. Ultimately, the sample rate of the simulated signal source drives the sample rate of the NCO. More specifically, the NCO is implemented as a sync block within GNU Radio. Sync blocks generate an equal number of output items as there are input items. Therefore, the NCO will inherit the sampling rate of the input clock source. With a sufficiently high sampling rate (i.e. relative to the near-baseband frequency of the output signal), the NCO can achieve a finer resolution than the original DDS.

For example, suppose the member clocks are downconverted to 73Hz. Measurements of the 73Hz signals are input to the filter/ensembling subsystem which generates a steering command. In the original algorithm, one of the 73Hz signals is passed through the frequency multiplier to produce a 292Hz pulse signal that triggers the DDS. The low resolution steered output of the DDS requires filtering and smoothing prior to upconversion. In contrast, suppose the NCO block is triggered by a signal source sampled at 120kS/s. Therefore, the steered output signal of the NCO block will be sampled at 120kS/s. The sampling rate is significantly higher, resulting in reduced phase noise compared to the DDS. Frequency power spectra of the NCO output show clean peaks at the desired output frequency.

Next, the desired output frequency of the NCO must be adjusted in real-time. Steering commands are generated at the desired steering interval by the filtering/ensembling subsystem. To this point, a steering interval of 1 second (i.e. one steering command per second) has been used for testing. However, longer steering intervals may be practical. The stream of steering commands cannot be directly input to the NCO block. The custom sync block
does not function correctly with multiple input streams at different rates. That is, the NCO
block will not work properly with the high sampling rate signal source (triggering the phase
increments) and the low rate steering command data stream.

To overcome this issue, an asynchronous message passing interface was constructed in
GNU Radio. Messages containing the steering commands with attached time tags (the
time tags are a record of when the steering command was generated) are sent from the
filtering/ensembling system to the NCO. A message handling function within the custom
NCO block operates independently of the main work function, and frequency adjustments
can be made without interrupting the flow of the sync block[8].

The message passing interface utilizes Polymorphic Types (PMTs). This special data
type is required to send data asynchronously between blocks. Steering commands generated
within one of the custom Python blocks of the filtering/ensembling subsystem are paired
with time tags to form tuples. The tuples containing the steering commands and time tags
are converted to PMTs. The PMTs are then published to an output message port attached
to the custom Python block.

A complementary input message port is added to the custom NCO block. Within the
NCO block, the message handler function converts incoming PMTs back to the tuples con-
taining the steering commands associated time tags. Another time tag is computed when the
message is decoded to maintain a record of when the steering commands are applied. From
there, the decoded steering commands are used to update the appropriate NCO parameters
(i.e. the desired output frequency and corresponding phase increment based on the NCO
sample rate), while the time tags are saved to file for post-processing. To reiterate, this
message handler function is separate from the main work function of the NCO block. As a
result, the NCO is able to output a continuous stream of I & Q data without having to wait
for steering commands.

Figure 20 shows the message passing interface in GNU radio. The data streams are
represented by colored ports and solid connecting arrows. The message passing interface is
represented by the gray ports and dashed connecting arrows. The Message Debug block can
be used to print the contents of the messages to the terminal in order to verify the messages
have been sent or received.
At this point, the NCO is able to continuously output a complex (I & Q) signal with the desired output frequency. The desired output frequency can be adjusted in real time via steering commands passed through an asynchronous messaging interface. The current version of the optimal controller is designed to compute steering commands at a rate of 1Hz. However, there is uncertainty associated with the input/output (I/O) rates of various blocks due to the I/O buffer sizes set by GNU Radio’s dynamic scheduler. GNU Radio passes large chunks of data at a time to optimize program performance. With simulated input clocks, the NCO block has been observed applying 8-16 steering commands at once. With that said, the steering commands cannot be lumped together. If multiple steering commands accumulate within an input buffer and are eventually applied at the same time, the stability of the output signal will be degraded.

The messages sent to the NCO block from the filtering/ensembling subsystem include time tags attached to each steering command at the moment it was generated. These time tags are referred to as generation time tags. When the messages are decoded, separate time tags are computed to record the moment each steering command is applied. These steering commands are referred to as reception time tags. The duration between consecutive generation time tags, $\Delta t_{\text{generation}}$, provides insight into the rate at which steering commands are generated. Similarly, the duration between consecutive reception time tags, $\Delta t_{\text{reception}}$, provides insight into the rate at which steering commands are applied by the NCO block.
In the following experiment, the Ettus N310 is configured to accept two inputs: the 10MHz Rubidium (Rb) frequency standard and the steered output signal of the NCO upconverted by the Ettus N310. Both inputs are downconverted using a center frequency of 9999927Hz (i.e. 10MHz-73Hz) and an LO offset of 250kHz. The downconverted signals are sampled at 120kS/s. The NCO sampling rate is set to 240kS/s, and the Ettus N310 upconverts the output of the NCO using the same center frequency and LO offset.

The frequency error of the steered signal relative to the Rb clock is computed every second. Initially, rather than using an optimal controller, this experiment uses a simple proportional feedback controller to compute steering commands. That is, the filtering/ensembling subsystem is omitted from the flowgraph. The generation time tags are computed for each steering command. The differences between consecutive generation time tags ($\Delta t_{\text{generation}}$) are recorded to a file. The reception time tags are computed as the steering commands are applied within the NCO block, and their differences ($\Delta t_{\text{reception}}$) are saved to a file. The results are shown in Figure 21.

![Steering Command Generation/Reception Intervals (1Hz Model)](image)

Figure 21: Steering Interval - Generation and Reception of Steering Commands

In the figure above, the interval number indicates which two adjacent steering commands are under inspection. Interval 1 corresponds to steering commands 1 and 2. Interval 2 corresponds to steering commands 2 and 3, and so on. The interval between generation of subsequent steering commands is 1.0000s, on average. Similarly, the mean interval between reception of subsequent steering commands is approximately 1.0000s. The standard
deviations of $\Delta t_{\text{generation}}$ and $\Delta t_{\text{reception}}$ are $4.3 \times 10^{-3}$s and $4.0 \times 10^{-3}$s, respectively. In fact, deviations from the expected steering interval are consistent between the two plots. For example, 1.0166 seconds elapsed between the generation of steering commands 32 and 33 (interval number 32). The results also show that approximately 1.0164 seconds elapsed between the application (i.e. reception) of steering commands 32 and 33, a difference of around 0.0002 seconds compared to $\Delta t_{\text{generation}}$. This trend is visible for each interval over the entire duration of the test.

The histograms in Figure 22 show the distributions of $\Delta t_{\text{generation}}$ and $\Delta t_{\text{reception}}$ across 20 bins. The histograms depict the means and standard deviations presented above. Moreover, the distributions are almost identical due to the trend described earlier.

![Steering Command Generation/Reception Intervals (1Hz Model)](image)

Figure 22: Steering Interval Distribution - Generation and Reception of Steering Commands

Although this test utilizes a simplified version of the testbed signal processing chain, the results show that steering commands are generated and applied at the desired steering interval. GNU radio’s dynamic scheduler is sensitive to changes in the sampling rates used in a flowgraph. In this case, the sampling rates are relatively high (120kS/s for the inputs and 240kS/s for the output). It is likely that the I/O buffers are being filled/emptied by these data streams at a sufficiently high rate compared to the steering interval, enabling the program to keep up with the hardware.

Despite achieving the desired steering interval with this particular flowgraph, a secondary approach was developed in the case that the observed steering interval does not match the
desired steering. In the event that steering commands are accumulating in the input buffer of the NCO block, a separate zero-order-hold (ZOH) control input block can be enabled. This ZOH block rapidly fills the input buffer of the NCO block with null steering commands. These null steering commands do not affect the NCO parameters. Their sole purpose is to saturate the input buffer, allowing the actual steering commands to reach the message handling function at the desired rate. The actual steering command inputs are "held" until the next one is received.

The ZOH block operates in a similar manner as the NCO block. A simulated signal source with a high sampling rate is input to a custom sync block. The custom sync block inherits the sample rate of the input signal and is able to generate null steering commands at the same rate. These null steering commands are delivered to the NCO via a message passing interface. The relevant GNU Radio blocks are shown in Figure 23.

Figure 23: GNU Radio Flowgraph - ZOH Control Input Blocks

7.2 Open-loop Steering Experiment

To further test the capabilities of Ettus N310 and the Numerically Controlled Oscillator (NCO) in making minor frequency adjustments, simpler open-loop and closed-loop experiments were set up. The main motive is to see if we can create a stable signal at a desired frequency; we try to do so by steering the output of Ettus N310 from a 100 Hz signal to 110
7.2 Open-loop Steering Experiment

Hz using the Numerically Controlled Oscillator (NCO) and the Message block (MSG). The block diagram representation of the test setup can be seen in Figure 24 shown below:

![Block Diagram Representation of Open-loop steering](image)

Figure 24: Block Diagram Representation of Open-loop steering

In this hardware-based experiment, a reference 10 MHz Rubidium signal at 120k samples per second was passed into Ettus N310. Since the input is at 10 MHz, it is first downconverted to 100 Hz. The input frequency of 10 MHz is shifted by the center frequency of 9999900Hz such that the signal output from the USRP Source block is 100 Hz. In this experimental setup, the Message block used is a Python Embedded Block (PEB), which passes a +0.05 increment frequency adjustment to the NCO based on the incoming signal’s sample rate. The input Rubidium signal passes through a Rational resampler block that decimates the signal from 120 kS/s to one sample per second before passing it on to the message block. The small frequency adjustments are now passed to the NCO block that triggers once it receives a signal from another source signal at 240 kS/s. Once the NCO receives the messages from the Message block, it starts steering the signal from 100 Hz to 110 Hz. Output from the into your block is fed back into the measurement system, Ettus N310, and the data is collected for further analysis. Something to note here is that the Ettus N310 collects the unsteered signal data, which is the reference Rubidium signal, and the steered signal (output of the NCO).

The purpose of this experiment was to understand the steering process better using the Ettus N310 and be able to compare the stability of unsteered and steered signals. To verify that the rubidium input and the NCO’s output are at 100 Hz and 110 Hz, the spectra of collected samples are computed and shown in Figures 25 and 26.
7.2 Open-loop Steering Experiment

Figure 25: Power Spectrum of Unsteered Signal

Figure 26: Power Spectrum of Steered Signal

In these figures, we can see that peaks at 98.47 Hz and 109.77 Hz, comparable to 100 Hz and 110 Hz, respectively—this small error is the constant frequency offset that we observe while experimenting with Ettus N310. Figure 27 and 28 shows the bias for both unsteered and steered signal. We could see that the unsteered signal fluctuates a lot more than the steered signal, but the bias is minimal in both cases.

Figure 27: Detrended Bias of Unsteered Signal

Figure 28: Detrended Bias of Steered Signal

To compare the stability, the Overlapping Allan deviations (OADEVs) of the recorded unsteered and steered data were computed. The results are shown in Figure 29. Since the same oscillator is used to create and measure the steered signal, the ODEVs show high stability, as seen in Figure 29.
To further test the steering performance of the Ettus N310, a beat signal was generated by mixing the output of the NCO with the input signal and passing it through a low pass filter to eliminate noise [15]. If the steering was performed correctly, the power spectrum should show us a peak at 10 Hz, which can be seen in Figure 30. Note that the peak is seen at 11.52 Hz rather than 10 Hz, which is the addition of the 1.52 Hz frequency shift we notice in the unsteered spectrum earlier. The difference in the frequency offset between the unsteered (1.52 Hz) and steered (0.223 Hz) could be because of the difference in the bias.
The next step in the development of the signal generation subsystem is to evaluate the capacity of the testbed hardware and software to steer an output signal to a reference clock. In contrast to the open-loop steering experiments, the closed-loop steering experiments are designed to test the ability of the signal generation system to keep pace with a real-time closed-loop steering algorithm. The steered output signal must respond to the dynamics of the reference clock and the dynamics of the oscillator used to generate the output signal (i.e. the internal oscillator of the Ettus N310). For this experiment, the output signal is steered toward the CSAC. The test setup is outlined in the paragraphs below and in Figure 31. The test results revealed another issue with using GNU Radio to execute the digital signal processing chain.
There are three inputs to the Ettus N310 SDR: a CSAC (Ralphie\(^1\)), the steered output signal, and the Rubidium (Rb) frequency standard. The ideally 10MHz input signals are downconverted to 73Hz using a center frequency of 9999927Hz and an LO offset of 250kHz. Each downconverted signal is sampled at 120kS/s, and immediately resampled at 400S/s using Rational Resampler blocks. The captured I & Q data for each input signal are recorded to a file for post-processing.

Next, the frequency error of the steered signal with respect to the CSAC is computed. There are a few steps to compute the frequency error. First, the CSAC signal is mixed with the complex conjugate of the steered signal. This results in a complex signal at the difference frequency. The phase of the complex signal at the difference frequency is computed at each sample epoch, giving the phase error between the two signals. The phase error is then unwrapped, giving the accumulated phase error between the two signals. The time derivative of the accumulated phase error is approximated using a first-order divided difference, giving the approximate frequency error in radians per second. Dividing by \(2\pi\) gives the instantaneous, observed frequency offset of the steered signal with respect to the CSAC, in Hz, between two samples. The instantaneous frequency errors are averaged over the course of a second (i.e. over 400 samples). In essence, the accumulated phase error at

\(^1\)A nickname for one of CONTACT’s 3 CSACs.
the end of one second is divided by $2\pi$ seconds to approximate the frequency error every second\[17\].

For this experiment, the steering commands are produced by multiplying the computed frequency errors by a constant feedback gain value of 0.4. In other words, each steering command is a frequency adjustment for the NCO that is proportional to the frequency error between the steered output signal and the CSAC. Preliminary tests showed that larger gain values tended to over-steer the output signal, resulting in larger frequency errors every second. Smaller gain values limited the ability of the proportional feedback controller to keep pace with the dynamics of the system (i.e. clock drift, clock faults, etc.).

The steering commands are paired with time tags and sent to the NCO via an asynchronous message passing interface. From there, the steering commands are decoded and the frequency adjustments are applied to the NCO which is sampled at 120kS/s. The output of the NCO is resampled at 240kS/s to match the required sampling rate for the Ettus N310’s DAC. Furthermore, the resampled output of the NCO is upconverted with the same center frequency and LO offset used for the three inputs. Again, the steered output signal is fed back into the Ettus N310 to complete the control loop.

The I & Q data for the steered signal and Rb are also recorded using the N200 as a measurement device. The same downconversion process is driven by equivalent tune requests (i.e. with the same center frequency and LO offset). This allows for characterization of the steered output signal (with the Rb as a reference clock to mitigate common errors associated with the SDR’s internal oscillator) using measurements from multiple devices. The GNU Radio Companion flowgraph for the experiment is included in the Appendix.

The results of the closed-loop steering experiments are shown in the plots below. The steering experiment was run for approximately 95 minutes, starting at 19:45 MST. Figures 32 and 33 show the detrended clock biases and fractional frequencies of the CSAC and steered signal with respect to the Rb reference clock, where all signals are measured by the Ettus N310.
The detrended bias plot shows that the frequency of the CSAC is drifting with relative to the Rb. That is, the clock bias is not growing at a constant rate. Therefore, the difference in the frequency between the CSAC and Rb must be changing over time. The fractional frequency is, on average, \(-2.9161 \times 10^{-10}\).
7.3 Closed-loop Steering Experiment

The detrended bias of the steered signal, with respect to the Rb reference, exhibits much different behavior. There are multiple spikes in the detrended bias curve. These spikes coincide with underruns in GNU Radio. Underruns are error flags that pop up in the command window during program execution to indicate an insufficient amount of data is being delivered to a USRP device. In this case, it is assumed that the Ettus N310 is not receiving enough data from the NCO to continuously transmit the steered output signal. Occasionally, the Ettus N310 is forced to wait for the digital signal processing chain to catch up with rest of the testbed hardware (i.e. the SDR). After a moment, the N310 resumes transmission of the steered output signal. However, this gap in the transmitted signal significantly degrades the stability performance of the steered output signal. The spikes are also visible in the observed fractional frequencies. Between the underruns, the fractional frequency is at a level comparable to the CSAC (Figure 34).
7.3 Closed-loop Steering Experiment

Colorado Nanosat Atomic Clock Testbed

The steered signal output before and after the underrun at 838.45 seconds into the test is shown in Figure 35. The I & Q data for around the underrun is shown in the plot on the left, while the power spectrum of the steered signal, computed over the entire duration of the test, is shown on the right. We see a few hundredths of a second where the signal level drops to around zero, skipping multiple cycles. Afterward, the transmitted signal resumes. Again, these gaps extend over multiple cycles of the steered output signal. The resulting phase errors and frequency errors following underruns are relatively large compared to the periods between underruns.

To verify the existence of these gaps in the steered output signal, the same post-processing
7.3 Closed-loop Steering Experiment

is applied to measurements taken with the Ettus N200. The detrended bias and fractional frequency of the steered signal (with the Rb reference) are shown in Figure 36.

![Detrended Bias vs. Time - Steered](image1)

![Fractional Frequency vs. Time - Steered](image2)

Figure 36: Steered Signal Detrended Bias and Fractional Frequency (Rb Reference) - N200

The detrended bias and fractional frequency data show spikes at the same locations coinciding with the underruns in GNU Radio. There are noticeable differences in these curves compared to those corresponding to measurements taken with the Ettus N310. The disparity between them requires further investigation.

The underruns clearly degrade the stability of the steered output signal. However, the observed fractional frequencies between underruns are comparable to the CSAC. So, the period between underruns from 4300 seconds to 5000 seconds is selected for further analysis. This is one of the longer stretches of the experiment without an underrun. The detrended bias and fractional frequency data for the CSAC and steered output signal are shown in Figure 37, where the Rb is used as a reference signal.
During this stretch of the experiment, the detrended bias of the CSAC is on the order of a few nanoseconds. The detrended bias of the steered signal is slightly larger. The detrended bias of the steered signal roughly follows the detrended bias curve for the CSAC. However, it is difficult to determine whether the steering algorithm is effectively driving the output signal toward the CSAC.

The fractional frequency data for the CSAC and steered signal overlap during this 700 second window of the test. In contrast to the detrended bias curves, the differences in the fractional frequency curves are less apparent. The observed fractional frequency for each signal is negative, indicating that the CSAC and steered output frequencies are lower than the Rb reference.

The detrended bias and fractional frequency data for the CSAC and steered output signal are shown in Figure 38, where the Rb is not used as a reference signal. The detrended biases
and fractional frequency are computed with respect to an ideal 10MHz signal downconverted to 73 Hz.

![Detrended Bias vs. Time (no Rb ref)](image1)

![Fractional Frequency vs. Time (no Rb ref)](image2)

Figure 38: CSAC and Steered Signal Between Underruns - N310

In this case, the common errors associated with the Ettus N310 internal oscillator are not removed. The detrended biases of the CSAC and steered signal are an order of magnitude larger compared to the equivalent curves in Figure 37. Interestingly, the detrended bias of the steered signal is smoother than the CSAC, and lags behind the CSAC by around 10 to 15 seconds during the test. While the mean fractional frequency is approximately \(-2.710^{-8}\) for both the CSAC and steered signal, the variances are at a similar level to Figure 37. This is reflected in the Allan Deviations.

The ADEVs for the period between underruns are computed. The segment of the test chosen for this analysis is 700 seconds. Accordingly, ADEVs are only computed for sample intervals up to 100 seconds. The results are shown in Figure 39, where the CSAC manufac-
turer frequency stability specifications are included for reference[18].

As expected, the measured CSAC signal ADEVs are close to the manufacturer specifications. In addition, the ADEVs for the CSAC with the Rb reference are lower at longer sample intervals. This supports the idea that the common clock errors associated with the SDR’s internal oscillator are being removed through comparison to the reference signal. The ADEVs of the steered signal with respect to a Rb reference are larger than those for the steered signal without a Rb reference over short time intervals. This may be due to the decent short-term stability of the SDR’s internal oscillator. This is not the case for longer time intervals, where the ADEVs for the steered signal with respect to a Rb reference are smaller. Again, this suggests that the common clock errors associated with the N310’s internal oscillator are being removed for longer sample intervals.

Without the Rb reference, the steered signal ADEVs are smaller than the CSAC ADEVs across all time intervals. At this point, the team suspects that the stability performance (characterized through ADEVs) may be misrepresented when a signal is both generated and measured using the same SDR device. This phenomenon warrants further investigation.

The closed-loop steering experiment shows poor stability performance for the output signal steered to the CSAC (Ralphie) due to frequent underruns. In the periods between underruns, the frequency error of the steered signal with respect to the Rb reference closely matches the frequency error of the CSAC with respect to the Rb reference. In fact, the ADEV for the steered signal is at most 2.6 times larger than the ADEV for the CSAC for
the same sample interval (100 seconds). This is a promising result. During this 700 second window, the non-optimal steering algorithm does not significantly degrade the output signal stability.

Eliminating the underruns during the steering experiments will eliminate the subsequent phase-time and frequency errors which will dramatically degrade the stability performance. Therefore, it is a priority for the signal generation subsystem to identify the cause of the underruns and modify the flowgraph in order to avoid them in the future. A longer test, free of underruns, will hopefully reveal whether the Ettus N310 is capable of generating a signal steered toward another clock (e.g. a CSAC) in the long-term.

7.4 Future Work

The signal generation subsystem is one of the last components of the testbed to be implemented in GNU radio. Work is ongoing to determine if the Ettus N310 is capable of generating a steered output signal with the desired stability performance. Further closed-loop steering experiments are required to determine if the underrun issue can be resolved.

If the underrun issue is resolved, the stability of the steered output signal can be re-evaluated. A few changes can be made to the closed-loop steering experiment setup. First, the steering interval and proportional gain value can be adjusted. Steering intervals of 5 seconds or 10 seconds may yield better stability performance at these longer sample intervals. Additionally, the gain value may need to be adjusted to account for changes in the steering interval. Second, the method of computing the frequency error can be updated. Rather than averaging the frequency error over the steering interval, a recursive least squares (RLS) estimation of the frequency error can be computed. This simplified approach is still less than complex than the Kalman filtering approach for the full clock ensemble testbed. However, in theory, an RLS algorithm will produce a more accurate estimate of the frequency error.

Alternative hardware options will be explored for the signal generation subsystem. Specifically, tunable OCXOs have been identified as potential external oscillators that could be used to realize the IEM. The OCXOs of interest have short term frequency stabilities better than the ensemble member clocks (i.e. the CSACs).

Once the output signal can be continuously steered with the desired precision, a full testbed integration test can be performed. That is, the various subsystems will be connected in GNU radio and run in an attempt to steer the output signal to the IEM. The full filtering/ensembling subsystem will provide steering commands to the signal generation subsystem instead of the simplified approaches detailed above. At that point, the fully integrated testbed output signal can be characterized.
8 CSAC Flight Experiment on MAXWELL CubeSat

The CONTACT team is working with the MAXWELL project at CU Boulder to develop an experiment to characterize a CSAC onboard a cubesat. For the flight experiment, the CSAC will be used as an unsteered, external reference to a NovAtel OEM729 GPS receiver. This will cause the receiver’s clock bias and frequency estimates to be driven by the CSAC. The experiment is planned to run for five days during which the flight computer will record pseudorange, phase, clock bias, and position from the GPS receiver [19]. Data from the CSAC, including the temperature and status data, are of particular interest. The temperature data will be obtained in orbit to determine how varying thermal environments affect CSAC operation. These data will be downlinked and used to evaluate the accuracy and sensitivity of the CSAC in low Earth orbit. A block diagram of the flight configuration is shown in Figure 40.

![Block diagram of the flight configuration and live sky testing setup for the CSAC experiment](image)

There are four main phases of the CSAC experiment which are displayed in Figure 41. The preliminary data collection is used to ensure that the CSAC and receiver are operating correctly and to verify that valid data are being recorded. The preliminary stage collects the BestXYZ, ClockModel, and temperature messages at an interval of 30 seconds for a total of 3 hours. After the preliminary stage is completed, the data from this section are relayed back to Earth via the UHF antenna. Stage 1 collects the same messages as the preliminary stage at a higher rate for approximately 2 orbits (3 hours). Stage 2 collects data every 10 seconds for 90 hours or approximately 60 orbits. The bulk of the data collected for the CSAC

Figure 40: Block diagram of the flight configuration and live sky testing setup for the CSAC experiment

There are four main phases of the CSAC experiment which are displayed in Figure 41. The preliminary data collection is used to ensure that the CSAC and receiver are operating correctly and to verify that valid data are being recorded. The preliminary stage collects the BestXYZ, ClockModel, and temperature messages at an interval of 30 seconds for a total of 3 hours. After the preliminary stage is completed, the data from this section are relayed back to Earth via the UHF antenna. Stage 1 collects the same messages as the preliminary stage at a higher rate for approximately 2 orbits (3 hours). Stage 2 collects data every 10 seconds for 90 hours or approximately 60 orbits. The bulk of the data collected for the CSAC
experiment will occur during this stage. The final stage reduces the frequency to collect data every 30 seconds and adds range messages as well. This stage will operate for approximately 2 orbits. After stages 1-3 are completed, the files from each section are relaid via the UHF antenna.

![Diagram](image)

**Figure 41: Concept of operations for the CSAC flight experiment**

As the MAXWELL team prepares to deliver hardware, the CONTACT team is performing ground testing on the NovAtel OEM729 receiver and CSAC. Performing ground testing is essential for the success of the MAXWELL project and the CSAC experiment as it ensures that the hardware and software interface properly. Ground testing allows us to understand the CSAC performance on the ground providing a baseline for how the CSAC is expected to behave on-orbit. We will compare the results of the flight experiment to the baseline results to understand how the space environment affects the CSAC performance.

The ground testing includes live sky testing, RF/GNSS simulation testing, thermal chamber testing, and electromagnetic field testing. Live sky testing uses the GPS receiver and a roof antenna and yields the baseline performance of the CSACs on Earth. The RF simulator test sends simulated GPS signals to the receiver to replicate space-like scenarios. The thermal chamber experiments will characterize the CSAC performance under varying temperature profiles. The electromagnetic (EM) field testing will provide an understanding of how an increased EM field affects the CSAC performance.
Table 5: CSAC experiment log messages

<table>
<thead>
<tr>
<th>Log Message</th>
<th>Data Collected</th>
</tr>
</thead>
<tbody>
<tr>
<td>BestXYZ</td>
<td>Position in ECEF coordinates</td>
</tr>
<tr>
<td>ClockModel</td>
<td>Clock bias &amp; bias-rate</td>
</tr>
<tr>
<td>Range</td>
<td>Pseudorange, phase, &amp; C/No</td>
</tr>
<tr>
<td>CSAC Telemetry</td>
<td>Temperature</td>
</tr>
</tbody>
</table>

8.1 Live Sky Testing

Live sky tests were performed with the CONTACT team’s three CSACs nicknamed Ralphie, Chip, and SpaceBuff, to understand the performance of each clock. All are mounted on custom breakout boards that were designed by the CONTACT team in Fall 2020 [14]. Some initial tests were conducted with Chip mounted on a Microsemi development board, however, it has been moved over to a custom breakout board for further testing.

The live sky tests were performed using the COMPASS lab’s roof antenna drop (roof drop) atop the Smead Aerospace Engineering Sciences building at CU Boulder. The roof drop was connected to a NovAtel OEM729 GPS receiver and the CSAC board was connected to the receiver as an unsteered external clock as shown in Figure 40. Pseudorange, phase, clock bias, and clock bias-rate were collected from the GPS receiver at a frequency of 1Hz while the temperature data were collected from the CSAC physics package at the same frequency. The duration of the live sky tests varied from one to four days. The STIg lab at CU Boulder also has a roof antenna drop where a few live sky tests were conducted with a similar set up.

Live sky tests were conducted using Chip, Ralphie, SpaceBuff and a PRS10 rubidium frequency standard ($Rb$) to get a baseline for each clock’s performance on the ground as displayed in Figure 42.
8.2 RF/GNSS Simulator Testing

8.2.1 Locked-Clock Testing

The purpose of the locked-clock tests is to determine whether the simulator is able to lock to an external reference. For the RF/GNSS simulator locked-clock test, a Sprient STR4500 simulator was connected to the MAXWELL engineering development (EDU) receiver through the antenna port on a NovAtel development board. The MAXWELL EDU receiver is a single frequency NovAtel OEM729 receiver. The SpaceBuff CSAC was connected to the receiver as an unsteered external oscillator via the development board. SpaceBuff was also connected to the Spirent simulator as an external reference. The test setup is shown in Figure 44. In this test, since the reference clock for both the simulator and the receiver are the same, the clock bias should be zero.

In order to verify the clock performance on the custom breakout boards, live sky tests were conducted with Chip on a Microsemi development board and a custom breakout board. The ADEV plot is shown in Figure 43. The Chip 2-9 yellow line represents the live sky test with Chip on the breakout board. This ADEV is very similar to the other two Chip tests which were completed with Chip on the Microsemi board. Figure 43 verifies that the custom breakout boards are successful in creating similar performance that was shown on the Microsemi development board.

8.2 RF/GNSS Simulator Testing

Figure 42: Allan deviation using receiver estimated clock bias from CSAC and Rb live sky tests

Figure 43: Allan deviation using receiver estimated clock bias comparing Chip live sky tests
The locked-clock tests were successful in verifying that the simulator was able to lock to the CSAC and rubidium as external references. Figure 45 shows a time series of the bias, bias-rate, and detrended bias from three separate tests in which the simulator, CSAC, and Rb were used to drive the simulator and GPS receiver clocks. The CSAC appears to wander more than the Rb and simulator clocks. Since the magnitudes of the bias and bias-rate are very small, we can conclude that the simulator is properly locking to the external reference.
Figure 45: Locked-clock testing using the CSAC, Rb, and Spirent simulator as external reference to GPS receiver and Spirent simulator

### 8.2.2 Flight Mode Testing

The success of the locked clock tests allowed the team to move on to testing in flight mode. Flight mode testing connects the Spirent simulator to the EDU receiver through the antenna port on the NovAtel development board. The Rb is used as an external reference to the simulator while the CSAC is an unsteered external reference to the EDU receiver. This setup is labeled flight mode because it is comparable to the setup that will be used on-orbit. Figure 46 shows the block diagram of the flight mode test.
A flight mode test was conducted using the Ralphie CSAC and the same simulation that was used for the locked-clock testing. The simulation was a LEO which ran for approximately 3.5 hours. The flight mode and locked-clock Allan Deviations are shown in Figure 47. The flight mode ADEV is very different from what we observed in the live sky tests using the CSACs and the Rb. However, this behavior is similar to the ADEV plots from the locked-clock tests using the same LEO simulation.

Figure 46: Flight mode testing setup using CSAC as external reference to GPS receiver and Rb as external reference to Spirent simulator

Figure 47: Allan deviation using receiver estimated clock bias from locked-clock, flight mode, and live sky tests
8.3 Thermal Testing

One candidate hypothesis for the cause of the large periodic effect on the clock bias, is an unmodeled or mismodeled, simulated relativistic effect on the receiver clock. Looking at the simulation parameters, we found that the orbital eccentricity is set to zero such that there should be no oscillation in the bias due to relativistic effects. An offset in the frequency due to the balance between special and general relativistic effects at the simulated receiver altitude should be present. However, in a live sky tests, this effect would be indistinguishable from a simple offset. In fact, both represent actual frequency efforts of the clock, so this effect is not one to be removed.

8.3 Thermal Testing

An important part of the CSAC flight experiment is determining a relationship between the environmental temperature and the performance of the CSAC. The clock performance is affected by temperature ramp time, dwell time, and the number of cycles [20]. We can conduct testing on the ground with a thermal chamber to get a better understanding of this relationship prior to launch.

The thermal test setup is similar to the live sky tests, however, for this setup, the CSAC and breakout board are placed inside the thermal chamber. The thermal chamber is located in the STIg lab, so the tests were conducted using the STIg antenna. Pseudorange, phase, clock bias, and clock bias-rate were collected from the GPS receiver and the temperature was collected from the CSAC. The data sampling rate was 1 Hz from both the GPS receiver and the CSAC. Three types of tests were conducted using the thermal chamber: a constant temperature hot test, a constant temperature cold test, and an orbital temperature profile test.

8.3.1 Constant Temperature Hot Test

For the constant temperature hot test, the thermal chamber was set to a constant 35C and data were collected for a minimum of 20 hours. The purpose of this test was to get an understanding of how the CSAC is affected by warm temperatures and gain experience using the chamber. The CSAC did not demonstrate any loss of performance when soaked at an elevated temperature. In fact, Ralphie performed slightly better than expected during the thermal test. The Allan Deviation plot can be seen in Figure 48. The Ralphie - Thermal red line represents the thermal test using Ralphie in the STIg lab. This ADEV behaves very close to the CSAC spec. It is unclear as to why Ralphie performed slightly better than expected in the thermal chamber hot test.
In addition to the ADEV plot, the receiver estimated clock biases were detrended and plotted versus time in Figure 49. The plot shows that the STIg live sky test has a slow, 12 hour sinusoidal trend over time while the thermal tests stays fairly constant. This difference in the bias over time is likely the reason we see the difference in the Allan Deviation plot. However, it is still unclear why the thermal test was less noisy than the live sky test.

8.3.2 Constant Temperature Cold Test

In order to prepare for the orbital thermal profile test, we conducted a constant temperature cold experiment in the thermal chamber. The purpose of this test was to determine whether condensation developed at the minimum on-orbit internal temperature for MAXWELL. MAXWELL has a heater that turns on when the battery reaches 4C; this is approximately the minimum internal temperature for the cubesat. For this test, no hardware was used; instead, a metal plate was placed inside the chamber during the test. The thermal chamber was set to a constant 4C. After approximately 20 hours, the thermal chamber and metal plate were examined for signs of condensation. No condensation appeared in the chamber or on the metal plate. A paper towel was used to wipe the inside of the chamber and both sides of the plate to determine whether any condensation was present.

8.3.3 Orbital Temperature Profile Test

The thermal profile used for ground testing is motivated by a MAXWELL simulation study conducted by Rybak et al. [4]. The study suggests that the external temperature on MAXWELL could vary 8C on-orbit. The MAXWELL orbital period is approximately 90 minutes. The profile is designed to begin at -4C and increase to 4C in 0.5 hours. Once the
temperature reaches 4C, the temperature decreases to -4C in one hour. The duration of the test is approximately 20 hours. The purpose of this test is to enhance our understanding of the CSAC’s ability to compensate for temperature changes.

Figure 50: Time series of bias, bias-rate, detrended bias, and temperature for CSAC orbital temperature profile test

Figure 50 shows the bias, bias-rate, detrended bias, and temperature versus time for the orbital temperature profile test. The detrended bias is on the order of 10e-8 seconds which tells us that the CSAC is compensating for some of the temperature variations. However, the periodic peaks in the detrended bias show that the CSAC is directly affected by temperature changes. Further analysis and testing are required to understand the relationship between the environmental temperature and the CSAC performance.

8.4 Electromagnetic Field Testing

The CSAC performance in an increased magnetic field is an important aspect that is being explored in ground testing. The motivation behind this testing is that the CSAC is going to be placed within 10 cm of multiple magnetic torque rods utilized on the MAXWELL cubesat for attitude control. While the CSAC is approximately equidistant from the three
torque rods, the effects of the magnetic field at these close distances, as well as the effect of the combined magnetic field could be significant. The CSAC’s specification for magnetic sensitivity is $\pm 9 \times 10^{-11}$ /Gauss, up to two Gauss [18]. The manufacturer has indicated that the CSAC has not been tested in magnetic fields over 2 Gauss, and in fields over 10 Gauss, the Mu metal shielding around the CSAC could be permanently magnetized, thus rendering the CSAC ineffective. A model implemented by a member of the MAXWELL ADCS team indicated that the CSAC would experience about a 2 Gauss field at its current location due to a single torque rod. However, this model has the potential to be unreliable at small distances, and does not account for the presence of all three torque rods. Therefore, the testing of the actual magnetic field is essential. The goal of characterizing the CSAC performance in increased magnetic fields is being conducted in two different ways.

The first method is to utilize MAXWELL’s Helmholtz cage. The magnetic field of this Helmholtz cage was first investigated using a cell phone magnetometer. The maximum magnetic field that the Helmholtz cage can generate was found to be about 1.5 Gauss. Since this is a relatively safe magnetic field for the CSAC, a live sky test was then conducted with the CSAC placed inside of the Helmholtz cage. It is important to note that during this live sky test, the CSAC was the only piece of hardware within the Helmholtz cage; other hardware such as the GPS receiver was outside of the cage.

The second method to characterize the CSAC will be to utilize the actual torque rods. It will be necessary to investigate the strength of the magnetic field at differing distances away from the torque rod, as well as different positions around the torque rod, since it produces a dipole magnetic field. Additionally, the magnetic field at the distance the CSAC will be from the torque rod onboard MAXWELL will also be measured. Then, the CSAC will be placed at that distance and a live sky test will be conducted at this distance. The clock messages will then be analyzed in order to determine if there is a significant effect on the CSAC due to the increased magnetic field. Finally, this live sky test will be repeated in the presence of all three magnetic torque rods.

8.5 Future Work

Ground testing will continue to be the focus for the CSAC flight experiment this summer and in the fall. Environmental tests such as the orbital thermal profile and electromagnetic field tests will take priority. To ensure the success of the experiment on-orbit, it is important that we test and understand how both thermal and electromagnetic fields affect the clock. Testing to measure the magnetic field of the torque rods and additional orbital thermal profile tests are set to begin early this summer.
9 Acknowledgements

The team would like to acknowledge Dr. Joanna Hinks and her colleagues at AFRL who attended our presentations and gave helpful feedback for moving forward with the project. Additionally, the team would like to acknowledge Dr. Nicholas Rainville for providing project guidance, and Anastasia Muszynski from MAXWELL for her important contributions on the orbital CSAC experiment. The CONTACT team would like to give special recognition to Dr. Penina Axelrad for her invaluable support during this unique semester.
REFERENCES

References


## Appendix

### 10.1 CONTACT Team

Table 6: Past and Present CONTACT Team Members

<table>
<thead>
<tr>
<th>Spring 2019</th>
<th>Fall 2019</th>
<th>Spring 2020</th>
</tr>
</thead>
<tbody>
<tr>
<td>Henry Dixon</td>
<td>Henry Dixon</td>
<td>Henry Dixon</td>
</tr>
<tr>
<td>Andrew Mezich</td>
<td>Daniel Dowd</td>
<td>Daniel Dowd</td>
</tr>
<tr>
<td>Alexander Nichols</td>
<td>Christopher Flood</td>
<td>Christopher Flood</td>
</tr>
<tr>
<td>Zachary Reynolds</td>
<td>Yashica Khatri</td>
<td>Yashica Khatri</td>
</tr>
<tr>
<td>Margaret Rybak</td>
<td>Tyler Morris</td>
<td>Tyler Morris</td>
</tr>
<tr>
<td></td>
<td>Rahul Ramaprasad</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fall 2020</th>
<th>Spring 2021</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prayag Desai</td>
<td>Laura Davies*</td>
</tr>
<tr>
<td>Caroline Dixon</td>
<td>Caroline Dixon*</td>
</tr>
<tr>
<td>Henry Dixon</td>
<td>Christopher Flood*</td>
</tr>
<tr>
<td>Daniel Dowd</td>
<td>Quinn LaBarge</td>
</tr>
<tr>
<td>Christopher Flood</td>
<td>Harkuver Preet Singh Sidhu*</td>
</tr>
<tr>
<td>Quinn LaBarge</td>
<td>Luciana Schement</td>
</tr>
<tr>
<td>Luciana Schement</td>
<td>Dawson Weis</td>
</tr>
</tbody>
</table>

An asterisk indicates team members continuing with CONTACT in Fall 2021.
10.2 Closed-Loop Steering Exp. Flowgraph

Figure 51: Closed-Loop Steering Exp. GRC (1/3)

Figure 52: Closed-Loop Steering Exp. GRC (2/3)
10.3 Clock Phase Subtraction

The phase measurements of the input signals in GNU Radio will grow linearly over time. After the unwrapping process, the expected phase growth is computed and subtracted from the measured phase growth. The issue with this method is that as the measured and expected phase values grow over time without bound, the difference between the signals will lose numerical precision. Depending on the data type, the representation of continuously growing numbers will reduce the number points after the decimal that are maintained in memory. An alternative to this is to mix the signals before the phase computation to avoid this particular numerical issue.

Prior to computing the phase from the complex input signal, the input signal is multiplied by the complex conjugate of a reference to produce a signal with a frequency equal to the difference in frequency between the input and reference signals. The frequency of the resulting signal should be orders of magnitude smaller than the input frequency, and as such the phase grows much slower over time. This frequency subtraction operation prior to the phase computation enables a more accurate representation of the phase measurements.

\[
e^{j\theta_{meas}} = \cos \theta_{meas} + j \sin \theta_{meas}, \quad e^{j\theta_{exp}} = \cos \theta_{exp} + j \sin \theta_{exp}
\]

\[
e^{j\theta_{meas}} \cdot e^{j\theta_{exp}} = \cos(\theta_{meas} - \theta_{exp}) + j \sin(\theta_{meas} - \theta_{exp})
\]
10.4 Gantt Chart

Figure 54: Gantt Chart (1/2) Spring 2021

Figure 55: Gantt Chart (2/2) Spring 2021