

# MIMO Ground Station

*for satellite applications*

George Vardakis, Electra Karakosta - Amarantidou  
October 30, 2019



# Outline

- ① Project Overview
- ② Algorithms and Challenges
- ③ Results
- ④ Next Steps

# Outline

- 1 Project Overview
- 2 Algorithms and Challenges
- 3 Results
- 4 Next Steps

# Project Overview

- SDR Makerspace sub-activity
- Develop a functioning Multiple Input Multiple Output (MIMO) ground station
- 4-element antenna array with SDRs
- Beamforming for improved satellite reception
- Signal Processing with GNURadio

# Project Overview - Gains

- More easily constructed than rotator setup
- Multiple times higher Signal to Noise ratio than a single omni-directional antenna
- Suppress interference
- Observe multiple satellites simultaneously

# Outline

- 1 Project Overview
- 2 Algorithms and Challenges
- 3 Results
- 4 Next Steps

# Beamforming Algorithms

Two broad categories:

- Blind Estimators
  - Steer angle at the strongest signal source. No knowledge of arrival angle needed
- Non-Blind Estimators
  - Steer angle at desired direction of arrival

# Beamforming Algorithms

- Non-Blind Estimators
  - Phase Shift Beamformer
  - Minimum Variance Distortionless Response (MVDR)
  - Linearly Constrained Minimum Variance (LCMV)
- Blind Estimators
  - Least Squares Constant Modulus (LSCM)
  - Least Mean Square (LMS)



# Challenges

- Antenna arrays are constructed by multiple elements
- Most SDR devices support limited number of antennas
  - ☹ Connect multiple SDR devices together
  - ☹ They need to be synchronized in frequency and time

# Challenges

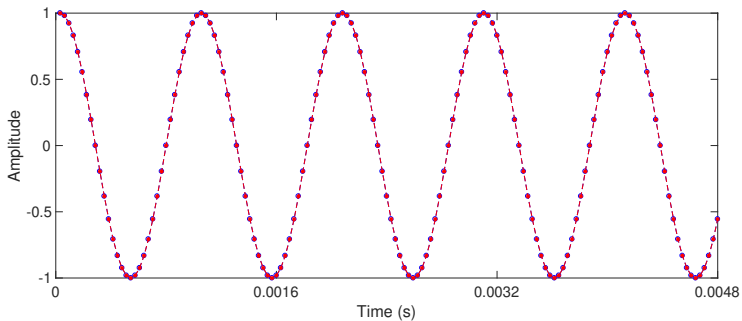
## **Frequency offset**

→ Different oscillators induce small offset

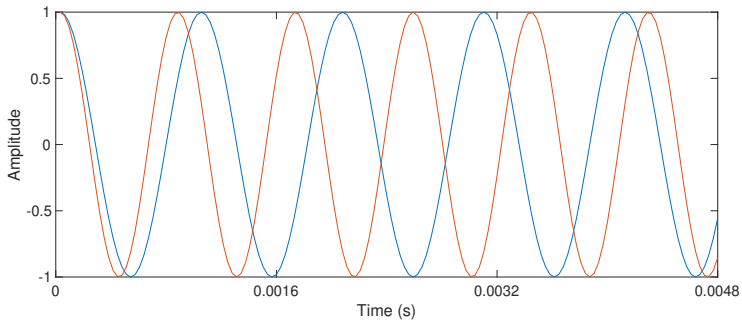
## **Time offset**

→ Different ADCs sampling intervals, RX chain delays etc

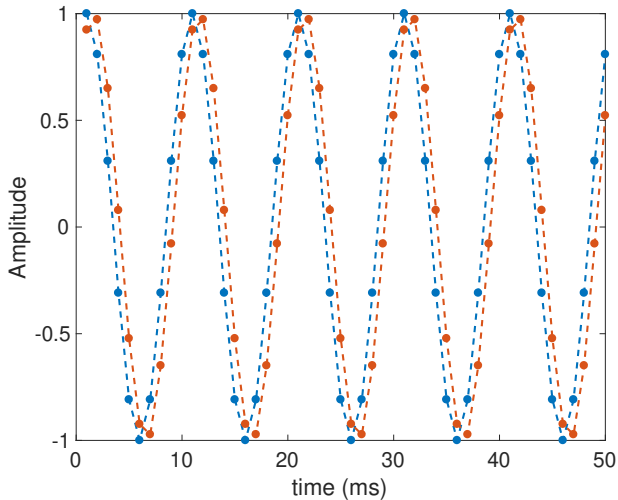
# Challenges - Ideal Reception



# Challenges - Frequency Offset



# Challenges - Phase Offset



# Challenges

## Solutions:

### **Frequency synchronization**

→ Use external clock

### **Time synchronization**

→ Digitally synchronize receiving streams

→ Use gr-devsync

# Outline

- 1 Project Overview
- 2 Algorithms and Challenges
- 3 Results**
- 4 Next Steps

# Test Platforms

## **Octave:**

- Implement all algorithms
- Compare bit error rates
- Choose algorithms to go on

## **GNURadio:**

- Implement chosen algorithms
- Use gr-leo for simulations

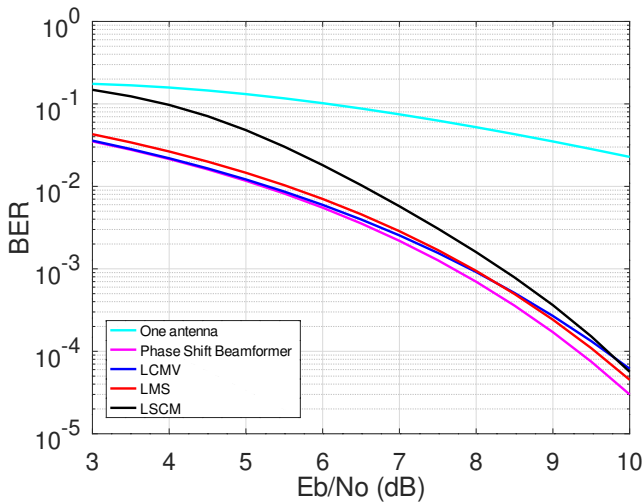


# Results

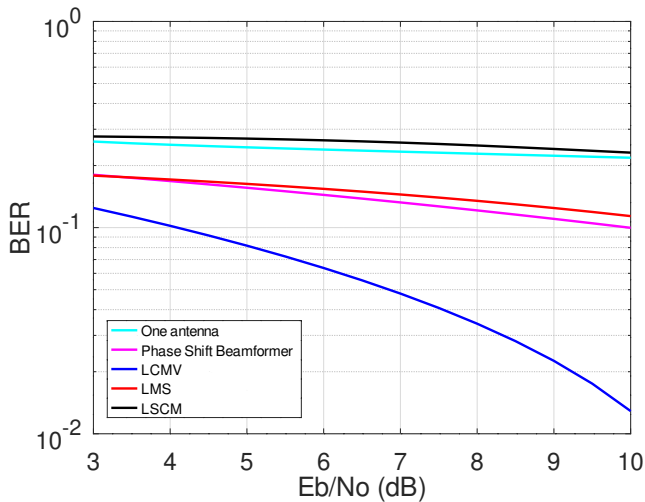
## Test scenarios:

- No interference
- One interferer
- Frequency mismatch
- Real/Assumed angle mismatch

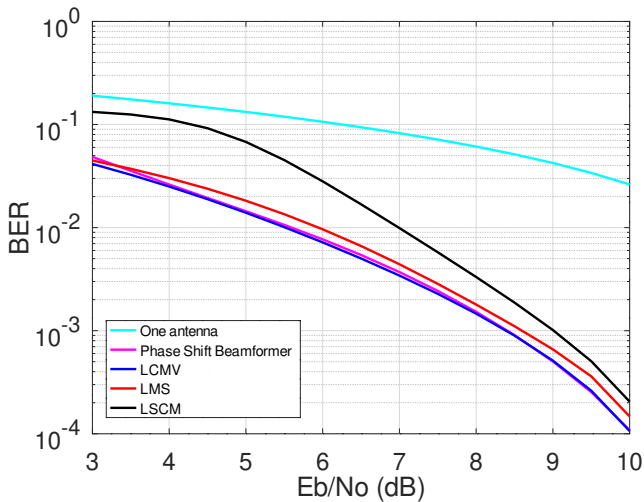
## Octave: No interference



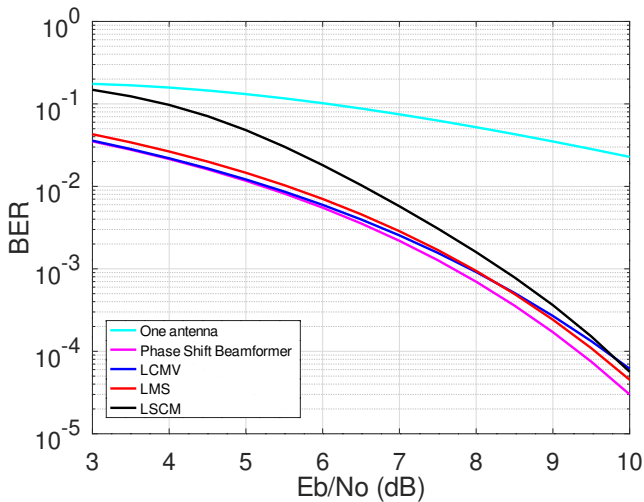
## Octave: Terrestrial interference



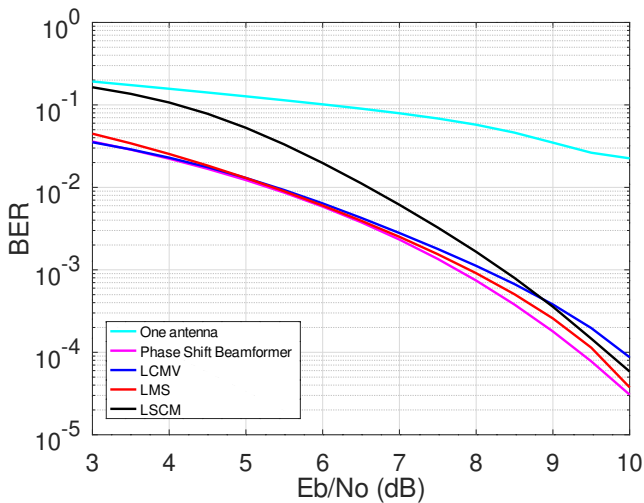
# Octave: Satellite interference



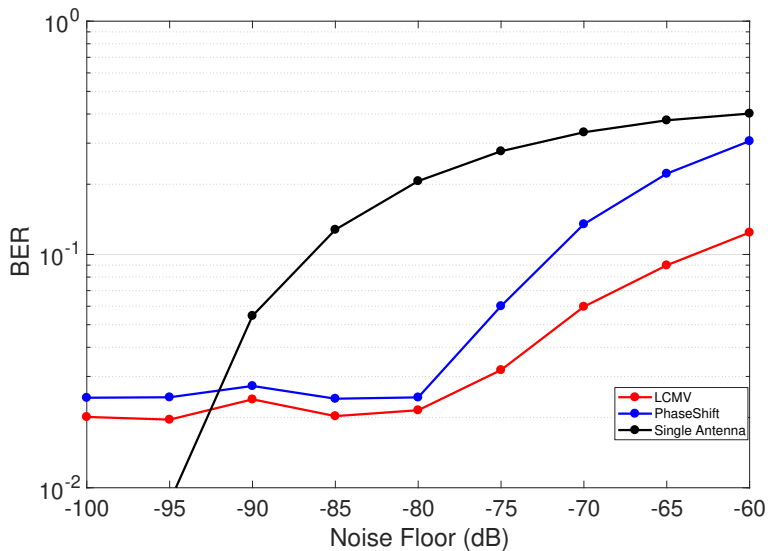
## Octave: Frequency Offset



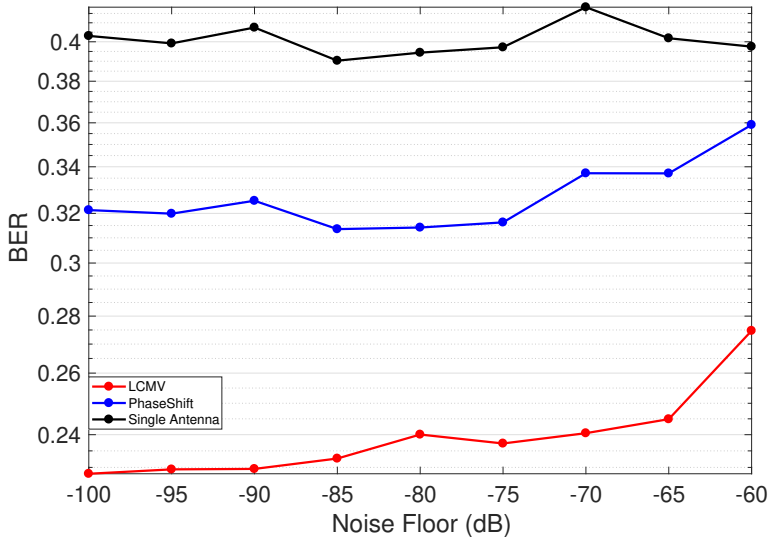
## Octave: Angle Offset



# GNURadio : No interference

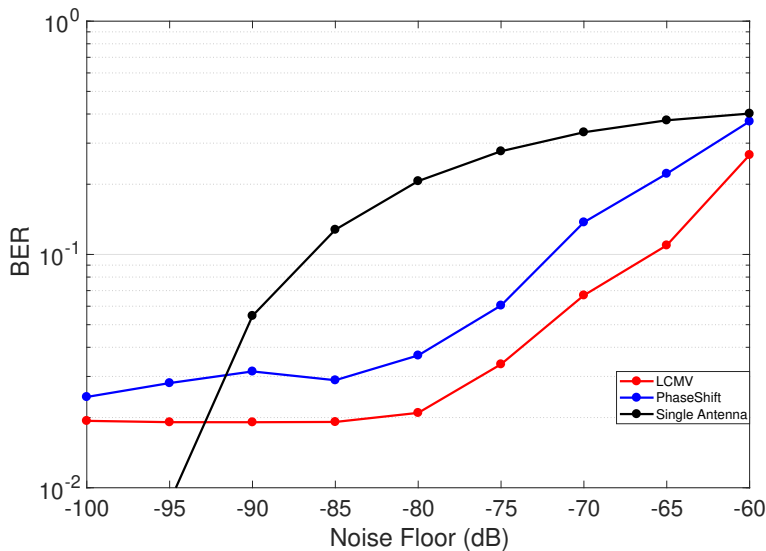


# GNURadio : Interference

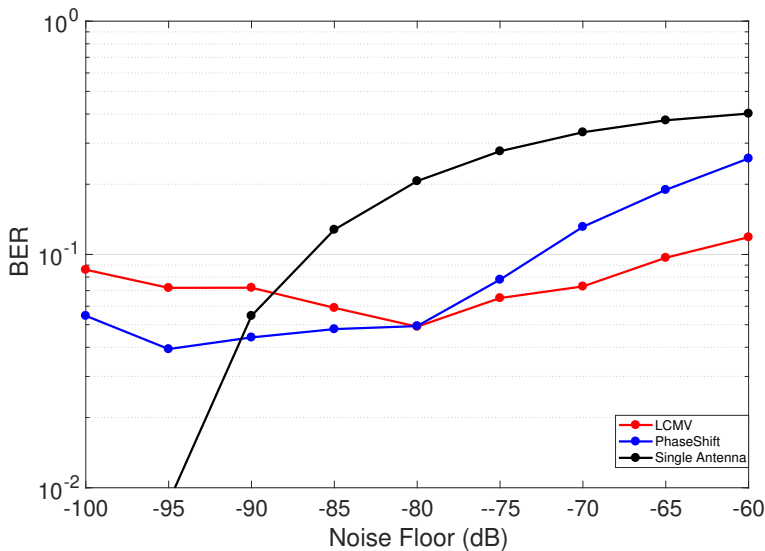




# GNURadio : Frequency Offset



# GNURadio : Angle Offset



# Outline

- ① Project Overview
- ② Algorithms and Challenges
- ③ Results
- ④ Next Steps

## Next Steps

- Construct Array
- Optimize gr-devsync
- Synchronize Streams
- Deploy Array

# Join Us!

Join us on gitlab:

<https://gitlab.com/librespacefoundation/sdrmakerspace/gs-array>