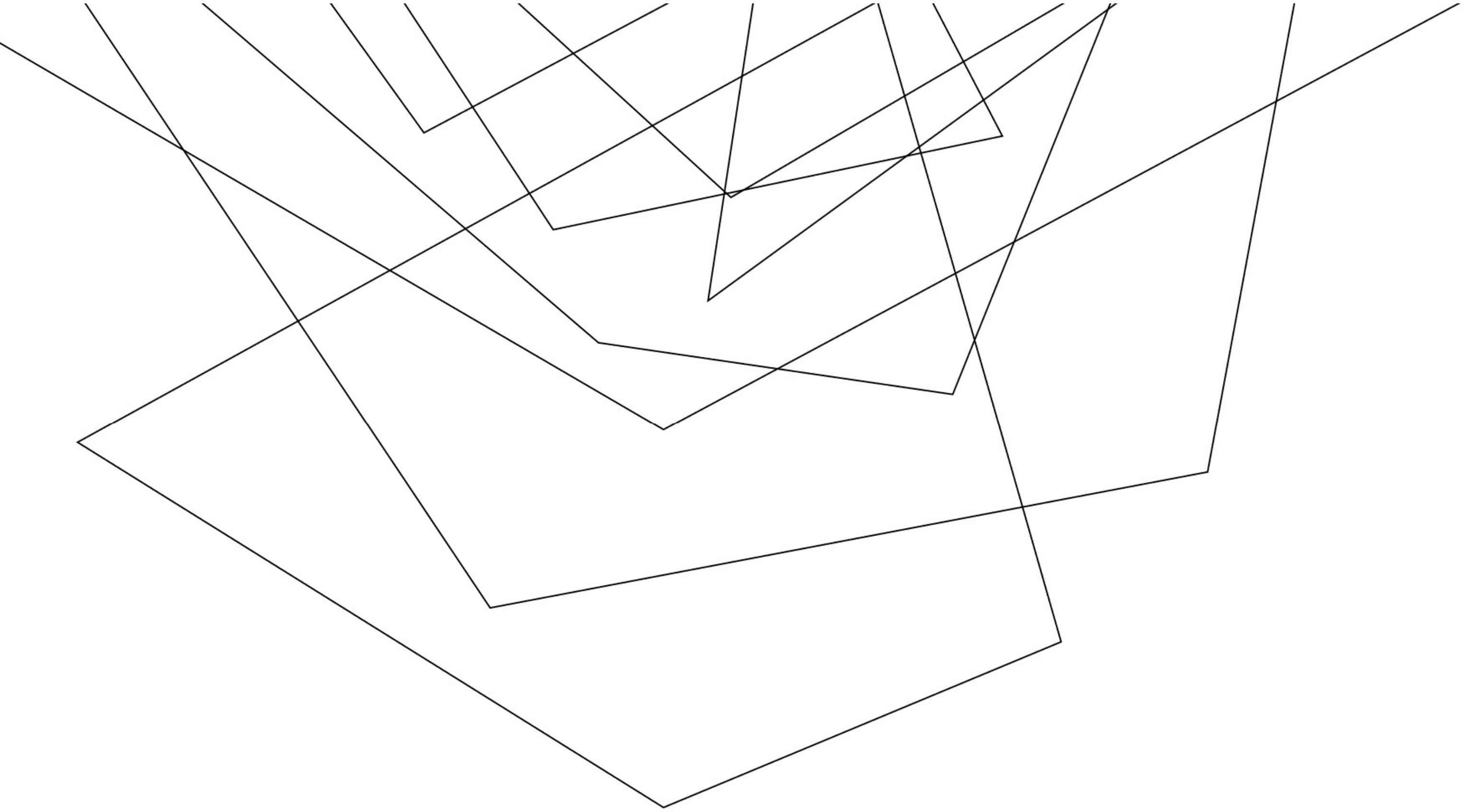
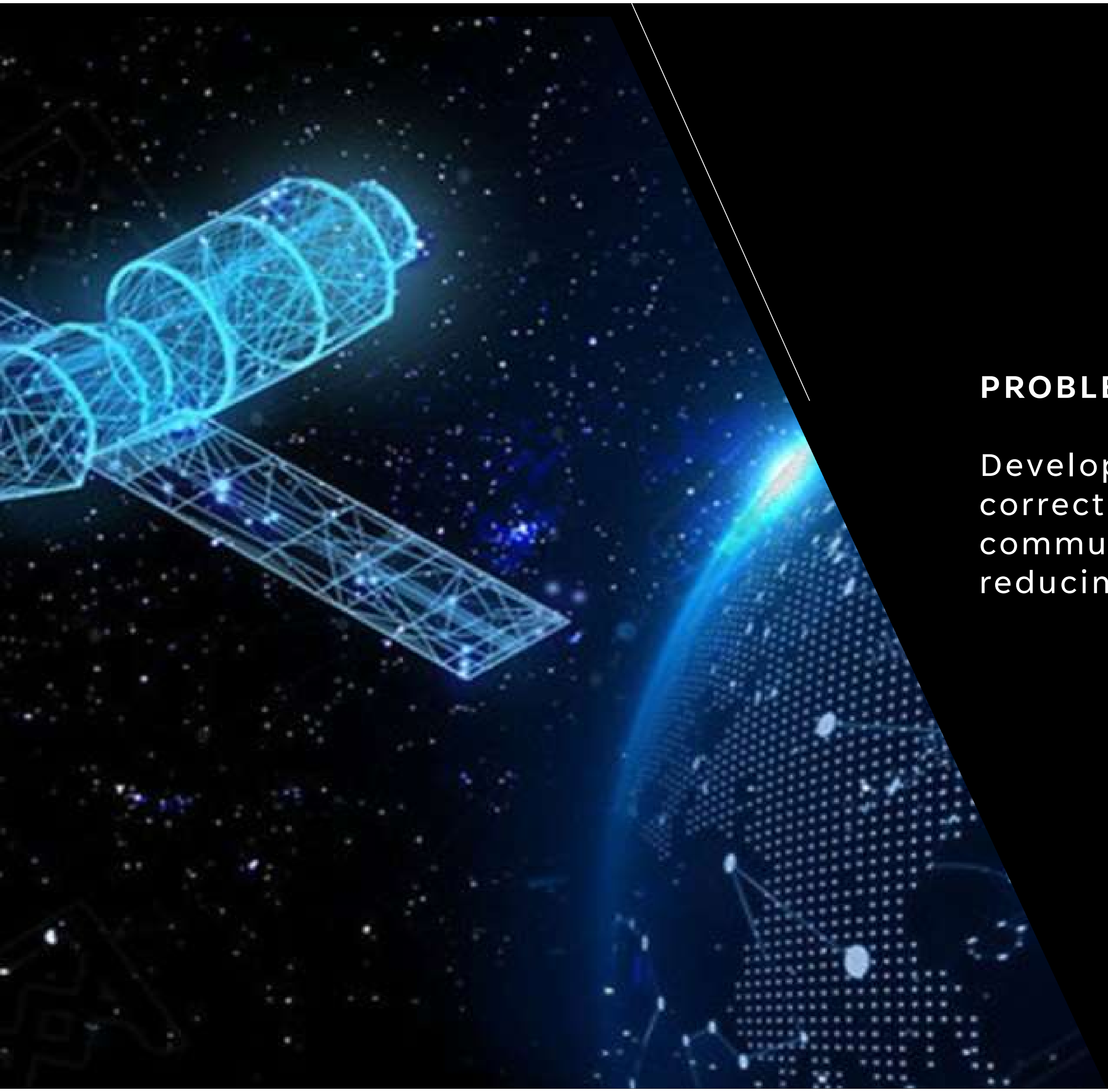


IMPROVING SATELLITE AND DEEP SPACE COMMUNICATIONS USING REED-SOLOMON CODES



PROBLEM STATEMENT:

Develop an optimized reed-solomon error correction scheme to enhance data communication reliability in space missions by reducing computational power requirements.

ENCODING :

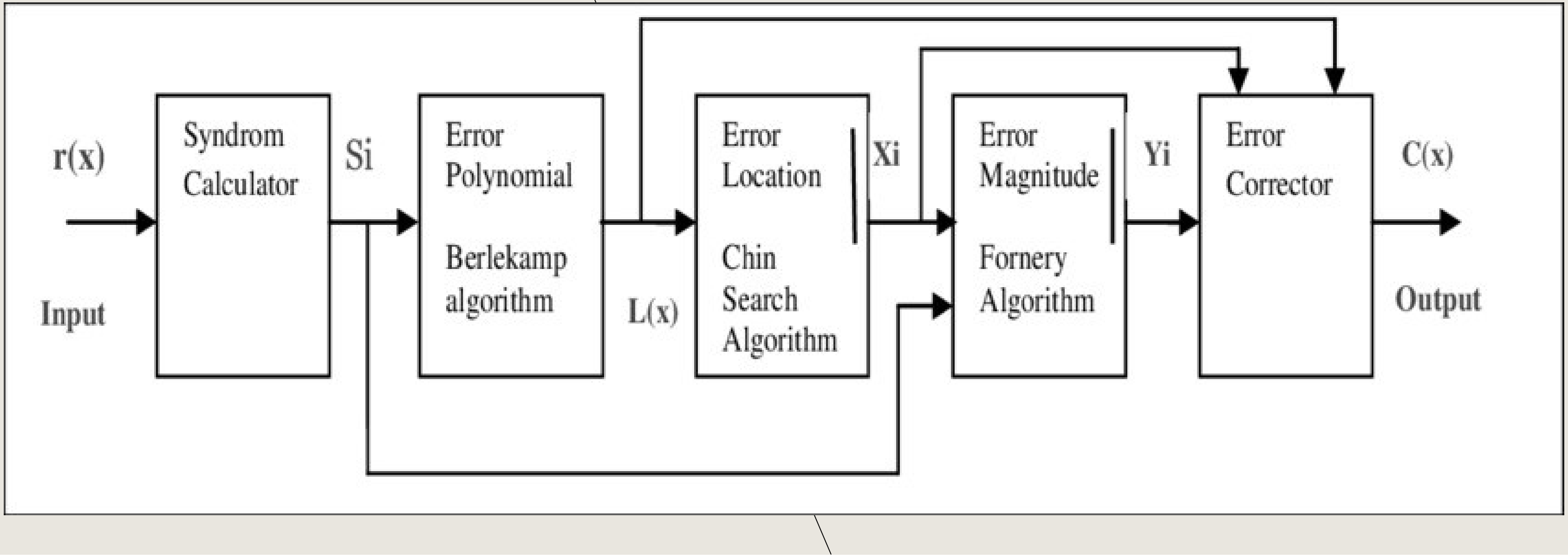
We used the Galois Library in python.

Firstly we defined the Galois field (GF16) and the generator polynomia, later we shifted message polynomial and computed the remainder when divided by the generator polynomial.

All this took place in a function which returned the codeword (codeword is the message followed by the remainder). We performed an example to demonstrate this.

DECODING:

- Syndrome Calculation
- Berlekamp -Massey Algorithm
- Chien Search
- Forney’s Algorithm

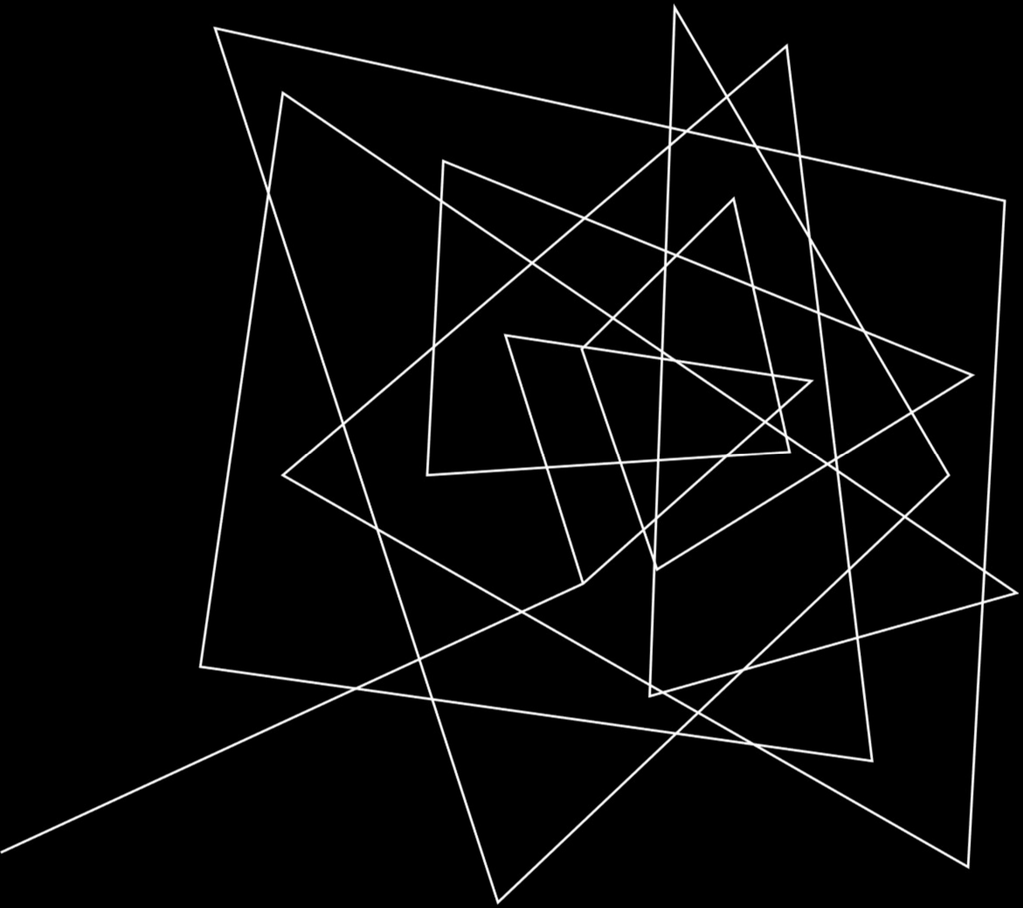


SYNDROME CALCULATION

The syndromes are values that help detect the presence of errors in the received codeword. They are calculated by evaluating the polynomial formed by the received codeword at specific points in the Galois Field.The syndromes are computed by evaluating the received polynomial at different powers of a primitive element (here, GF16(2)), which are the roots of the generator polynomial. If all syndromes are zero, the received codeword is error-free. Non-zero syndromes indicate errors.

BERLEKAMP -MASSEY ALGORITHM

The Berlekamp-Massey algorithm is used to find the minimal polynomial that describes the error locations, known as the error locator polynomial (sigma).The error locator polynomial has roots that correspond to the inverses of the error positions in the codeword.The algorithm iteratively updates the error locator polynomial by calculating discrepancies and adjusting the polynomial if discrepancies are found



CHIEN SEARCH:

The Chien Search is used to find the roots of the error locator polynomial. The positions where the polynomial evaluates to zero correspond to the error positions in the codeword.The positions are calculated in reverse order because the highest degree term in the polynomial corresponds to the first position in the codeword

FORNEY'S ALGORITHM:

Forney's algorithm is used to calculate the magnitude of the errors at the identified positions. The error values are determined by evaluating the error evaluator polynomial (omega) at specific points and dividing by the derivative of the error locator polynomial at those points. This process corrects the received codeword by determining exactly how much the received symbols deviate from the correct ones

ERROR CORRECTOR FUNCTION:

Error corrector function subtracts the error values from the corresponding positions in the received codeword to correct the errors. The result is the original, error-free codeword. For each error position, this function subtracts the corresponding error value from the received symbol at that position and return the corrected codeword. We took the same example to demonstrate all the steps as mentioned.

THANK YOU

Team - HANA